



Process Expert

User Guide

EIO0000001114.17
02/2022



Legal Information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.

As part of a group of responsible, inclusive companies, we are updating our communications that contain non-inclusive terminology. Until we complete this process, however, our content may still contain standardized industry terms that may be deemed inappropriate by our customers.

Table of Contents

Safety Information	11
About the Book.....	12
Understanding EcoStruxure Process Expert.....	15
General Information.....	16
Terminology Used in This Manual.....	16
Introducing EcoStruxure Process Expert	17
EcoStruxure Process Expert.....	18
EcoStruxure Process Expert Architecture	20
General Concepts.....	21
Object Model Concept.....	22
Object Based Engineering	23
Object Template Description	25
Application Templates.....	28
Topological Templates	29
Interface Models	30
Instances of Templates	31
System Engineering Life Cycle Concept	32
The System Engineering Life Cycle	33
Platform Components.....	35
Global Templates Definition Stage	36
Instantiation Stage	37
Project Definition Stage	41
Assignment Stage	44
Generation Stage.....	46
Refinement Stage	47
Configuration Stage.....	49
Mapping Stage.....	51
Build Stage	53
Deployment Stage.....	55
Execution Stage.....	56
Simplified System Engineering Life Cycle.....	57
Working with the Software	58
Starting and Quitting System Server and Clients	59
Logging In and Out of the System Server and Clients	60
Locking and Unlocking the Software.....	62
Starting the System Server Again.....	63
Client/Server Connection.....	64
Quitting the Software.....	68
Multiuser Functionality	70
Multiuser Functionality.....	70
Action Traceability	72
Notification Panel.....	73
Audit Trail.....	76
Activity Log Files.....	80
Software Capabilities	82
Hardware and Software Capabilities.....	83
Supported Data Types.....	86
Naming Rules.....	88

Using the Help.....	92
Software Participant Help	93
Using Help for Different Versions of Schneider Electric	
Templates	94
Help on User-Created Templates	96
User Interface	100
Engineering Client User Interface Description	101
About Dialog Box	104
Engineering Client and Explorer Work Areas.....	105
Organizing Work Areas	108
Working with Panes	111
Using Menus	115
Software Participant Windows	116
Filtering and Sorting Data	118
Keyboard Shortcuts	121
Engineering a System	122
Quick Reference Guides	123
System Creation Workflow.....	124
Task Quick Reference	128
Managing Systems	132
Organizing Systems	133
Creating Systems	135
Managing System Access Passwords	138
Creating System Backups	142
Scheduling System Backups.....	145
Restoring Systems.....	147
Instantiation Stage.....	152
Instantiating Templates.....	153
Organizing the System Application	154
Application Explorer	155
Browsing Application Templates	158
Using the Application Browser Pane	161
Creating Application Folders	165
Assigning Icons to Application Folders.....	170
Creating Instances	171
Counting Instances	174
Accessing Instance Properties	176
Configuring Instances.....	182
Using the Asset Workspace Pane.....	186
Instance Indicators.....	188
Instance Progress Indicators.....	189
Instance Validity Indicator	191
Instance Write-Lock Indicator	192
Linking Instances by Using the Asset Workspace Editor.....	193
Asset Workspace Editor.....	193
Using the Simplified View.....	196
Instance Link Status	197
Linking Instances	199
Using the RTNS Interface	201
Managing Application Folders and Instances	203
Viewing Instance Configuration	204

Changing the Application Hierarchy	206
Copying and Pasting Application Folders and Instances.....	207
Moving Application Folders and Instances.....	209
Updating and Replacing Templates of Instances.....	212
Exporting and Importing Application Objects to/from CSV and XML Formats	219
Exporting Folders and Instances to CSV and XML Formats	220
Exporting Asset Workspaces to CSV Format.....	222
CSV Application Export File	223
Editing the CSV Application Export File	227
XML Application Export File	236
Editing the XML Application Export File to Manage Folders and Instances.....	238
Editing the XML File to Manage Interface Links	243
Importing a CSV or XML File Into the Application	246
Project Definition Stage.....	254
Project Explorer and Project Browser	255
Project Explorer	256
Project Browser	258
Control Project Definition Stage	259
Creating Control Projects.....	260
Creating Control Project Sections.....	264
Managing Control Projects	265
Control Project Section Order.....	266
Exporting Control Projects	267
Importing Control Projects.....	271
Importing a Control Expert Project	284
Managing Data Backup Files.....	286
Supervision Project Definition Stage.....	288
Internal Supervision Architecture	289
Creating Supervision Projects	290
Creating Supervision Project Components.....	294
Creating Clusters	295
Creating Supervision Servers and Clients	296
Creating I/O Devices	299
Creating Communication Ports.....	302
Implementing Redundancy	304
Managing Supervision Projects.....	306
Exporting a Supervision Project.....	307
Importing a Supervision Project.....	311
Participant Project Indicators	321
Participant Project Progress Indicators	322
Participant Project Validity Indicators	324
Participant Project Write-Lock Indicator.....	326
Viewing the Last Action Summary	327
Assignment Stage	328
Creating and Managing Sections, Instances, and Containers	329
Assignment Editor	330
Creating Control Project Sections.....	335
Creating Instances In-Place	338
Creating Supervision Tag Containers	340

Creating Supervision Pages	345
Selecting Supervision Page Templates	346
Assigning Facets of Instances.....	349
Defining the Control Facet Assignment Mechanism	350
Previewing Facets Assignments.....	359
Assigning Control Facets of Instances	362
Assigning Supervision Facets of Instances.....	364
Assigned Facet Description.....	366
Automatic Assignment of Remaining Facets.....	368
Automatic Assignment of Supervision Facets	370
Viewing the Application Structure in Control Projects	371
Managing Assigned Facets.....	376
Facet Actions After Assignment.....	377
Viewing Facet Assignments	379
Order of Control Facets	381
Navigating from Facets to the Instance	382
Generation Stage	383
General Description	384
Generation Process Description	385
Control Project Specifics	388
Automatic Generation of Sections.....	394
Selecting Versions of Types in Control Facets During Generation	396
Statuses of Facets	411
Generating a Project for the First Time.....	413
Generating Entire Projects	414
Generating Sections, Containers, and Nodes	415
Generating Projects After Changes	416
Generating Participant Projects After Changes	417
Regenerating a Control Project	421
Generating Equipment in Supervision Projects.....	424
Generating Equipment Based on the Application	425
Generation Diagnostic Messages.....	427
Generation Diagnostic Messages	428
Managing Generated Facets.....	430
Managing Generated Facets	431
Refinement Stage.....	434
Control Project Refinement Stage	435
Refining the Logical Control Project	436
Identifying Code Generated by the Software.....	442
Creating and Updating Device DDT Variables in Logical Control Projects	445
Managing the Consistency of Control Logic.....	449
About Variables for Peer to Peer Communication.....	452
Managing Network Variables.....	459
Managing Peer to Peer Communication.....	462
Overview	462
Peer to Peer Communication Configuration – Remote Project Selection 1/2.....	463
Peer to Peer Communication Configuration – Variable Mapping 2/ 2	465

Sections and Logic Created for Peer to Peer Communication	467
Creating Peer to Peer Communication Mapping	470
Modifying and Deleting Peer to Peer Communication Mapping	472
Managing Peer to Peer Communication Example	476
Supervision Project Refinement Stage	493
Refining the Supervision Project	494
Configuration Stage	498
Creating the Topology of the System	499
Organizing the System Topology	500
Selecting Topological Templates	504
Using Configuration Change on the Fly (CCOTF)	505
Configuring a Controller	507
Configuring a Station Node	516
Configuring an STB Island	523
Configuring a PROFIBUS Remote Master and Decentralized Peripherals	527
Configuring Modbus TCP and EtherNet/IP Devices	531
Configuring HART Modules	536
Configuring an Ethernet Network	538
Connecting to an Ethernet Network	540
Managing the Topology	548
Moving Topological Folders and Entities	549
Updating Templates of Topological Entities	550
Exporting Topological Entities	552
Importing Topological Entities	555
Exporting and Importing Devices to/from CSV Format	564
Exporting Devices to CSV Format	565
Editing the CSV Device Export File	567
Importing a CSV File Into the Topology	571
Mapping Stage	576
Control Project Mapping Stage	577
Creating Control Project Executables	578
Mapping Services	580
Mapping Communication	582
Mapping Network Variables	594
Mapping Hardware	602
Exporting and Importing Hardware Mappings	609
Exporting Hardware Mapping Data	610
Editing the Hardware Mapping Export File	612
Importing Hardware Mappings	616
Supervision Project Mapping Stage	618
Creating Supervision Project Executables	619
Mapping Services	622
Mapping Communication	624
Build Stage	628
Control Project Build Stage	629
Building the Logical Control Participant Project	630
Changing the Built Control Participant Project	636

Managing Hardware Abstraction Layer (HAL) Variables and DFBs.....	639
Supervision Participant Project Build Stage.....	642
Building the Supervision Participant Project.....	643
Changing the Built Supervision Participant Project.....	647
Deployment Stage	648
Control Project Deployment Stage	649
Control Participant Project Deployment Process	650
Deploying to the Controller Simulator	654
Communication Between the Computer and Engines.....	657
Managing Passwords of Deployed Control Participant Projects	658
Deploying to Redundant Controllers	665
Deploying the Control Participant Project	667
Re-Deploying the Control Participant Project.....	672
Deploying Data to Controllers and Controller Simulators	677
Deploying Control Project Changes	682
Changes Requiring to Stop the Controller	692
Supervision Project Deployment Stage.....	693
Preparing for Deployment	694
Deploying the Built Supervision Participant Project from the Executable	695
Deploying the Built Supervision Participant Project from the Station Node.....	698
Deploying Supervision Participant Project Changes	703
Execution Stage	706
Control Project Execution Stage.....	707
Control Participant Project Execution Process	708
Execution With Redundant Controllers.....	711
Starting and Stopping Controllers	713
Synchronizing Redundant Controllers.....	719
About Changes Made Online	723
Making Changes Online.....	727
Backing Up Controller Data.....	733
Importing Sections from the Logical Control Participant Project.....	738
Main Redundant Controller Scenarios.....	741
Redundant Controller Reservation and Connection Scenarios	744
Updating the Logical Control Participant Project and Controller Configuration	748
Supervision Project Execution Stage	757
Executing the Deployed Supervision Participant Project.....	758
Engineering M580 Safety Systems	759
Using Modicon M580 Safety Controllers	759
M580 Safety Systems in the Instantiation Stage	760
M580 Safety Systems in the Project Definition Stage	761
Creating M580 Safety Control Projects	761
Exporting and Importing M580 Safety Control Projects.....	763
Importing a Safety Control Project Associated to an Existing M580 Safety Controller	764

M580 Safety Systems in the Assignment Stage.....	766
M580 Safety Systems in the Generation Stage.....	767
M580 Safety Systems in the Refinement Stage	767
M580 Safety Systems in the Configuration Stage	769
Creating and Configuring M580 Safety Controllers.....	769
Managing the Safety Password	770
Exporting and Importing M580 Safety Controllers	772
M580 Safety Systems in the Mapping Stage	774
M580 Safety Systems in the Build Stage.....	774
M580 Safety Systems in the Deployment Stage	775
M580 Systems in the Execution Stage.....	776
Using Supervision with M580 Safety Systems	777
Managing Communication.....	779
Communication Through Modbus TCP Ethernet Implicit Messaging (I/	
O Scanning)	780
Communicating with Devices Through Modbus TCP Implicit	
Messaging (I/O Scanning).....	781
Communicating with STB Islands Through Modbus TCP Implicit	
Messaging.....	784
Communicating with PROFIBUS Remote Masters Through	
Modbus TCP Implicit Messaging	785
Peer to Peer Communication Through Modbus TCP Ethernet	
Implicit Messaging.....	787
Communication by Using Modbus TCP Ethernet Explicit	
Messaging	788
Communication with Devices by Using Modbus TCP Ethernet	
Explicit Messaging	789
Communication by Using EtherNet/IP Explicit Messaging	794
Communication with Devices by Using EtherNet/IP Explicit	
Messaging.....	795
Communication Through Modbus Explicit Messaging	798
Communication with Devices Through Modbus Explicit	
Messaging.....	799
Example Overview	804
Example A.....	808
Example B.....	812
Example C.....	818
Example D.....	825
Monitoring Modbus Device Data Directly from Supervision.....	830
Monitoring Modbus Device Data Directly from Supervision	831
Communication with Devices by Using a PROFIBUS Master	
Module.....	835
Communication with Devices by Using a PROFIBUS Primary	
Module	836
Managing, Modifying, and Creating Templates	837
Global Templates Library.....	838
Global Templates Explorer.....	839
Global Templates Folders	845
Global Templates	847
Managing the Control Constituent Password.....	849
Global Templates Editors	852

Opening Global Templates Editors	853
Opening Global Templates Editors.....	853
Interface Editor	854
Interface Editor	854
Facet and Composite Template Editors.....	859
Facet Editor	860
Composite Editor	864
Common Template Editor Components	867
Common Template Editor Toolbars and Menus.....	868
Common Template Editor Panes	874
Common Template Editor Filters.....	876
Using the Radar View	877
Using the Simplified View in Template Editors.....	878
Managing Global Templates	880
Managing Global Templates	881
Saving Changes In Global Templates	882
Copying and Pasting Global Templates and Folders	885
Exporting Templates.....	886
Importing Templates.....	889
Purging the Global Templates Library	896
Updating, Replacing, and Duplicating Global Templates.....	898
Template Modification Strategy	899
Updating Global Templates and Templates of Elements	901
Updating Global Templates at the Folder Level.....	905
Replacing the Template of an Element of a Global Template	907
Duplicating Global Templates	912
Replace Dialog Box	916
Editing and Extending Interfaces In-Place	917
Updating References in Parent After Editing Child Element – Example	922
Software Services	926
System Documentation	927
About System Documentation.....	927
System Documentation Report Content.....	928
Application Domain Report Data.....	929
Project Domain Report Data	929
Topology Domain Report Data.....	931
Global Templates Domain Report Data	933
Generating System Documentation Reports	934
Content Repository	935
Content Repository Explorer.....	936
Temporary Contents Repository Files	940
Appendices	941
Linking Instances by Using the Links Editor	942
Links Editor	943
Linking Instances by Using the Links Editor.....	946
Glossary	949
Index	957

Safety Information

Important Information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

Please Note

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

Document Scope

This manual contains general information about EcoStruxure Process Expert and guides you through the stages to engineer a system.

This manual is organized in five main parts:

- Understanding EcoStruxure Process Expert:
Provides information on the object model and the system engineering life cycle concepts that the implementation of automation systems is based on.
- Working with the software:
Describes the graphical user interface and how to use the different tools that are at your disposal.
- Engineering a system:
Provides step by step instructions to help you implement a system by following the system engineering life cycle.
- Global Templates library and editors:
Describes the Global Templates library, the templates that it can contain, and the editors that allow you to manage them.
- Software services:
Provides information on the content repository and on how to generate system documentation reports.

It is written for users who are familiar with the engineering of Control and Supervision systems.

For further information, refer to the related documents that are mentioned in this manual.

Validity Note

This document is valid for EcoStruxure Process Expert 2021 or later. It supersedes any previous version.

Related Documents

Title of documentation	Reference number
EcoStruxure™ Process Expert - Security Deployment Guide	EIO0000004234 (eng)
EcoStruxure™ Process Expert - General Purpose Library User Guide	EIO0000004045 (eng)
EcoStruxure™ Process Expert - General Purpose Library Process Templates Reference Manual	EIO0000004043 (eng)
EcoStruxure™ Process Expert - General Purpose Library Device Templates Reference Manual	EIO0000004044 (eng)
EcoStruxure™ Process Expert - General Purpose Library Classic Process Templates Reference Manual	EIO0000000987 (eng)
EcoStruxure™ Process Expert - General Purpose Library Classic Device Templates Reference Manual	EIO00000001308 (eng)
EcoStruxure™ Process Expert - General Purpose Library Classic Communication Templates Reference Manual	EIO00000001311 (eng)
EcoStruxure™ Process Expert - General Purpose Library Classic Diagnostic Templates Reference Manual	EIO00000001526 (eng)
EcoStruxure™ Process Expert - Foundation Application Templates User Guide	EIO0000002403 (eng)

Title of documentation	Reference number
EcoStruxure™ Process Expert - Control Participant Services User Guide	EIO0000001524 (eng)
EcoStruxure™ Process Expert - Supervision Participant Services User Guide	EIO0000001525 (eng)
EcoStruxure™ Process Expert - Runtime Navigation Services User Guide	EIO0000001574 (eng)
EcoStruxure™ Process Expert - Global Templates Reference Manual	EIO0000001986 (eng)
EcoStruxure™ Process Expert - Installation and Configuration Guide	EIO0000001255 (eng)
EcoStruxure™ Process Expert - Licensing Guide	EIO0000001261 (eng)

You can download these technical publications at <https://www.se.com/myschneider>, *Document Downloads* section.

Registration required.

mySchneider Support Portal

Visit <https://www.se.com/myschneider> for support, software updates, and latest information on EcoStruxure Process Expert.

Registration required.

Product Related Information

The examples in this manual are given for information only.

⚠ WARNING	
UNINTENDED EQUIPMENT OPERATION	
Adapt examples that are given in this manual to the specific functions and requirements of your industrial application before you implement them.	
Failure to follow these instructions can result in death, serious injury, or equipment damage.	

NOTE: Templates shown in examples throughout this manual may differ from the actual templates contained in the supplied Schneider Electric libraries.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines.¹
- Test each implementation of a system for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Understanding EcoStruxure Process Expert

What's in This Part

General Information.....	16
Introducing EcoStruxure Process Expert.....	17
General Concepts	21

Overview

This part introduces EcoStruxure Process Expert and describes the concepts and underlying mechanisms that are applied to engineer an automation system.

In this manual, *software* is used to refer to EcoStruxure Process Expert.

General Information

What's in This Chapter

Terminology Used in This Manual	16
---------------------------------------	----

Overview

This chapter contains general information about the software.

Terminology Used in This Manual

EcoStruxure Process Expert

To make reading easier, *the software* is sometimes used in place of EcoStruxure Process Expert throughout this manual.

Item Names

Because you can customize the names of most items that are displayed in the user interface, this manual uses predominantly the default names created by the software to describe items such as:

- Systems
- Topological entities
- Folders
- Projects and their components
- Instances

Introducing EcoStruxure Process Expert

What's in This Chapter

EcoStruxure Process Expert	18
EcoStruxure Process Expert Architecture.....	20

Overview

This topic introduces EcoStruxure Process Expert and describes its main distinctive features.

EcoStruxure Process Expert

Features

EcoStruxure Process Expert is the next generation process automation system to engineer, operate, and maintain your entire plant.

It is a single environment, which interconnects and manages all the components of the automation architecture:

- Standard field buses.
- Instrumentation.
- Intelligent, connected devices.
- Ethernet-based control networks.
- Controllers.
- Engineering and operator stations.
- Alarm management.

The software delivers everything you need to run a sustainable, productive, and market-agile plant:

- Scalable and powerful automation platform.
- Full process traceability.
- Embedded energy management.
- Fast and efficient engineering and deployment of configurations.
- Powerful native diagnostic features.
- Fast access to process data by using Runtime Navigation Services.
- Integrated and specialized, validated object libraries.

Benefits

Using EcoStruxure Process Expert provides the following benefits, which help you:

- Reuse automation objects with proven, standard designs.
- Reduce commissioning time and cost.
- Reduce engineering and integration risks.
- Put in place appropriate preventive, predictive, and prescriptive maintenance so that unplanned downtime approaches zero.
- Improve quality of production-related decisions and maximize operational profitability.
- Optimize the management of flows and resources to maximize output and profitability.
- Manage regulatory requirements.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

When engineering automation systems by using EcoStruxure Process Expert:

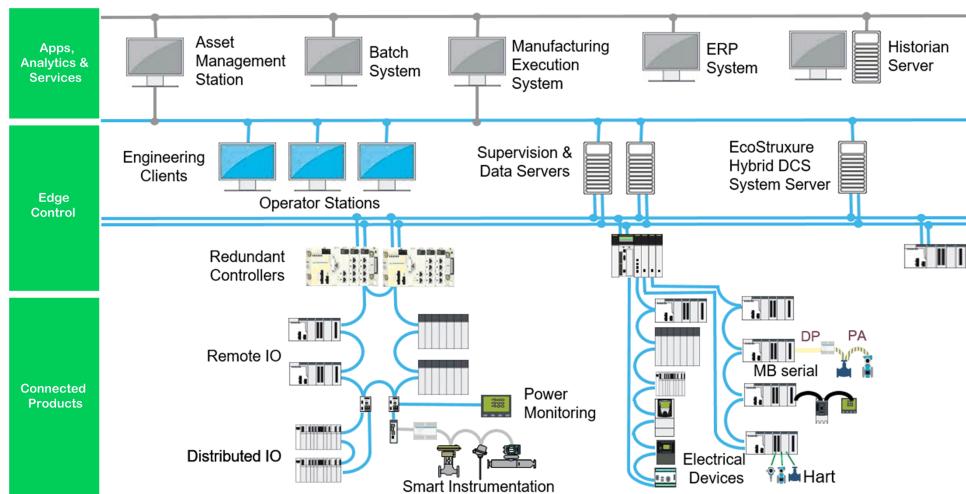
- Identify and understand all implications and consequences of any changes on the automation system.
- Evaluate and simulate any actions that have an influence on the automation system.
- Implement elementary protective measures in the process of engineering automation systems.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

EcoStruxure Process Expert Architecture

Integration with the Plant

EcoStruxure Process Expert offers a scalable architecture, future-ready network with high availability for continuous operation of your plant.



General Concepts

What's in This Chapter

Object Model Concept	22
System Engineering Life Cycle Concept.....	32

Overview

This chapter describes the object model concept and how it is applied within the software. It also describes the concept of engineering an automation system in stages.

Object Model Concept

Overview

This section describes the concept of object model and the resulting templates that are used to engineer automation systems by using the software.

Object Based Engineering

Description of the System

In EcoStruxure Process Expert a *system* is the representation of a physical automation system and is defined by three independent parts:

- The *application*: Defines the functions of the system and the corresponding process hierarchy.
- *Projects*: Define the Participant projects that implement the logic and function of the system:
 - Control Projects
 - Supervision Projects
- The *topology*: Defines the hardware and software infrastructure of the system. It consists of:
 - Controllers
 - Devices connected to field buses
 - Networks
 - Station nodes (their hardware and software components)

Object Model Concept

Object models represent the application and topology of a system.

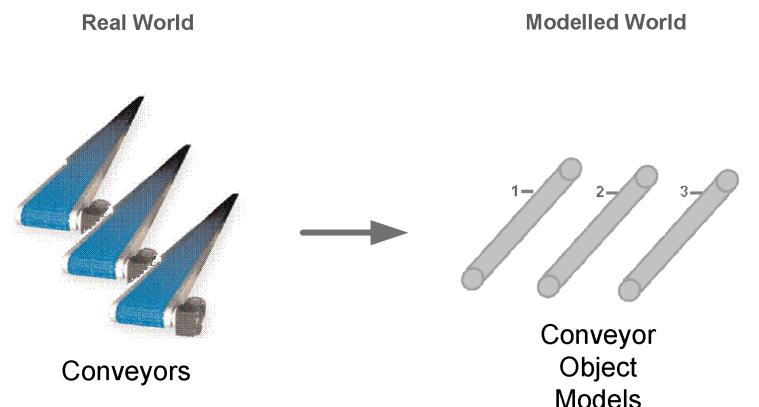
There are different types of object models representing the application of the system based on their purpose:

- *Process objects* representing, for example, the I/Os of pumps, valves, and sensors in several tanks.
- *Device objects* representing, for example, the functions to control a process object such as a motor by using a drive.
- *Communication objects* representing, for example, the functions to manage communication with device objects through the port of an M580 controller with Modbus TCP explicit messaging.

Likewise, object models representing the topology of the system are called *topological objects*. They can represent, for example, the CPU module of an M580 controller.

The approach of representing any entity or function that exists in the real world with the use of object models is referred to as object-oriented modeling and is based on the abstraction principle: Process of reducing the information to retain only what is relevant for a particular component or function independently of the context.

The following figure illustrates the application object-oriented modeling of a conveyor.



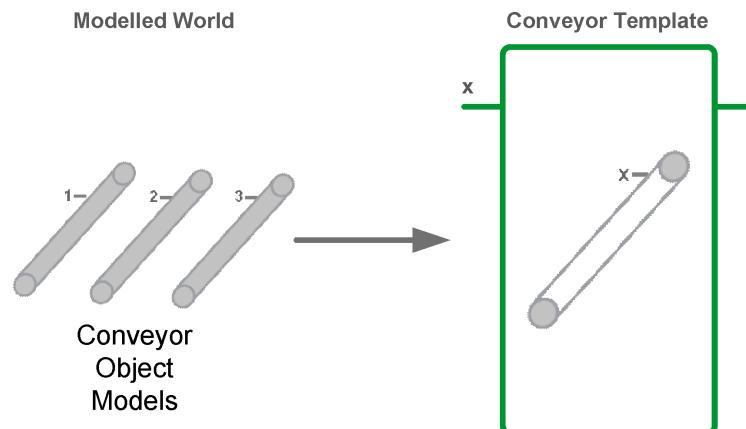
From Object Model to Object Template

The next step consists in creating, based on the object model, a component that models an item or a function of the real world. This component needs to be generic enough to be reused in different systems.

The standardization principle is used to achieve this objective: Process of finding the common characteristics of different items or functions.

The resulting component is called an object *template*.

The following figure illustrates the process of standardizing an object model representing a conveyor to create a template, which allows modelling conveyors and similar items (such as a mixer) in different systems.



Object Template Description

Overview

Schneider Electric object templates created from object models are called Global Templates and are grouped in libraries. When a library is installed, its templates appear in the **Global Templates Explorer**, page 839. These templates can be used in any system that is created by using EcoStruxure Process Expert.

To make reading easier, the word *templates* is used to refer to Global Templates.

Templates provide collaboration and data sharing between Participants and the software platform.

By doing so, templates reduce the engineering effort and provide consistency.

Object Template Classification

Templates can be placed one inside the other, linked, and can include several capabilities to model an item or a functionality of the real world. Therefore, the distinction is made between three types of templates:

- **Facet templates:** They are the smallest templates available. They encapsulate a constituent, which is provided by a specific Participant to deliver the respective service. As such there are:
 - Control facet templates providing Control services. For example, in the General Purpose Library, the service that is provided by the \$MOTORGP_UL facet template is the on/off motor management functionality. The functionality is implemented by using the MOTORGP function block, which is encapsulated inside the template.
 - Supervision facet templates providing Supervision services. For example, in the General Purpose Library, the service that is provided by the \$BlowerLeftGP_m_CG genie facet template is the animated graphical representation of a motorized blower, which can be displayed on an operator station. The animated graphic is encapsulated inside the template.
- **Composite templates:** They consist of one or more facet templates and/or other composite templates. Composite templates that are included inside another composite template are called composite references.

Depending on their composition, composite templates can be:

- Control composite templates: Contain only Control references.
- Supervision composite templates: Contain only Control references.
- *Control module* templates: Highest level of composite template, which contains both Control and Supervision references.

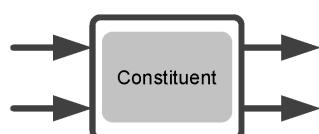
NOTE: Facet templates that are included inside a composite template are called facet references.

- **Interface models:** They are considered as a template because they are generic and can be reused but they are actually a mechanism that allows linking to each other:
 - Any combination of facet and composite templates.
 - Instances of templates.

There are different types of interface models depending on the objects that they link. You can configure interface models.

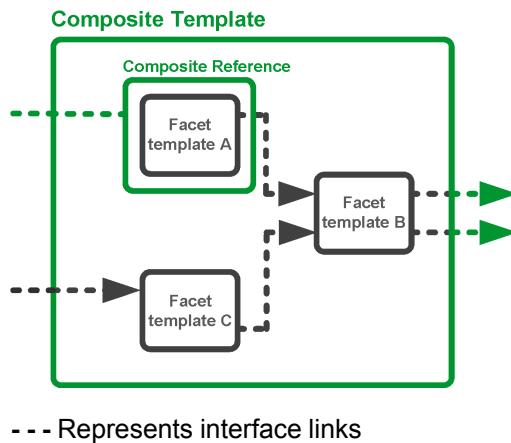
For details, refer to the help of the Schneider Electric libraries for EcoStruxure Process Expert.

The following figure illustrates a single facet template.



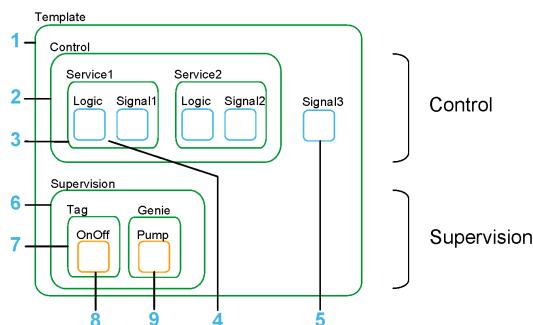
The arrows represent the inputs and outputs of the facet which propagate the data.

The following figure illustrates a composite template including another composite template (composite reference) and two facet templates.



Template Terminology

The following example shows the graphical representation of a sample composite template. The names of elements may vary.



NOTE: Interfaces are not represented since they are mechanisms for managing data exchange between references.

The table explains the color coding that is used in the generic composite template **Template**.

Color	Description
Light blue	Control facet template, which encapsulates a functionality that is provided by the Control Participant.
Orange	Supervision facet template, which encapsulates a functionality that is provided by the Supervision Participant.
Green	Composite template, which can contain: <ul style="list-style-type: none"> Control and/or Supervision facet templates Other composite templates

NOTE: This color coding is used in the user interface to differentiate the different types of templates and related services.

The table describes the items that compose the generic composite template **Template** and the terminology that is used to designate these items depending on the context.

Item	Description
1	Template is the highest level composite template and is called a control module because it provides both the Control and Supervision functionalities that are required to manage the object that is modeled by the template.
2	Control is a Control composite template because it contains Control facet templates. It is also called a composite reference as it is referenced inside the composite template Template . In the context of Template , Control is an element of Template
3	Service1 is a composite reference. In the context of composite reference Control , Service1 is an element of Control . The same applies to composite reference Service2 . Both Service1 and Service2 are Control composite templates since they contain only Control facets.
4	Logic and Signal1 are Control facet references. In the context of the composite reference Service1 , they are elements of Service1 .
5	Signal3 is a Control facet reference. In the context of the control module Template , Signal3 is an element of Template .
6	Supervision is a Supervision composite reference because it is contained in composite template Template . It contains Supervision elements (composite references Tag and Genie). In the context of the control module Template , Supervision is an element of Template .
7	Tag is a Supervision composite reference because it is contained in composite Supervision and contains only a Supervision element (OnOff). Likewise Genie is a Supervision composite reference.
8, 9	OnOff and Pump are Supervision facet references and elements of templates SuperTagvision and Genie respectively.

Types of Object Templates

Like object models represent the application and the topology of the system, so do templates.

As a result, the following types of templates exist:

- *Application templates*: Facet and composite templates, which define the functional components in the application of the system.
- *Topological templates*: Facet and composite templates, which define hardware and software components of the topology of the system.

Application Templates

Overview

Control and Supervision facet templates encapsulate the respective Participant capabilities:

- For Control:
 - **Logic** facet templates: Implement a specific functionality, for example, discrete motor management.
- For Supervision:
 - *Data* facet templates: Allow defining tags and user interface messages that are required for acquiring runtime data.
 - *Genie* facet templates: Represent runtime data by using an animated graphic.

Facet templates encapsulate the respective Participant capabilities. For Control, these are **Logic** facet templates, which implement a specific functionality, for example, discrete motor management.

Based on a combination of these, application control module templates have been created, which allow you to model the application of a system in terms of Control and Supervision services.

The templates are available in the form of libraries and after installation, appear in the **Global Templates Explorer**, page 839 where they are grouped by category.

For details, refer to the help of the Schneider Electric libraries for EcoStruxure Process Expert.

NOTE: There is a relation between Control and Supervision facet templates because for a given Control service, which is provided by a Control facet, the Supervision facet represents the part of the service that is visible to the user (for example supervisory user interface and mode of operation).

Topological Templates

Introduction

Topological templates allow you to model the hardware and software infrastructure of the topology of a system by creating topological entities.

They are located in the Foundation library and grouped by category.

Sub-Types of Topological Templates

The following subtypes of topological templates are available:

- **Network:** Models control and field bus networks.
For example: Ethernet LAN, Modbus, CANopen, Profibus DP, RIO S908, RIO Ethernet, local bus for controllers and Advantys islands).
- **Network devices:** Models the field bus devices.
For example: Gateways.
- **Station node:** This template models the components of the different types of stations that are connected to the Ethernet network.
For example: network interface cards (NICs), Supervision software, OPC Factory Server (OFS) software, and Control software.
- **Controller:** Models the controller hardware that is managed by the Control Participant.
For example: Hardware of the Quantum platform.
- **Device I/O:** Models devices and remote I/Os including third party devices that can be connected as nodes to Ethernet networks.
For example: Advantys Island I/Os, energy devices, motor starters, and instrumentation devices.

Interface Models

Introduction

Interface models allow you to define:

- data exchange between two object instances or two object references.
- dependencies between two object instances or two object references such as compatibility or requirement rules.

Types of Interface Models

The following types of interface models are predefined:

- **Physical**: Allow data exchange at the physical level (modules) between:
 - Topological instances
 - Topological references
- **Communication**: Allow data exchange at the logical level (I/O scanner, OPC Factory Server software, Supervision I/O devices) between:
 - Topological instances
 - Topological references
- **Application**: Allow data exchange between:
 - Application instances/references
 - Topological instances/references
- **Mapping**: Allow data exchange during the mapping stage between application objects (representing the logical projection of the hardware) and topological objects (representing the actual hardware defined in the topology).

Instances of Templates

From Templates to Instances

The software lets you create application and topological instances from the corresponding templates.

Because control module templates integrate a range of Control and Supervision services, editable parameters, and expose a set of interfaces, you can customize each instance to fulfill specific requirements.

The process of creating instances from templates and configuring them is called *instantiation*.

To access and configure their parameters, instances produce facets, which are based on the facet templates that each control module template contains.

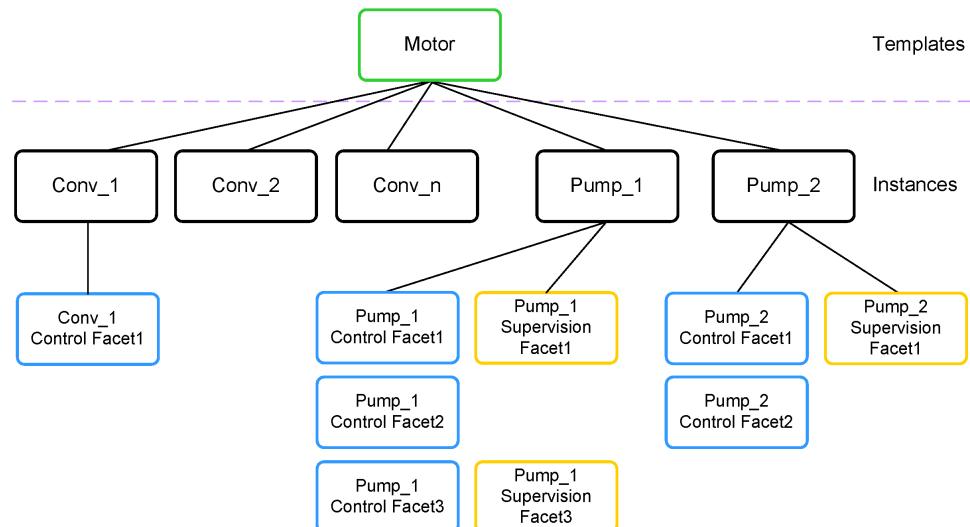
During the instantiation, you can:

- Select services among those that the template provides. For Schneider Electric templates, some services are mandatory (for example, **Logic**) and some are optional (for example, **Supervision**).
- Edit the values of parameters (for example, provide a high limit value for an analog level sensor).
- Link instances to each other (for example, link a motor instance to the instance representing the functions of the variable speed drive that controls the motor).

When you enable (select) a service, the software creates a facet, which is an instance of the facet template that provides the service. This facet is associated to the instance of the template that references the service.

The other way around, when you disable (unselect) a service, the software deletes the corresponding facet.

The following figure shows an example of several instances with different configurations that you can create from one template.



Only some facets of instances Conv_1, Pump_1, and Pump_2 are shown for illustration purposes.

System Engineering Life Cycle Concept

Overview

The concept of breaking down into stages the workflow to engineer an automation system is referred to as the *system engineering life cycle* in this manual.

This section provides a general description of each stage using:

- A stage map showing the position of the stage within the system engineering life cycle.
- A definition to explain the purpose of the stage.
- An example to facilitate the understanding.

The System Engineering Life Cycle

Overview

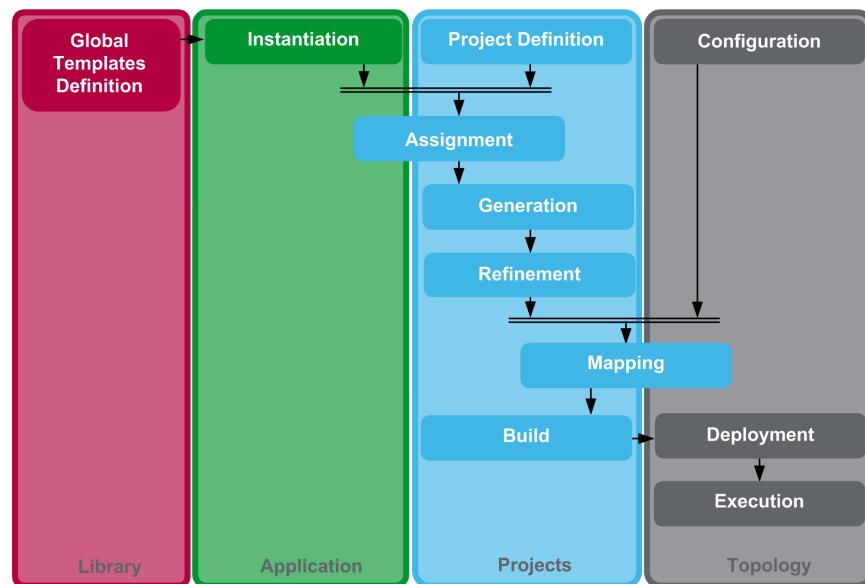
For a description of the representation of a system in EcoStruxure Process Expert, refer to *Object Based Engineering*, page 23.

Graphical Representation

The system engineering life cycle represents the stages of the engineering workflow that you need to complete to engineer a system, page 124 from start to end by using the software.

NOTE: Starting with version 2021, a simplified system engineering life cycle, page 57 exists, which lets you perform changes and maintenance on an existing system faster.

The following figure shows the main system engineering life cycle.



The stages are grouped into 4 main parts where Topology, Projects, and Application represent the components of a system. Library represents the Global Templates, which are used in the other three parts. To each part corresponds an explorer, which provides the functionality to complete each of the stages.

The arrows represent the order in which you need to complete the stages. The horizontal double bars represent the stage gates where the previous stages have to be completed before you can proceed further in the life cycle.

The **Assignment** and **Mapping** stages make the link between the different components of the system.

You can find a list of the stages and the main tasks that you can complete in each stage in quick reference Guides, page 123.

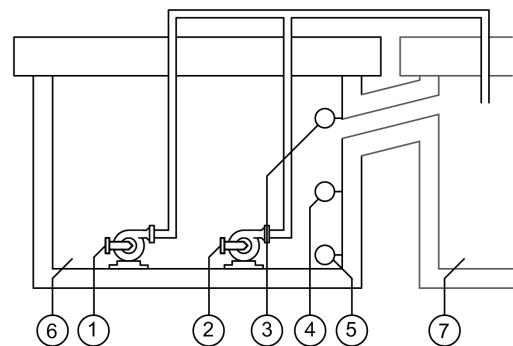
Illustrating the Concept

When necessary, to illustrate the concept of a stage, a concrete example is used.

The example is a water tank system that collects rain water. The installation consists of the following equipment:

- 2 electrical pumps, which transfer the water from the tank into a separate reservoir.
- 3 digital switches, which indicate the rain water level in the tank.

The following figure illustrates the water tank system:



- 1** Pump: Pmp1
- 2** Pump: Pmp2
- 3** High-level switch: Lsh1
- 4** Low-level switch: Lsl1
- 5** Very low-level switch: Lsll1
- 6** Rain water tank
- 7** Reservoir

Pump Management Operating Mode

In the example, 1 of the 2 pumps transfers the rain water into the reservoir. The other pump serves as a back-up pump in case the transfer pump requires maintenance or is out of order. Both pumps exchange their roles periodically and are equipped with a soft starter.

Level Management Operating Mode

In the example, the 3 digital level switches act on the pumps at 3 different water levels: very low, low, and high level. The Lsh1 high level and Lsl1 low-level switches provide information on the level of the water in the tank. The Lsll1 very low-level switch is directly wired to the power supply of both pumps so that they do not run dry.

Platform Components

Overview

The software lets you engineer each part of a system (application, projects, and topology) independently by using specific tools, which provide the necessary functionality.

You can engineer several systems simultaneously.

Platform Managers

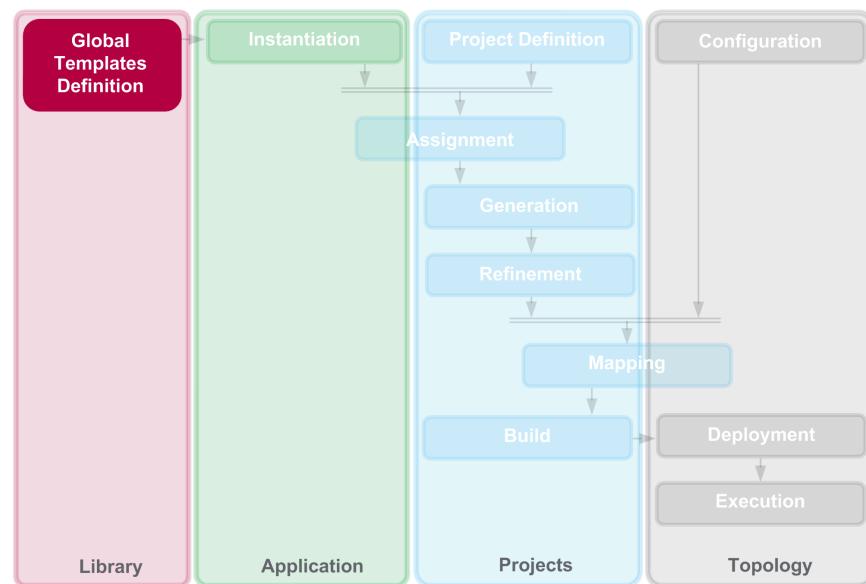
The following explorers are available as part of the software platform.

Explorer	Description
Global Templates Explorer	<p>Displays the Global Templates of the installed libraries and those that you have created and/or imported.</p> <p>They are organized in a folder structure, which makes the templates easily accessible.</p> <p>The explorer lets you search, create, modify (see EcoStruxure™ Process Expert, Global Templates, Reference Manual), and manage Global Templates.</p>
Systems Explorer	<p>Displays the systems that exist in the database of the system server.</p> <p>The explorer is the entry point to start engineering systems.</p>
Application Explorer	<p>Displays the application of a system in a tree view, which lets you create a hierarchy.</p> <p>The explorer lets you search, create, configure, and manage the instances that model the application of a system.</p>
Project Explorer	<p>Displays, at the platform level, the projects of a system that are managed by the software Participants:</p> <ul style="list-style-type: none">• Control projects• Supervision projects <p>The explorer lets you create and manage the Control logic, Supervision data, and the executables of each project.</p>
Topology Explorer	<p>Displays the topology of a system in the form of a folder structure.</p> <p>The explorer lets you create and manage the different topological entities that model the hardware infrastructure. Also, it allows you to interact with the controllers, devices, station nodes, and networks of a system.</p>

Global Templates Definition Stage

Stage Map

The following figure shows the position of the **Global Templates Definition** stage within the system engineering life cycle:



Definition

The Schneider Electric application and topological templates that are provided in various libraries allow you to engineer systems. Yet, you have the possibility to modify or create templates.

The **Global Templates Definition** is the process of creating and/or modifying templates by:

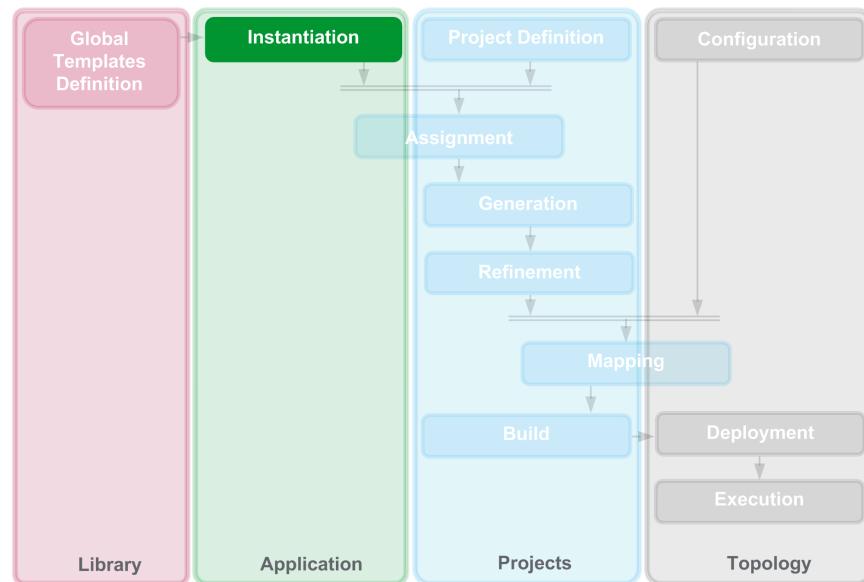
- Using existing templates from the Global Template library and modifying them to meet your particular requirements.
- Creating new templates by using generic base elements with the help of template editors.

This process is described in a separate manual (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

Instantiation Stage

Stage Map

The following figure shows the position of the **Instantiation** stage within the system engineering life cycle.



Definition

The instantiation stage involves modeling:

- The process hierarchy by using a folder structure.
- Functional components by using instances, page 31 of templates.

The resulting folder and instance organization represents the application of the system and is unique.

Once you have created a system, page 132, you can create instances from templates that you select among a choice of templates that are available in the Global Templates library.

Depending on the composition of the template, during the instantiation, the template creates a set of Control and/or Supervision facets, which are associated to the instance and provide specific services.

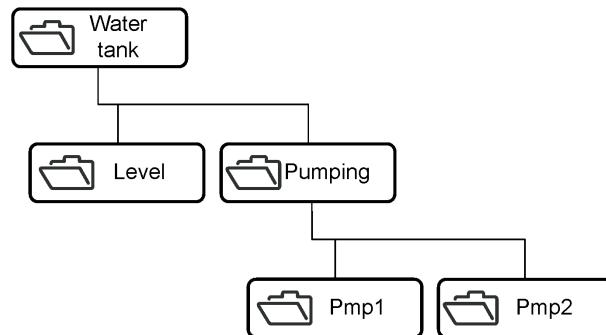
To fulfill the specific requirements of the application, the instantiation process includes the step of configuring the application folders and the instances that they contain by:

- Selecting/deselecting optional Control and Supervision facets among those that are built into the instance template to enable/disable the corresponding services.
- Editing folder and instance properties and/or parameters.
- Creating links between instances.

For more information on completing the instantiation stage, refer to the chapter describing its implementation, page 152.

Example of the Process Hierarchy of the Water Tank System

The following figure shows an example of how you can organize the application of the water tank automation system, page 33.



Water tank: Water tank system root folder.

Level: Level management folder, which will contain the instances modeling the water level sensor functionality.

Pumping: Pump management folder.

Pmp1: Folder containing the instances modeling the functionality of the motor of Pump 1.

Pmp2: Folder containing the instances modeling the functionality of the motor of Pump 2.

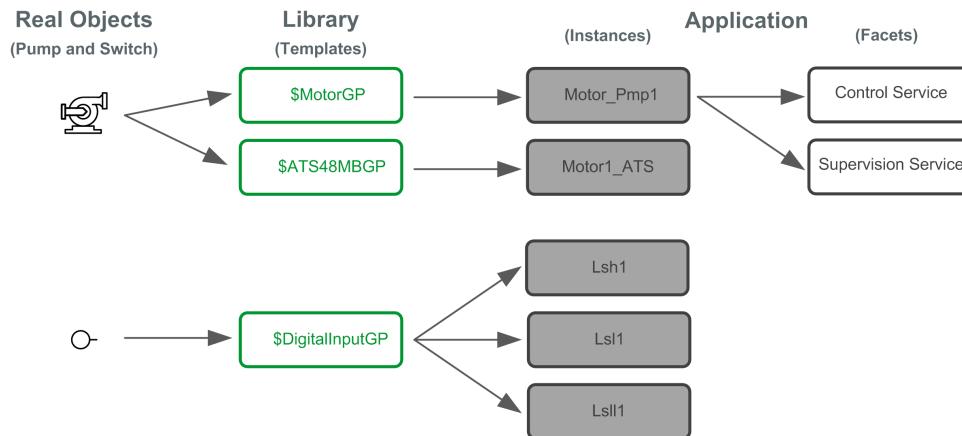
Selecting Templates to Model the Application

The table shows examples of templates of the General Purpose Library that can be used to model the water tank application.

Template	Application
\$MotorGP	Motors of Pump 1 and Pump 2
\$ATS48MBGP	Altistart 48 progressive starters for Pump 1 and Pump 2
\$DigitalInputGP	Digital switches (Lsh1, Lsl1, LslI1)

Example of the Instances Modeling the Water Tank Application

The following figure illustrates the concept of instantiation for the water tank application.



NOTE: For simplicity, only the following items are represented:

- Pump 1 and level switch objects.
- Instances inside folders Pmp1 and Level.
- Facets of the Motor_Pmp1 instance.

The table describes the instances that are created during instantiation of the templates and gives an example of the configuration possibilities.

Selected template	Instances created from template ⁽¹⁾	Instance description	Purpose of the facet contained in the template ⁽²⁾	Configuration example ⁽³⁾
\$MotorGP	Motor_Pmp1	Model the electrical motors of Pump 1 and Pump 2.	Control facet for on/off logic.	The on/off logic facet is selected by default and cannot be deselected. The corresponding service is provided for the instance. You can configure various parameters for this Control service.
			Control facet to manage interlock conditions.	The optional interlock condition summary facet is selected, created, and the service is available for the instance.
			Supervision facet to manage runtime motor on/off data.	The optional Supervision facets are selected, created, and the corresponding services are available for the instance.
			Supervision facet to manage interlock data.	You can configure several parameters for these Supervision services.
\$ATS48MBGP	Motor1_ATS	Model the management of motors for Pump 1 and Pump 2, each by an ATS 48 soft starter device.	Not detailed for the example.	
	Motor2_ATS			
\$DigitalInputGP	Lsh1	Models a digital input for the Lsh1 high-level detection switch.	Control facet to manage the logic.	The logic facet is selected by default and cannot be deselected. The corresponding service is provided for the instance. You can configure several parameters for this Control service.
			Supervision facet for signal data.	The optional Supervision facet is selected, created, and the corresponding service are available for the instance. You can configure several parameters for this Supervision service.
	Lsl1	Models a digital input for the Lsl1 low-level detection switch.	Control facet. (Same as for Lsh1)	The logic facet is selected by default and cannot be deselected. The corresponding service is provided for the instance. You can configure several parameters for this Control service.
			Supervision facet. (Same as for Lsh1)	The optional Supervision facet is selected, created, and the corresponding service is available for the instance. You can configure several parameters for this Supervision service.
	Lsll1	Models a digital input for the Lsll1 low-low-level detection switch.	Same as for Lsh1	Not detailed for the example.

(1) The name of the instance is user-defined.

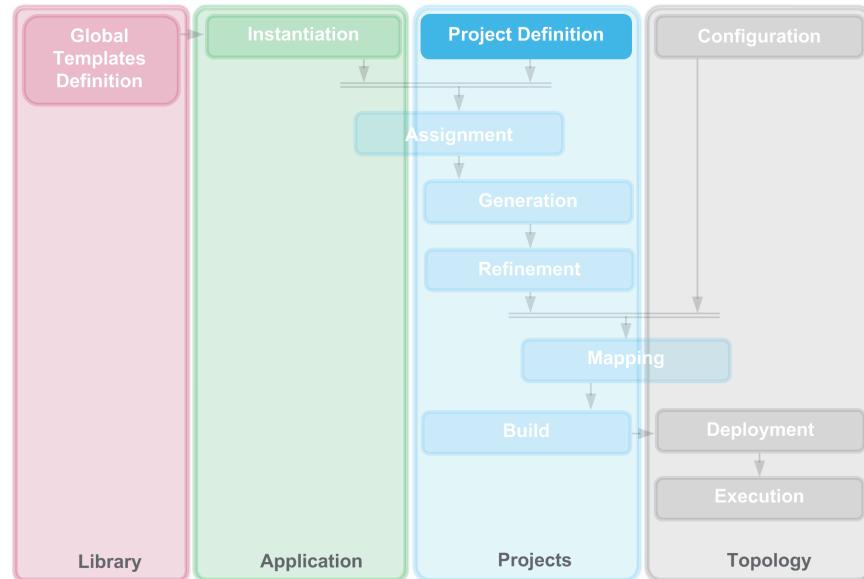
(2) The actual templates contain more facet templates than shown in the table.

- (3) The facets are created for each instance. The example shows only the configuration for instances *Motor_Pmp1* and *Lsh1*. In fact, each instance of a template can be configured individually.

Project Definition Stage

Stage Map

The following figure shows the position of the **project definition** stage within the system engineering life cycle.



Definition

The project definition is the process to define the organizational structure of projects that will be integrated into the system through the Participants. You define projects independently from specific hardware components, which gives you flexibility and allows you to use one project with different hardware infrastructures.

Ultimately, towards the end of the system engineering life cycle, to each project will be associated one or more executable files that you can deploy to the topology of the system.

For each system, you can define as many projects as required.

Each project is associated to a Participant and is therefore referred to as a *Participant project*.

A system is composed of the following Participant projects:

Control	This type of project is specific to a hardware platform (for example, Modicon M580 platform) and manages the controller program.
Supervision	This type of project allows you to manage data from the application, represent it graphically, and provide the operator with a user interface to interact with the controller.

NOTE: Supervision projects are optional and you create them depending on the requirements of the system.

Project Structure

The organizational structure of a project is specific to the Participant that it is related to because it reflects the components that the Participant manages.

A Control project consists mainly of nodes representing:

- The program structure: A **Containers** node containing FBD and non-FBD program sections, which are referred to generically as containers.
- The binary program: An **Executables** node providing access to program files and related services.

A Supervision project consists mainly of nodes representing:

- Data: A **Containers** node containing tag containers for each cluster.
- Graphic elements: A **Pages** node that contains the user interface visible to the operator.
- Components: A **Cluster** node for managing servers and communication for data acquisition and representation by cluster.
- The program: An **Executables** node providing access to services related to the program files.

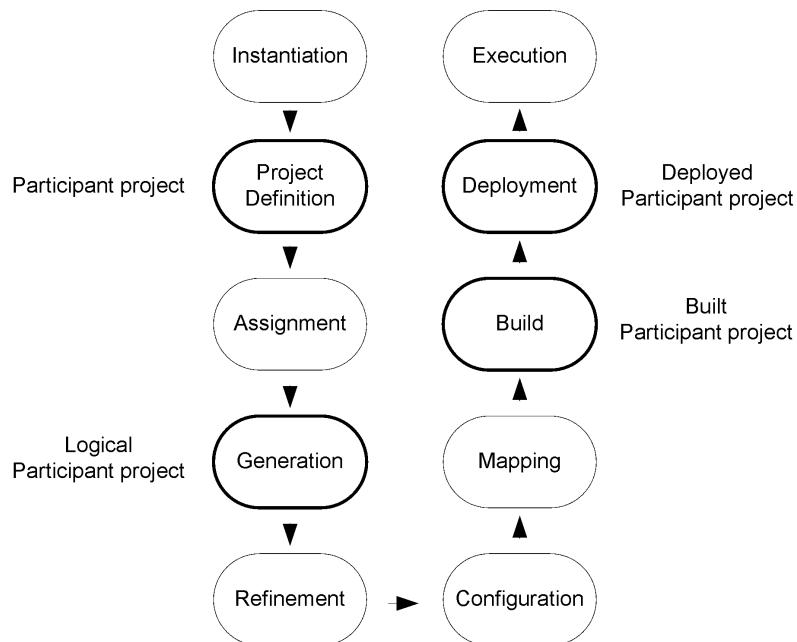
Types of Projects

As you progress through the system engineering life cycle, a Participant project evolves. To distinguish between the stages that projects go through, specific terminology is used to describe them.

The software manages three types of Participant projects throughout the system engineering life cycle:

Logical Participant projects	Contain the information of the application and the organizational structure of the project with no relation to the hardware infrastructure of the system. The software creates these projects during the generation , page 46 stage.
Built Participant projects	Contain the information of the logical Participant project and associate it to the hardware infrastructure of the system. The software creates these projects during the build , page 53 stage.
Deployed Participant projects	Are the built Participant projects that you have deployed to engines of the system (for example, controllers or station nodes). You deploy projects during the deployment stage, page 55.

The following figure illustrates at which stages of the system engineering life cycle, the different types of projects are managed.



For more information on completing the project definition stage, refer to the chapter describing its implementation, page 254.

Example

The following projects are defined for the water tank system, page 33:

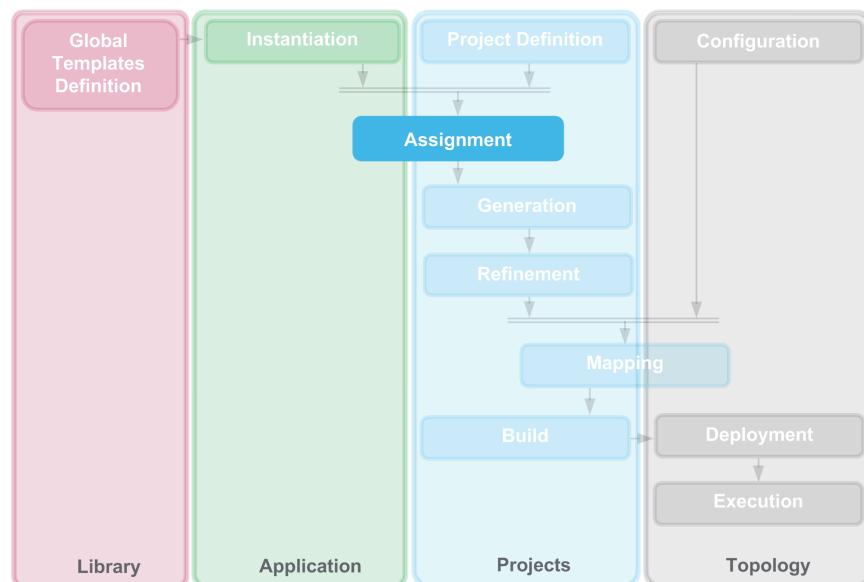
- A Control project to manage the logic (for example, variables and DFBs,) of the instances of the application. It contains three containers in the MAST task folder corresponding to the folder structure of the application:
 - Level
 - Pmp1

- Pmp2
- A Supervision project to manage the Supervision services (tags, alarms, supervision pages) of the instances of the application. It contains the following components:
 - One tag container for the Supervision data facets of the instances of the application.
 - One page for the Supervision genie facets of the instances of the application.
 - Components: One of each I/O, alarm, and trend server, and I/O device with its communication port.

Assignment Stage

Stage Map

The following figure shows the position of the **assignment** stage within the system engineering life cycle.



Definition

The assignment is the process of associating Control and Supervision facets of instances of the application to a container of the corresponding project of the system.

The purpose is to define where the constituents that are encapsulated inside the facets will be generated in the Participant project during subsequent stages of the system engineering life cycle.

You need to assign the two types of facets to the corresponding containers:

- Control facets: To containers of Control projects, which are FBD sections.
- Supervision data facets: To containers of Supervision projects, which are tag containers.

You can assign a facet only to one container per project but you can assign a facet to several projects.

NOTE: Supervision genie facets are assigned to a page of a Supervision project during the refinement stage.

For more information on completing the assignment stage, refer to the chapter describing its implementation, page 328.

Example

The table shows an example of assignment of the facets of the instances *Motor_Pmp1* and *Lsh1* of the water tank system, page 33 that you can perform.

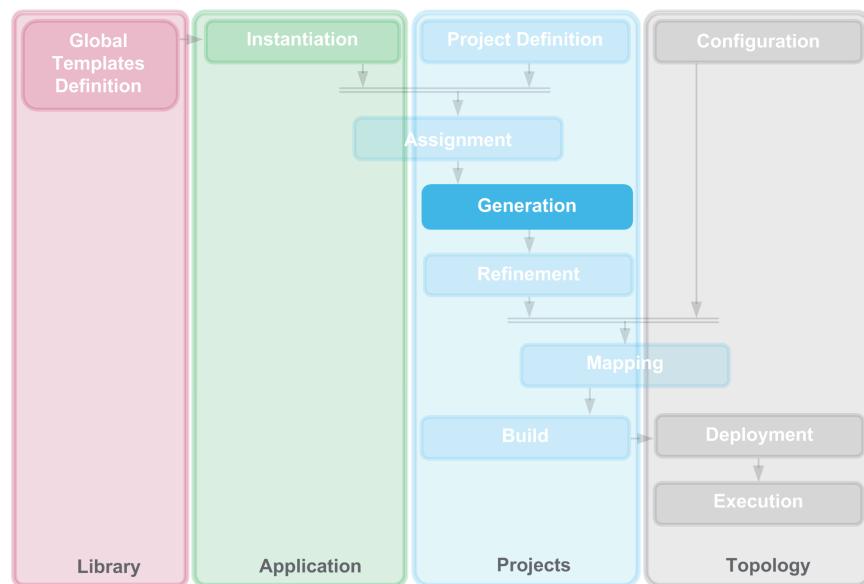
Instance	Facets	Assigned To
<i>Motor_Pmp1</i>	Control facets: <ul style="list-style-type: none"> • \$MOTORGP_UL: On/off logic data facet • \$INTERLOCKONGP_UL: Optional interlock condition summary facet 	FBD section Pmp1 of the Control project.
	Supervision facet:	Tag container of the Supervision project.

Instance	Facets	Assigned To
	<ul style="list-style-type: none">• \$MOTORGP_CD: Optional on/off data facet	
<i>Lsh1</i>	Control facets: <ul style="list-style-type: none">• \$MOTORGP_UL: Logic data facet• \$DISignal_UL: Optional digital input signal	FBD section Level of the Control project.
	Supervision facet: <ul style="list-style-type: none">• \$DINPUTGP_CD: Optional data facet	Tag container of the Supervision project.

Generation Stage

Stage Map

The following figure shows the position of the **generation** stage within the system engineering life cycle.



Definition

The generation is an automatic process that creates a logical Participant project by using the constituents of the assigned facets and taking into account the structure of the project. At this point, the logical Participant project has no relation with the topology of the system.

For more information on completing the generation stage, refer to the chapter describing its implementation, page 383.

Example

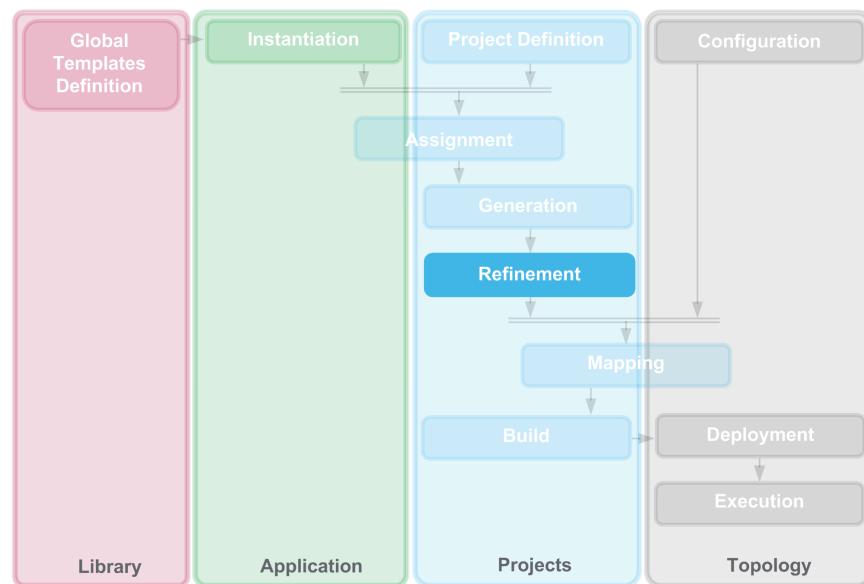
During the generation process of the facets of the water tank system, page 33 that are assigned to projects, the software creates:

- For Control: The program of the Participant.
- For Supervision: populates the different database files that are used by the Participant.

Refinement Stage

Stage Map

The following figure shows the position of the **refinement** stage within the system engineering life cycle.



Definition

Refinement is an optional process during which you can modify and/or add information to a logical Participant project that the software has created during the generation stage.

For example, you can refine a Control project to:

- Add logic between control facets of instances.
- Create new relationships with other constituents such as interlocks or control sequences.
- Add sections in languages supported by the Participant other than FBD.

To facilitate the refinement of Control projects, the software locks and identifies visually code that it has generated.

You can refine a Supervision project, for example, to:

- Assign genie facets to a Supervision page.
- Configure user privileges.

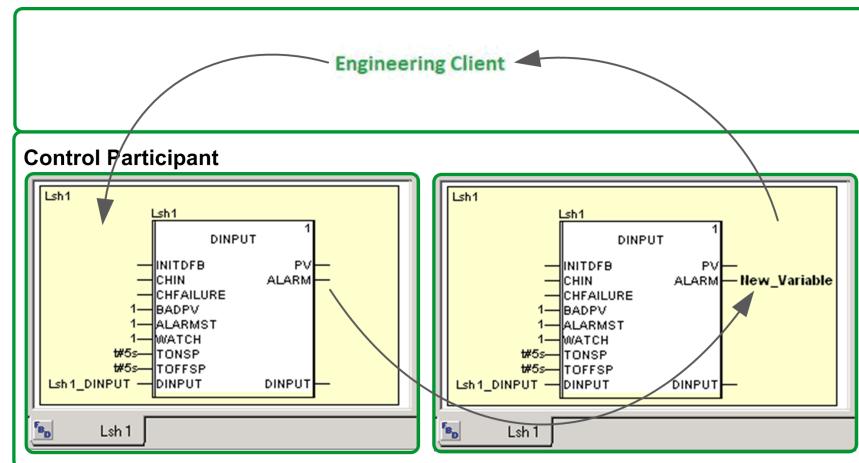
The software opens the respective Participant allowing you to use the interface and some functionalities of the embedded software.

For more information on completing the refinement stage, refer to the chapter describing its implementation, page 434.

Example

During the refinement of the Control project of the water tank system, page 33, you can access and modify the control logic using the embedded editors to create or modify constituents.

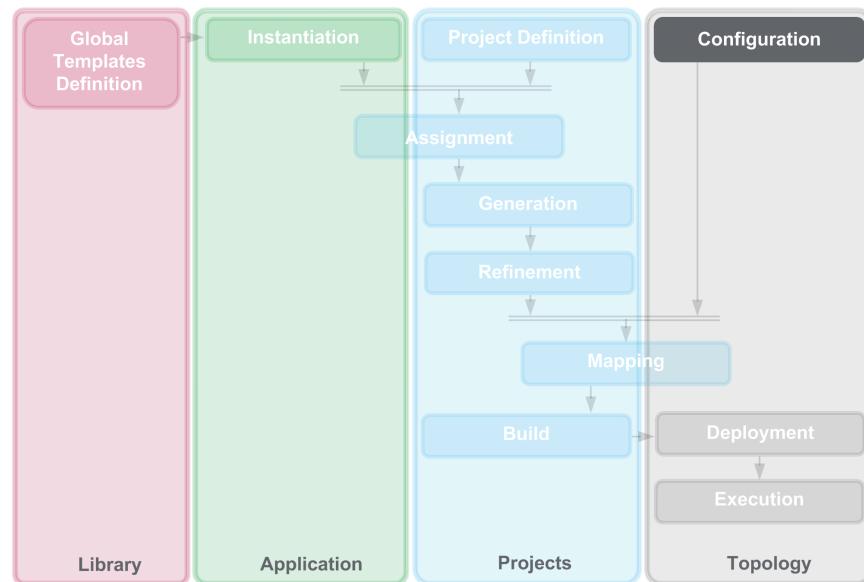
The following figure illustrates the refinement of the constituents generated by the Control logic facet of the *Lsh1* high-level switch sensor instance. During this refinement, a new variable is added.



Configuration Stage

Stage Map

The following figure shows the position of the **configuration** stage within the system engineering life cycle.



Definition

The configuration stage consists in defining the topology of the system, which models the hardware and software infrastructure.

You can create topological entities by using either of two methods depending on the entity that you are defining:

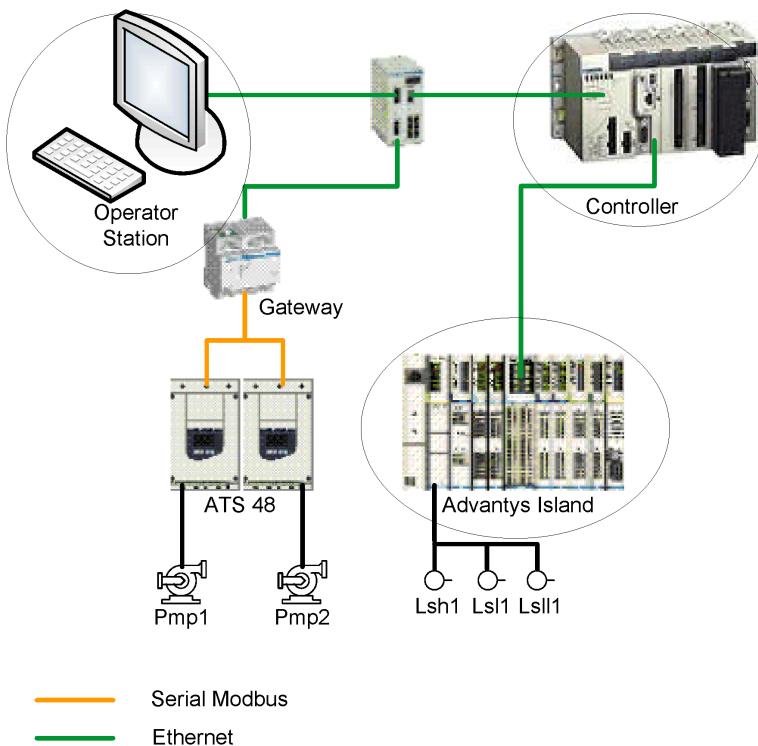
- In the engineering client user interface by selecting manually the templates to be used to create entities.
- In the Control Participant by using the functionality of the embedded software. Saving the configuration in the Participant creates the corresponding folders and devices in the **Topology Explorer** by selecting automatically the appropriate templates of the Foundation library.

For more information on completing the configuration stage, refer to the chapter describing its implementation, page 498.

Example

For the water tank system, page 33, during the configuration stage, you can define the hardware that will run the program, the Ethernet control network, and the operator station from which the operator will operate the pump, monitor the pumping process, and the water level.

The following figure shows examples of topological entities of the water tank system that you can create during the configuration stage.



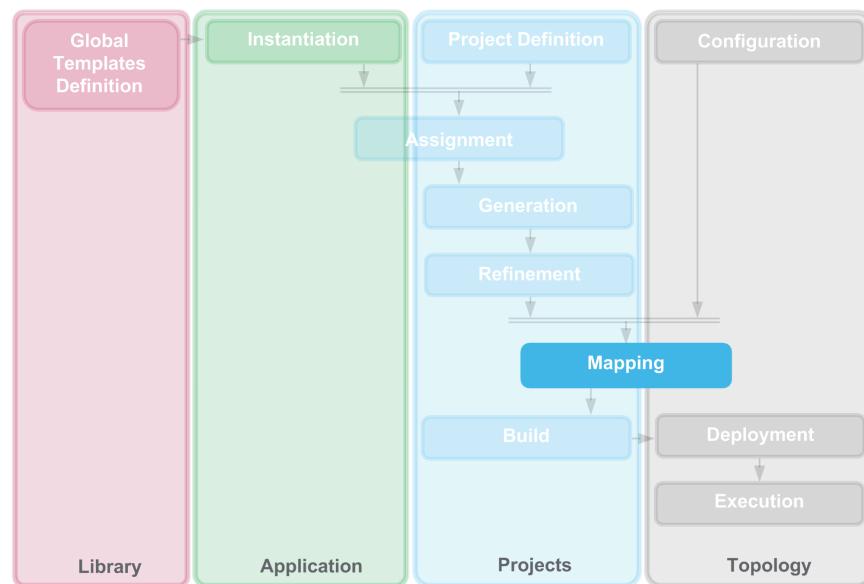
Controller	Local rack with an Ethernet based architecture.
STB Island	Distributed I/Os to which level switches Lsh1, Lsl1, and LslI1 are connected, communicating with the controller through I/O scanning.
Operator station	Supervision station to control and monitor the system, configured with its network interface card and the supervision software.
Ethernet network	Networks connecting the controller, the STB Island, the gateway, and the station node.
Gateway	The gateway communication instance is created in the application of the system modeling the communication functions. The device itself is not represented in the topology.
ATS 48	The soft starter instances are created in the application of the system modeling the motor management functions. The device itself is not represented in the topology.

NOTE: When communication with devices is performed by using Modbus TCP explicit or implicit messaging, topological instances of devices are created also in the topology to manage TCP/IP addressing.

Mapping Stage

Stage Map

The following figure shows the position of the **mapping** stage within the system engineering life cycle.



Definition

The mapping stage consists in associating facets of a project to resources of the topology of the system.

You can perform three types of mappings independently for each project:

- *Service mapping*: Defines which engine executes the project (for example, which controller runs the program or, which station node hosts the I/O server).
- *Hardware mapping*: For Control projects only. Links the logical projection of the hardware defined in the application (for example, a variable representing an I/O signal of a motor) to the hardware defined in the topology (for example, an I/O channel in an I/O module of the controller).
- *Communication mapping*: Defines runtime communication (for example, which controller will communicate with the I/O device and through which communication module), communication between controllers (peer to peer communication), and between controllers and devices (for example, communication with STB islands or Modbus TCP Ethernet devices by using I/O scanning).

NOTE: You can map a project to several different engines, for example:

- To a controller in the laboratory for test purposes.
- To a controller in the production plant.

For more information on completing the mapping stage, refer to the chapter describing its implementation, page 576.

Example

The table gives examples of the mappings that you can perform for the water tank system, page 33.

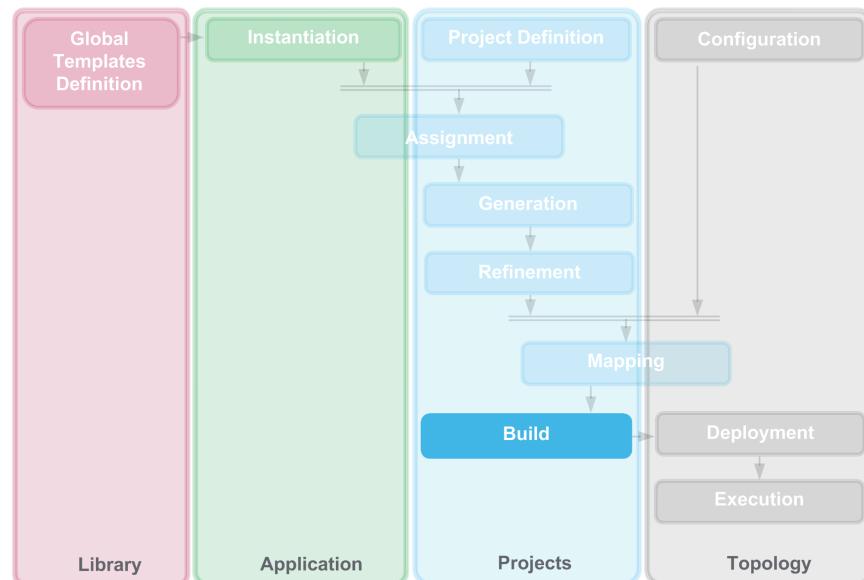
Mapping	Description
Control project	
Service	Defines to controller the program is associated for deployment and execution.

Mapping	Description
Hardware	If for instance <i>Motor_Pmp1</i> , you select the optional Running Control service, the software creates the corresponding facet. It creates, for example, a boolean variable that indicates the state of the motor switch. You can map this variable of the application to the channel of an input module of the STB island.
Communication	Defines the I/O scanner channels that are created for communication between the controller and the STB island.
Supervision project	
Service	Defines the I/O, alarm, and trend servers that are used for the Supervision project and on which operation stations of the topology they are installed.
Communication	Defines from which controller I/O data is acquired and which communication module of this controller is used.

Build Stage

Stage Map

The following figure shows the position of the **build** stage within the system engineering life cycle.



Definition

The build stage consists in starting an automated process during which a Participant merges the following information in order to create the built Participant project:

- The logical Participant project
- Mappings
- The topology

The built Participant project is the executable that you can deploy to engines of the system

The software stores the executable files resulting of this process in the contents repository.

At this point, you can view the complete Control and Supervision projects in the corresponding Participant.

Once built, a project is referred to as the built Participant project, which is the project associated to an engine for execution.

For more information on completing the build stage, refer to the chapter describing its implementation, page 628.

Example

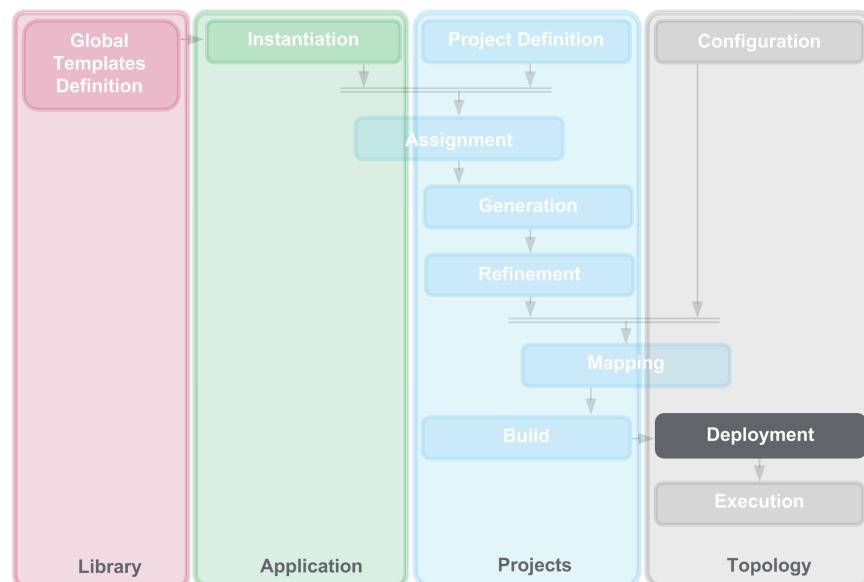
During the build process of the water tank system, page 33, Participants create the following files.

Participant	Generated Files
Control	The executable file, which is the binary to be deployed to the controller.
Supervision	The master project file (.ctz) to be deployed to the operation server and operator stations.
	Files of the included projects (.ctz) to be deployed to the operation server and operator stations.
	The OFSOPC/OPC UA configuration files (.xml) to be deployed to the operation server.

Deployment Stage

Stage Map

The following figure shows the position of the **deployment** stage within the system engineering life cycle.



Definition

During the deployment stage, you transfer a built Participant project to the corresponding engine that is defined in the topology of the system.

Depending on the project, an engine can be:

- For Control:
 - a controller
 - a station node for emulating a simulator
- For Supervision: A station node acting as operation server or operator station.

Once deployed, a project is referred to as the deployed Participant project, which is the project that is loaded into the engine for execution.

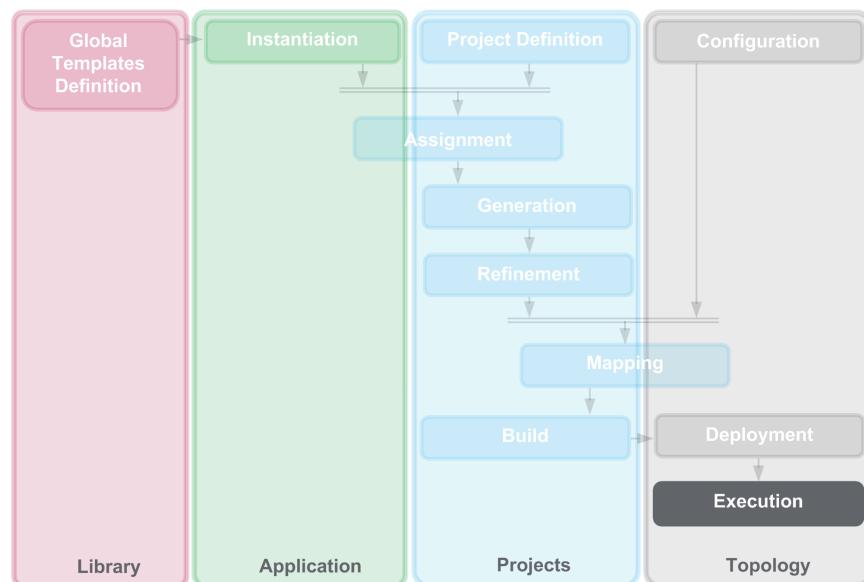
The stage also includes the steps of redeploying an already deployed Participant project and to update a deployed Participant project with changes that you have made to the corresponding logical Participant project.

For more information on completing the deployment stage, refer to the chapter describing its implementation, page 648.

Execution Stage

Stage Map

The following figure shows the position of the **execution** stage within the system engineering life cycle.



Definition

The execution stage includes the steps to:

- Start and stop the execution of a deployed Participant project.
- Make online changes to a deployed Participant project.
- Update the corresponding logical Participant project with the changes that you have made online.

The execution stage takes into consideration actions that you can perform with and without stopping the controller to keep you informed of the impact of your changes.

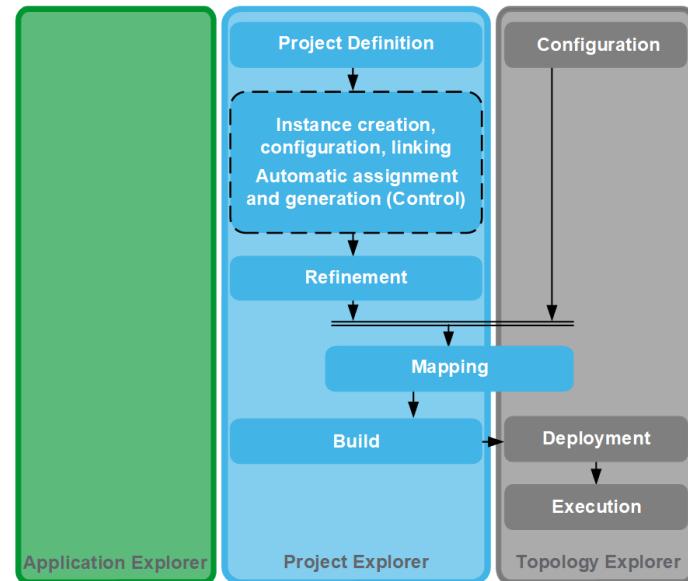
For more information on completing the execution stage, refer to the chapter describing its implementation, page 706.

Additional runtime services are described in the **Navigation Services User Guide** (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).

Simplified System Engineering Life Cycle

Graphical Representation

The following figure shows the simplified system engineering life cycle. The instantiation, assignment, and generation stages are grouped into one, which is completed by using the **Project Explorer** and the **Assignment Editor** for Control projects. The change compared to the main system engineering life cycle is outlined with a dashed line.



NOTE: The Global Templates definition stage is not part of the simplified system engineering life cycle.

Definition

The purpose of the simplified workflow is to make it possible to perform small changes to the Control project of a system during the maintenance phase with little effort. Such changes can be, for example, adding one or more I/O signals.

As part of the simplified workflow, you can perform the following actions from a Control project:

- Create, page 335, configure, copy, and link, page 377 one or more instances and assign their facets automatically.
- Generate changes automatically, page 394.

Working with the Software

What's in This Part

Starting and Quitting System Server and Clients	59
Multiuser Functionality	70
Action Traceability	72
Software Capabilities	82
Using the Help	92
User Interface	100

Overview

This part explains how to access and use the client user interface (UI). It describes the various functionalities of the software and provides general information to help you use them effectively.

Starting and Quitting System Server and Clients

What's in This Chapter

Logging In and Out of the System Server and Clients	60
Locking and Unlocking the Software	62
Starting the System Server Again	63
Client/Server Connection	64
Quitting the Software	68

Overview

This chapter describes how to start and quit the system server and clients after a first start, and describes the client/server connection.

Logging In and Out of the System Server and Clients

Overview

Any user who is a member of at least one EcoStruxure Process Expert group can log in to the system server or a client.

These groups are created automatically or must be created manually by the system/network administrator depending on the *Authentication Type* setting that was selected during installation. The administrator must add users to one or more groups depending on the role of each user.

To be able to use the functionality of a given component, you must be a member of the appropriate group (for example, to use the engineering client, you must be a member of the *ESX HPCS Engineer* group).

NOTE: For information on how to get started with the software, refer to the topic describing the first steps to perform (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Authentication Domain Selection

When you log in, enter your username without domain name or prefix.

The system server uses the *Authentication Type* setting (see EcoStruxure™ Process Expert, Installation and Configuration Guide) to validate your credentials either locally or with the Active Directory® domain service.

NOTE: If *Domain* is the configured *Authentication Type*, the domain controller must be reachable from the computer running the system server.

NOTE: If the computer that runs the client is connected to a Windows® domain and the connection to the domain network is interrupted, users can log in locally by changing *Authentication Type* to *local*.

For more information, refer to the topic describing role based access control (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Single Login and Logout

The software implements a single-login mechanism. You need to log in when you open the first software component (for example, a client). Thereafter, you are logged in automatically when you open other components on the same computer.

If several components are open on the same computer, you are logged in to all of them at once. You need to log in again only after you log out of a component on the computer.

The software also implements a single-logout mechanism. Logging out of a software component also logs you out of the other software components to which you are logged in on the same computer.

NOTE: Only the system server can be running without a user being logged in when it has been started automatically (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Logging In

To log in to a software component, enter your username (without domain) and password in the log-in window. The log-in process may take a moment to complete.

The window opens when you start the first component on a computer, open a second component while no user is logged in to the first component, or by clicking **Log In** from the user menu, page 101.

If you close the log-in window without logging in, the component remains open.

NOTE: Logging in to a client is subject to the same log-on policies as those that are defined locally for the computer or for the domain.

Logging Out

To log out of a software component, select the **Log Out** command from the user menu, page 101.

To be able to log out, all operations that are in progress for components running on the computer must have completed.

Logging out of a software component:

- Does not quit the component.
 - The system server is not stopped if it is running. Also, you cannot stop the system server when no user is logged in.
 - For clients, if you had opened additional windows by using the **Open in New Window** command, page 108, these windows are closed. Tabs are closed.
- Logs you also out of the other software components to which you are logged in on the same computer.

NOTE: You are logged out of the software on a computer when you quit the last component to which you were logged in on this computer.

Viewing Connected Clients

From any client that is connected to a system server, you can view information about the clients that are connected to this server in the **Client Details** tab of the about dialog box.

For a description of the dialog box, refer to the topic describing the **About EcoStruxure Process Expert** dialog box, page 104.

Tracing Login Attempts and Log Outs

The following actions are recorded:

- Attempts to log in to a component even if they are unsuccessful. The username that is entered is recorded.
- Locking and unlocking, page 62 a component.
- Logging out of a component.

Actions performed in the system server console appear as notifications in the console and are recorded in the system server log files.

Actions performed in a client appear in the notification panel and in the system server console).

For details, refer to the topic describing the traceability of actions, page 72.

Locking and Unlocking the Software

Locking Software Components

To lock a software component, select the **Lock** command from the user menu, page 101.

Locking the server still allows clients running on other computers to connect or remain connected.

When components are locked, the log-in window opens on top of the main window. The username of the logged-in user shown is in the **User Name** field in read-only mode. Only the user who has locked the software can unlock it.

Impacts of Locking Software Components

Locking a software component has the following impacts:

- Locking any software component also locks all the other components that are open on the computer and to which you are logged in.
- Dialog boxes and notifications in the notification panel are queued and suppressed. These messages are shown once the component is unlocked.
- You cannot quit a component that is locked.
- You cannot open other EcoStruxure Process Expert components on the computer.

Unlocking Software Components

To unlock a component, enter the password in the login dialog box.

Unlocking any software component also unlocks all the other components that were locked on the computer.

To unlock the software when *Domain* is the configured *Authentication Type* requires that the domain controller is reachable from the computer running the system server.

Starting the System Server Again

Temporary Content Repository Files

When you start the system server, it creates folders to store temporary files, page 940 from the content repository to speed up access to these files.

Using the Auto-Start Functionality

You can configure the system server to start automatically (see EcoStruxure™ Process Expert, Installation and Configuration Guide) when you restart the computer.

Starting the System Server Manually

To start the system server manually after you stopped it, page 68, proceed as follows.

Step	Action
1	Verify that Caché is running by checking the presence and the color of the Caché icon in the notification area. When Caché is running, its icon is blue.
2	If the system server tray icon is not visible, from the Windows® Start menu, click EcoStruxure Process Expert > System Server . Result: The log-in window opens.
3	Enter your username without domain or prefix, your password, and click Log In . Result: The system server console opens.
4	Click Action > Start . Result: The system server starts. NOTE: During the system server start, several Participant windows may open briefly. Do not close them manually; otherwise, the functionality of the software may be impacted.
5	Wait until the system server tray icon turns green or the console displays the following message at the bottom of the window to start engineering and operation clients (see EcoStruxure™ Process Expert, Installation and Configuration Guide): Server is ready.

Client/Server Connection

Overview

Clients connect with the system server through an Ethernet cable connection, for example to:

- Access the database.
- Access software Participants.
- Launch tasks that are performed by the system server.
- Receive notifications from the system server.
- Verify digital certificates and license rights.

You cannot use clients while the connection to the system server is interrupted or while the system server is not running.

You can test the client/server connection (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

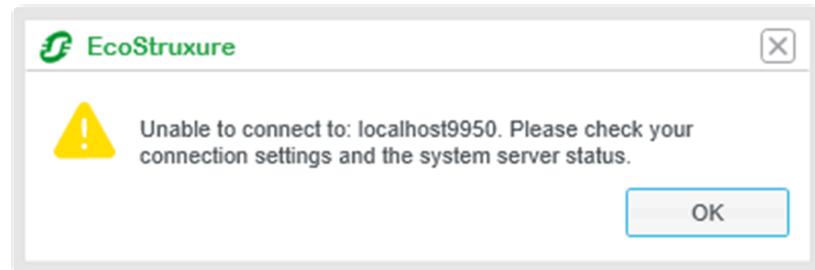
Notifications About Client/Server Connection Interruption

The table describes the various notifications that appear client-side for different scenarios where a client cannot connect to the system server or a software Participant.

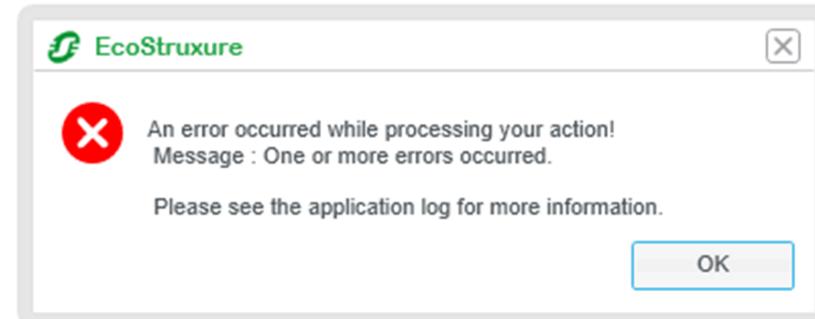
Scenario	Operation in progress at the time of the interruption	Client notification	Corrective action
1	No	No	None
2	You launch a task while the connection to the system server is interrupted.	Notification that the client is unable to connect to the system server.	Launch the task again when the connection is re-established.
3	Yes	Notification about an unexpected communication interruption.	If the system server is still running, the action completes server side and the notification of clients is updated when the connection is re-established. If the system server has stopped, an action is required, page 66.
4	Yes. An action is performed in an open software Participant window.	Notification that the action in the Participant cannot be performed and indication of the possible causes. NOTE: If you are refining a Control project online (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) and have clicked the button to build and deploy changes after the connection was interrupted, the changes are not deployed nor saved.	If the connection to the system server is still functional, save changes, close the Participant window, and start over once the problem is fixed. If the connection to the system server is interrupted, refer to scenario 3. If the reason for the connection interruption cannot be determined, you need to contact Schneider Electric support, page 104.
5	The system server is stopped by user intervention.	Notification in the notification panel, page 73 to inform that the server is shutting down.	From this point on, you cannot perform any action on the client. Restart the system server.
6	Yes. An action is performed in an open software Participant window.	None. A message appears in the system server console.	Contact Schneider Electric support, page 104.

NOTE: Refer also to the topic providing information to help troubleshoot client/server communication (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

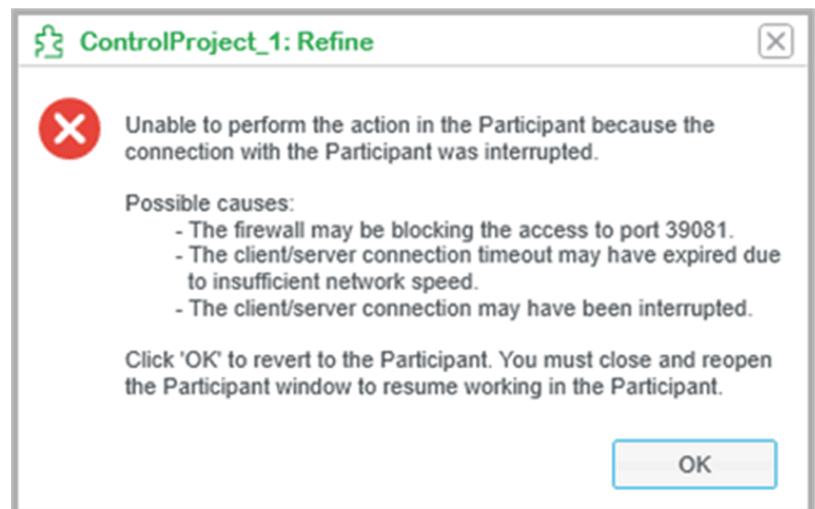
The following figure shows an example of the notification that appears for scenario 2. The name or IP address and the port of the system server computer are indicated.



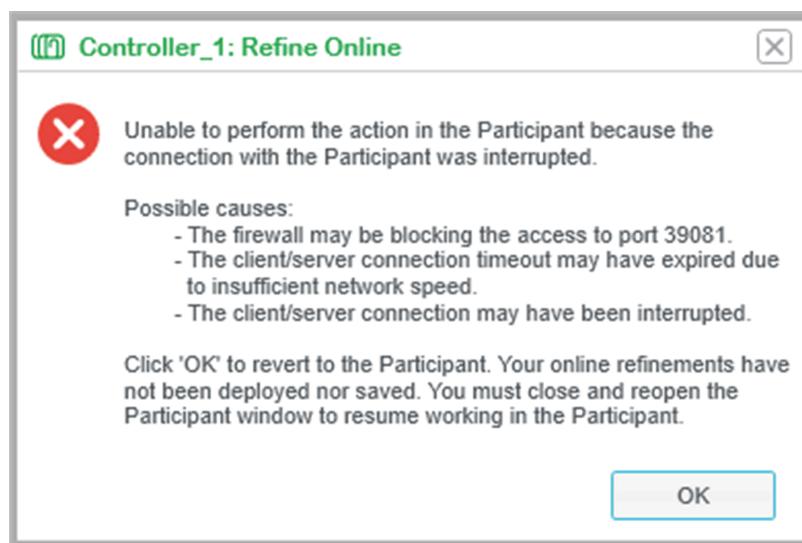
The following figure shows an example of the notification that appears for scenario 3.



The following figure shows an example of the notification that appears for scenario 4.

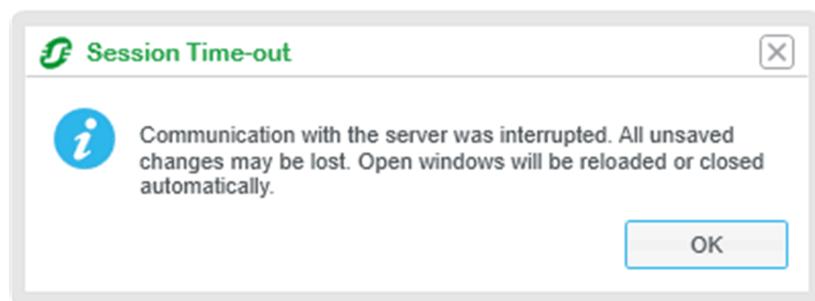


The following figure shows an example of the notification that appears for scenario 4 in case you are refining a Control project online.



After Re-establishing the Connection to the System Server

Clients automatically reconnect to the system server when the connection is re-established or when the system server is running again. After reconnecting, clients display a notification to inform you of the communication interruption. Unsaved changes are discarded and open windows are reloaded or closed automatically.



If the system server has stopped, the notification panel is empty when the system server restarts. To view the history of activity up to the interruption of the communication, open the log files, page 80.

After an unexpected interruption of the client/server connection, the states of both the engineering client and the system server may have become inconsistent. This can be the case, for example, if at the time of the interruption an operation was in progress such as an import of several Participant projects or pasting of many instances.

NOTICE

LOSS OF COMMUNICATION

After an interruption of the client/server connection, verify that the last operation executed by the software was completed successfully, and if necessary, repeat the last operation.

Failure to follow these instructions can result in unresponsive clients.

You are able to resume using the software when the connection to the system server is re-established or when it is running again.

Restarting Clients After Closing Unexpectedly

If an engineering client closes unexpectedly (for example, because the computer on which it is running stopped working), you do not need to restart the system server. You can restart the client after 60 seconds. This is the time that is required by the system server to terminate a client session, which ended abnormally.

Restarting a client earlier may trigger messages about missing engineering client license, for example.

Quitting the Software

Overview

To quit the software, quit components in the following order:

- Any clients that are connected to the system server that you are stopping.
- The system server.

You cannot quit a component if it is locked, page 62. Also, you cannot stop the system server when no user is logged in.

To restart the system server, quit the software in the same order and then start the system server, page 63.

NOTE: You are logged out of the software on a computer when you quit the last component to which you were logged in on this computer.

Quitting Clients

When you quit a client, it does not close if any task is still in progress.

To quit a client, proceed as follows.

Step	Action
1	Verify that any processes are finished executing by checking the information in the notification panel.
2	Click the close button of the client window. Result: The client window closes, disconnecting it from the system server.

Stopping the System Server

When you stop the system server, tasks in progress are completed before the server is stopped. If any client is still connected to the system server, a notification is displayed, which indicates the identity of the logged-in user by using the format *full name (computer name)*.

You cannot use a client when the communication between the system server and the client is interrupted, for example, when the system server is restarted or not running.

NOTICE

LOSS OF COMMUNICATION

Stop the system server only when no client is running.

Failure to follow these instructions can result in unresponsive clients.

To stop the system server, proceed as follows.

Step	Action
1	From the menu bar of the system server console, click Actions > Stop . Result: A confirmation dialog box opens. NOTE: If a client is still running, the full name of the user who is still logged in is displayed in the dialog box.
2	If a client is still connected, click No ; otherwise proceed to step 4. Result: The dialog box closes without stopping the system server.

Step	Action
3	Quit the client, page 68 and stop the system server again by starting over this procedure.
4	When no more client is connected, click Yes . Result: The system server completes tasks in progress and then stops. NOTE: Click No to close the dialog box without stopping the system server.

Multiuser Functionality

What's in This Chapter

Multiuser Functionality	70
-------------------------------	----

Multiuser Functionality

Overview

Several clients can connect to a system server simultaneously, allowing several users to work on the same component or different components of a system at the same time.

For example, User A and User B can work on the application of the system, or User A can work on the application and User B on the projects of the system. These operations could also be performed by a single user.

In both cases, the software manages system-wide concurrent user access to editors and other tools to preserve the integrity of data of the system.

It may allow you to view data by granting access to certain editors in read-only mode.

When a user is finished working on a system component (for example, an instance) by closing the corresponding tool (for example, the **Instance Editor**), the software allows other users to access components that manage or use data of that component.

As such, users do not need to check out or lock the component that they are working on.

Managing User Activity

At the following stages of the system engineering life cycle, the software indicates the status of objects by using icons:

- Instantiation, page 192
- Project Definition, page 326

At any stage, when the command that you have selected cannot be completed, you can obtain information about the reason through:

- Dialog boxes that clients display when you try to access system data that is being used or edited by another user.
- Information in the notification panel on the activity of any user that is logged in to a client of the same system server.

Examples

By using examples, the table illustrates some of the locking mechanisms that the software implements when two users work on the same system.

Action executed by User A	Examples of restrictions for User B
Editing properties of <i>Instance A</i> .	Cannot edit <i>Instance A</i> but can open the Instance Editor in read-only mode.
	Can generate the section to which facets of <i>Instance A</i> are assigned but the facets of <i>Instance A</i> are not generated. If the section contains other facets, they are generated. Information is displayed in the notification panel.
Refining the Control Participant project in the Project Explorer .	Cannot refine the same Participant project or any section of it. Information is displayed in a dialog box.

Action executed by User A	Examples of restrictions for User B
	Cannot generate the same Participant project.
	Cannot build the same Participant project.
Configuring a controller entity in the Topology Explorer .	Cannot configure the same controller entity. Cannot do mapping (hardware and communication) of the Control executable to which the engine is mapped. Cannot build the Control Participant project executable to which the engine is mapped.

NOTE: The locking mechanisms apply also when a single user is performing these operations (if applicable).

Action Traceability

What's in This Chapter

Notification Panel	73
Audit Trail	76
Activity Log Files	80

Overview

This chapter describes how to obtain information related to the activity of the system server and clients.

Notification Panel

Overview

The notification panel allows you to view and trace engineering activity. The panel is displayed automatically when you start an engineering client. You can modify its height and show column headers.

The information that is displayed is the same for each engineering client that connects to the same system server. When you open a second engineering client, you can see in its notification panel the history of actions starting from the moment the first engineering client was opened.

Certain system server tasks are displayed also.

An entry is logged when you launch a task and it is updated as the execution of the task progresses. When the task is complete, the status of the task and the actions that the software has executed are indicated.

For each task, a color icon allows you to see the status at a glance.

At any time, the notification The information remains available while the system server to which the client is connected is running. Closing the client or the notification panel does not delete the entries.

You can show/hide the notification panel from the engineering client toolbar, page 101.

Information Availability

At any time, the notification panel can display a maximum of 2000 entries. When this limit is reached, the most recent entries replace the earliest ones on a first-in first-out basis.

The information remains displayed while the system server to which the client is connected is running. Closing the client or the notification panel does not delete the entries.

Aborting Tasks

When you select or confirm certain commands, the  icon appears in the **Action** column of the notification panel. You can click it to cancel the task. Once the task is near completion and cannot be canceled anymore, the icon appears disabled



. Only the user who has initiated the command is allowed to abort it.

The possibility to abort and the behavior when you click the icon is described in the topics documenting the commands for which the abort functionality is available. For example, the **Paste** command, page 207 of the **Application Explorer**.

Aborted tasks are identified in the notification panel in the **Message** column with *(Aborted)*. When you export data of the notification panel, it is the most recent subtask of the process that is identified with *Aborted* in the **Action** column, not the task itself.

Data Description

The table describes the data that is displayed in the notification panel.

Column header ⁽¹⁾	Description
–	Icons indicate the status of the task: <ul style="list-style-type: none"> •  Task in progress (spinning). •  Task completed successfully (green). <i>Success</i>⁽²⁾ •  Task did not complete (red). <i>Not successful</i>⁽²⁾ •  Task completed with notifications (yellow). <i>Warning</i>⁽²⁾ •  Information (blue). <i>Informational</i>⁽²⁾
Start Time	Date and time when the task was initiated.
Elapsed Time	The time it took to complete the task independently of its final status.
Message	Short description of the task that was executed. The status of the task is indicated in brackets. A task can have messages for subactions. Expand the row by clicking the expand icon on the outer left of the row.
Action	Keyword describing the action that was executed. Typically, this corresponds to the command you selected.
User Name	Name of the user who is logged in to the client from which the task was executed. Format: <i>Full name (Username)</i>
Station Node	Name of the PC on which the client from which the task was executed is installed.
System	Identifier of the system in which the task was executed. When the task is not related to a system, the entry is Global .
Domain	Identification of the domain or project component on which the task was executed. When the task is not related to a system component (for example, when a user logs in), the entry is Global .
Entity Path	Path the entity on which the task was executed.
Entity Type	Type of the entity on which the task was executed.
Attribute	Identifier of the modified attribute (if applicable).
Old Value	Value of the attribute before the modification was performed (if applicable).
New Value	Value of the attribute after performing and/or saving the modification (if applicable).

(1) To show column headers, right-click any row inside the notification panel and click **Show Table Headers**. Headers are hidden by default.

(2) Corresponding keyword in the export file, page 75.

Using the Notification Panel

For each task that you launch, it is good practice to verify in the notification panel if the task is completed and its final status.

Select a task that has subentries and press **Enter** (or double-click it) to expand/collapse it. Subentries provide more detailed information about the task.

NOTE: When performing activities on a large number of entities, expanding entries in the notification panel may increase the response time of the engineering client temporarily. Such activities can be, for example, importing or generating 500 or more objects at once.

Filtering and Sorting Activity Data

Use the filter function, page 118 on columns to view, for example, only activity of one user by filtering on the **User Name** column.

By default, the most recent activity is displayed at the top of the list.

Exporting Activity Data

To save the history of actions, export activity data on a regular basis and before stopping the system server.

To export to file (.csv) activity data that is displayed in the notification panel covering a selected period, proceed as follows.

Step	Action
1	Right-click any row inside the notification panel and click Export . Result: The Export dialog box opens.
2	In the Destination field, enter a file name and location, or use the default file name and location.
3	In the Period field, select a start date and time, and an end date and time to define the time span of activity data to export. NOTE: Only activity data that appears in the notification panel can be exported.
4	Click OK . Result: The software creates the export file and opens the folder that contains it. NOTE: Click Cancel to close the Export dialog box without exporting data. NOTE: Click Reset to reset the values of the Destination and Period fields to their default values.

NOTE: If a file with the same name exists at the specified location, the software displays a dialog box asking you for confirmation before overwriting the file.

Audit Trail

This topic describes the audit trail functionality for system-engineering activities performed by users and the data that is sent to the syslog server.

Configuring Audit Trail

You can enable and configure audit trail (see EcoStruxure™ Process Expert, Installation and Configuration Guide) from the **Audit Trail** section of the **System Server Configuration Wizard**.

Working Principle

Once you have enabled and configured audit trail, when the system server is started, it connects to the syslog server and sends audit trail messages. Messages are sent in chronological order for actions that users perform on systems by using the engineering clients of the infrastructure.

Messages for certain user actions that are performed in the system server or operation clients are also sent.

The audit trail functionality of EcoStruxure Process Expert is independent of the **Event Logging** setting of the Control Participant (**Project Settings > PLC Diagnostics**).

NOTE: Syslog server software needs to be purchased and installed separately.

Scope of the Audit Trail

Typically, audit trail messages that are sent to the syslog server correspond to a user-action entry in the notification panel, page 73 of the engineering client.

For each child notification pertaining to a parent entry, one message is also sent. Messages for the main entry and its child notifications have the same *Domain* value.

No message is sent for system-generated entries and when a user action is canceled.

The following tables outline the type of actions for which messages are sent to the syslog server.

Actions Performed in Engineering Clients

Actions performed at the platform level	Comment
Logging in and out of the client.	—
Locking and unlocking the client.	—
Actions on systems.	For example, creating, deleting, or restoring systems.
Actions on system components.	Application, topology, projects.
Import operations.	—
Deployment operations.	—
Actions on global templates.	—
Actions on content repository data.	—

Actions performed in a Participant	Comment
Control Participant: <ul style="list-style-type: none"> • Refinement offline of project and sections • Controller and PRM configuration 	Participant-generated messages are sent when changes are saved. Typically, this information does not appear in the notification panel of the engineering client.
Control Participant: Refinement online.	Messages are sent when the Build and Deploy command is used. A message is sent also when you perform an action that impacts the controller directly (for example, initializing an animation table) even if you close the refinement window without building and deploying changes.
Supervision Participant: <ul style="list-style-type: none"> • Refinement • Advanced settings • Page editing 	The information that is available is the same as that appearing in the notification panel. No details are provided about Participant data that is modified.

Actions Performed in Operation Clients

Actions	Comment
Logging in and out of the client.	—
Locking and unlocking the client.	—
Connecting to a controller when using runtime navigation services.	A message is sent for obtaining the controller status when opening the Operation Client Viewer.

Actions Performed in the System Server

Actions	Comment
Starting and stopping the system server.	—
Logging in and out of the system server.	—

Audit Trail Message Format

The content of the message portion (message text or details) of a message that is sent to the syslog server uses the format defined by the syslog protocol (RFC 5424 specification), which is as follows:

<Time stamp> <Machine name> <Application name> – <User> : <Domain> : <Action> : <Entity> : <Message> : <Attribute> : <Old value> →<New value>

Element	Description
Time stamp	Date and time when the action was performed. Format: YYYY-MM-DDThh:mm:ss:ms Milliseconds (ms) are indicated with 7 digits (0000000 if not available). The time stamp is based on the system time of the computer on which the EcoStruxure Process Expert system server is installed in UTC. Nevertheless, Schneider Electric recommends that the system time of computers on which EcoStruxure Process Expert clients are running be synchronized to the system time of the system server computer. NOTE: For actions performed inside the Control Participant, the time stamp corresponds to the Participant system time in UTC.
Machine name	Name of the computer on which the action was performed.
Application name	EPE
User	Name of the user who is logged in to the EcoStruxure Process Expert component on which the action was performed. Format: <i>Full name (Username)</i> . If the full name is not defined, the username appears instead.
Domain	Domain or project component on which the action was performed. For example, <i>Systems</i> or <i>Application</i>
Action	Description of the action that was performed.
Entity	Name of the object on which the action was performed.
Attribute	Identifier of the modified attribute (if applicable).
Message	Description of the action. Typically, the information that is available in the notification panel of the engineering client.
Old value	Value of the attribute before the action was performed.
New value	Value of the attribute after the action was performed.

NOTE: If an element contains no value, it is not shown.

Connection to the Syslog Server

The table describes the actions of the system server when it connects to the syslog server and when the connection is interrupted.

Connection of system server to syslog server	Description
Initial connection	A message is shown in the notification panel and sent to the syslog server to indicate the status of the connection. In the notification panel, messages related to the connection status show <i>Global</i> in the System and Domain columns and <i>System</i> in the User Name column.
Connection is interrupted	The system server detects a disconnection when it is not able to send messages to the syslog server. In this case, it attempts to reconnect automatically up to three times. If unsuccessful, a message is shown in the notification panel. Thereafter, the system server attempts to reconnect at the configured interval.
Connection is reestablished	Two messages are shown in the notification panel and sent to the syslog server to indicate the following: <ul style="list-style-type: none">• Date and time the disconnection was interrupted.• Date and time the connection was reestablished.

NOTE: Actions logged while the connection to the syslog server is interrupted are discarded. Notifications continue to show in the notification panel.

Activity Log Files

Overview

The software records the activity of each component (system server, engineering client, and operation client) by using log files.

In addition, the software maintains a log file that records information related to activated licenses for troubleshooting purposes.

The file size and number of files that the software maintains for each component is limited. This is done to make the log files more user-friendly to view and to save space.

NOTE: For clients, the data that is logged is different from the activity data that is displayed in the notification panel, page 73.

Log File Location

Log files are located at the following paths:

- System server: C:\ProgramData\Schneider Electric\Process Expert X\Logs
- Clients: C:\Users\Username\AppData\Roaming\Schneider Electric\Process Expert X\Logs

Where:

- X corresponds to the version of the software that is installed.
- *Username* corresponds to the username that was entered to log on to the current session of Windows®, independently of the username that was entered to log in to the client, page 60.

Log File Name Syntax

The names of log files have the following syntax: ComponentName.File number.log

Component names are:

- *Server* for the system server.
- *Client* for engineering clients.
- *OperationClient* for operation clients.
- *License* is used for license-related information.

File numbers:

- The most recent log file has no file number.
- The second latest log file has number 0.
- The log files thereafter have incremental numbers from 1 to 9 where 9 represents the earliest log file containing the oldest information.

For example:

- *Server.log* is the current log file in which the software is recording server activity.
- *Client.1.log* is the log file in which the software recorded past engineering client activity. The file content is not updated anymore because *Client.log* is currently recording activity. *Client.1.log* was created when *Client.log* reached its size limit for the second time (the first time, *Client.0.log* was created).

Log File Size

The size of any log file cannot exceed 4 Mb. When the size of a log file has reached this limit, the software creates a new file. Existing log file numbers are incremented.

Number of Log Files

The software keeps up to 10 log files per component. When the software needs to create a new log file because the current one has reached the size limit, and the number of files for a component is reached, the software deletes the earliest file for that component before creating the new one.

Software Capabilities

What's in This Chapter

Hardware and Software Capabilities	83
Supported Data Types	86
Naming Rules	88

Hardware and Software Capabilities

Overview

The following describes the main hardware and software capabilities of EcoStruxure Process Expert and additional capabilities of the embedded Control and Supervision software Participants.

For more information on the specific capabilities of the Control Participant and Supervision Participant, refer to the help of the respective Participants, page 93.

Some capabilities and/or features may not be supported by the software.

Hardware Platforms

For information on the integrated and not supported hardware architectures, refer to Configuration Stage (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: You can include other hardware platforms in systems engineered by using the software, however, the software will not provide any services for these platforms.

Programming Languages

The software generates the program for controllers by using the Function Block Diagram (FBD) programming language.

The Control Participant supports additional programming languages, page 84 for creating the user program, which can be used together in Control projects managed by EcoStruxure Process Expert.

The Supervision Participant supports a structured, proprietary programming language for monitoring and controlling plant equipment, which can be used together in Supervision projects managed by EcoStruxure Process Expert. For more information, refer to the Supervision Participant help, page 93.

Template Libraries

The software uses object-oriented modeling for the engineering of applications. For this purpose, an extensive selection of Schneider Electric templates are provided in various libraries such as templates modeling signals, hardware templates modeling the system infrastructure, up to complex control module templates modeling the entire Control and Supervision functionalities of process objects.

When a library is installed, the templates are available in the **Global Templates Explorer** and can be used in explorers and editors specific to the software. The encapsulated Control and Supervision constituents can be viewed in the corresponding software Participants.

For information on architectures, for which templates are available, refer to Supported Hardware Architectures (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

Function and Function Block Libraries

The Control Participant includes an extensive library, which extends from blocks for simple boolean operations, through blocks for strings and array operations to blocks for controlling complex control loops.

For a better overview, the different blocks are arranged in libraries, which are then broken down into families.

The blocks can be used in the Control Participant in the programming languages FBD, LD, IL, and ST and be added to Control projects managed by EcoStruxure Process Expert.

Controller Program

In the software, the controller program is organized in:

- A Master task (MAST)
- A FAST task
- FBD sections, which are assigned to one of the defined tasks

By using the Control Participant, you can add to the controller program:

- Sections, which are assigned to one of the defined tasks
- AUX tasks (depending on the platform)
- Sections, which are assigned to AUX tasks
- Sections for processing time controlled events (Timerx)
- Sections for processing hardware controlled events (EVTx)
- Subroutine sections (SR)

The program components that are supported by the Control Participant, can be added to Control projects managed by the software. However, not all of them can be viewed in the software outside of the Control Participant (for example, non-FBD sections).

Performance Scope

The table shows the main characteristics of the software and its Participants.

	EcoStruxure Process Expert	Control Participant	Supervision Participant
Programming languages			
Function Block Diagram (FBD)	+	+	-
Ladder Diagram (LD)	-	+	-
Instruction List (IL)	-	+	-
Structured Text (ST)	-	+	-
Sequential Function Chart (SFC)	-	+	-
Ladder Logic 984 (LL984)	-	+	-
Cicode	-	-	+
Schneider Electric Global Templates libraries			
Foundation library	+	-	-
General Purpose and segment libraries	+	-	-
Function block libraries⁽¹⁾			
Standard library	-	+	-
Control library	-	+	-
Communication library	-	+	-
Diagnostics library	-	+	-
I/O Management library	-	+	-
System library	-	+	-
Obsolete library	-	+	-
Memory card file management library	-	+	-

	EcoStruxure Process Expert	Control Participant	Supervision Participant
General information			
Project browser/explorer	+	+	+
Template explorer	+	-	-
Instance editor	+	-	-
Create and modify object templates	+	-	-
Access control	+	+	+
Simulator	-	+	-
Context-sensitive and PDF help	+	+	+
Control			
Encapsulate DFB constituents in templates	+	-	-
Create and use data structures (DDTs)	+	+	+
Create and use derived function blocks (DFBs)	-	+	-
HMI operator screens	-	+	-
Supervision			
Encapsulate tags and genies in templates	+	-	-
Design pages, genies, face plates	-	-	+
Use tags and alarms	+	-	+
Clusters	+	-	+
Server and client events	-	-	+
I/O devices	+	-	+
I/O, alarm, trend, and report servers	+	-	+
Redundancy	+	-	+
Load balancing (operation servers)	+	-	+

+ Available

(1) Availability of the function blocks depends on the hardware platform.

- Not available

Supported Data Types

Data Type Conversion

The software may use data types that are different from the types used by the Control Participant (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

The conversion to Participant-supported types is done automatically.

Standard Data Types

The software uses the following data types.

Data Type	Description	Lower Limit	Upper Limit	Size
DateTime	A specific instance in time ⁽¹⁾ .	19-01-01 00:00:00.000	2100-12-31 23:59:59.999	8 bytes
Date	A Gregorian calendar date ⁽¹⁾ .	1900-01-01	2100-12-31	8 bytes
Time	An instant of time that recurs every day ⁽¹⁾ . hh:mm:ss.ms	00:00:00.000	23:59:59.999	8 bytes
Duration	A duration of time as specified by Gregorian day, hour, minute, second, and millisecond components ⁽¹⁾ . D.hh:mm:ss.ms Example: 1.05:10:30.245 If the value contains no day or ms component, they are not shown as 0 values. For example: 11:20:30 is displayed to represent 0 days, 11 hours, 20 min., 30 sec., and 0 ms.	0	9,223,372,036,854,775,807 in ticks NOTE: 1 tick = 100 ns	8 bytes
String	A sequence of unicode characters.	1 character of the standard ASCII table.	n characters of the standard ASCII table.	1 character = 1 byte
Boolean	A binary logic value.	False	True	1 byte
Float	Represented by using the 32-bit single-precision IEEE 754 format ⁽¹⁾ . 7-digit precision.	From approximately -3.4 x 10 ³⁸ to -1.5 x 10 ⁻⁴⁵	From approximately 1.5 x 10 ⁻⁴⁵ to 3.4 x 10 ³⁸	4 bytes
Double	Represented by using the 64-bit double-precision IEEE 754 format ⁽¹⁾ . 15 to 16-digit precision.	From approximately -1.7 x 10 ³⁰⁸ to -5.0 x 10 ⁻³²⁴	From approximately 5.0 x 10 ⁻³²⁴ to 1.7 x 10 ³⁰⁸	8 bytes
Decimal	A fixed-point decimal value with arbitrary precision ⁽¹⁾ . A maximum of 4 digits are allowed after the decimal separator.	0.0001	922337203685477.5807	12 bytes
Byte	8-bit signed integer.	-128	127	1 byte
Short	16-bit signed integer.	-32768	32767	2 bytes
Integer	32-bit signed integer.	-2,147,483,648	2,147,483,647	4 bytes
Long	64-bit signed integer.	-9223372036854770000	9,223,372,036,854,774,807	8 bytes
UnsignedByte	8-bit unsigned integer.	0	255	1 byte
UnsignedShort	16-bit unsigned integer.	0	65,535	2 bytes
UnsignedInt	32-bit unsigned integer.	0	4,294,967,295	4 bytes

Data Type	Description	Lower Limit	Upper Limit	Size
UnsignedLong	32-bit unsigned integer.	0	18446744073709551615	4 bytes
(1) Microsoft® Windows® Regional Options may apply to the format of the type.				

User-Defined Data Types

Data Type	Description	Lower Limit	Upper Limit	Size
Enum	An enumeration value is stored as String.	–	–	–
Variant	Any data type.	–	–	–

Naming Rules

Overview

When you change default parameters such as identifiers or Supervision element properties, EcoStruxure Process Expert validates at different stages of the system engineering life cycle if they satisfy the naming rules that are described in this topic.

These rules can apply at the platform level and/or can be Participant-specific.

NOTE: Other Participant specific rules may apply; however the software does not validate them. For more information, refer to the respective Participant help, page 93.

Folder Identifiers

Folder identifiers:

- Can start only with a letter or the \$ character.
- Can contain only letters, digits, underscores, the \$ character, and spaces⁽¹⁾.

These rules apply to the following types of folders:

- System folders
- Application folders
- Global Templates folders
- Content containers of the content repository.

Instance Identifiers

Instance identifiers can contain only letters, digits, and underscores.

The maximum character length is set through the **Validation** property (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of the **\$InstanceId** parameter of the template that the instance uses.

NOTE: Participants do not accept the use of certain reserved words for object names (for example, DO for the Supervision Participant). While the software allows using these reserved words as identifiers of instances in the application, doing so may prevent the corresponding Participant project from generating successfully when the reserved word is used as name of Participant project items (for example, equipment, page 425).

Topological Entity, Device and Folder Identifiers

Topological entity, device, and folder identifiers:

- Can start only with a letter.
- Can contain spaces⁽¹⁾.
- Cannot contain any special character other than underscore (_).
- Cannot exceed 32 characters in length.

Control and Supervision Participant Project Identifiers

Identifiers of Control and Supervision Participant Projects, and of their components:

- Can start only with a letter.
- Can contain spaces⁽¹⁾.
- Cannot contain any special character other than underscore (_).
- Cannot exceed 32 characters in length.

Variable Names and Section Identifiers

Names of variables and identifiers of sections that are managed by the Control Participant, must satisfy the following rules:

- Maximum length: Refer to the maximum length of identifiers as specified under *Implementation-Dependent Parameters* in the IEC 61131-3 standard (*Appendices of the Languages Reference* manual in the Control Participant help).
- Permitted characters: Refer to *naming conventions* in the glossary of the Control Participant help, *Operating Modes*.

Refer to the topic describing data validation before generating, page 385.

Supervision Cluster Identifiers

Identifiers of Supervision clusters cannot exceed 16 characters in length.

Supervision Page Identifiers

Identifiers of pages must satisfy the following rules:

- Maximum length: 64 characters
- Can contain spaces⁽¹⁾.
- Cannot contain these characters: \/: * ? " < > |

(1) The software automatically removes leading and trailing spaces.

Supervision Field Lengths

For variable tags:

Rule	Field	Value
Maximum field length	<i>Item Name</i>	79
	<i>Address</i>	254
	<i>Raw Zero Scale</i>	11
	<i>Raw Full Scale</i>	
	<i>Eng Zero Scale</i>	
	<i>Eng Full Scale</i>	
	<i>Eng Units</i>	8
	<i>Format</i>	11
	<i>Comment</i>	254
	<i>Deadband</i>	11
	<i>Data Type</i>	16

For trend tags:

Rule	Field	Value
Maximum field length	<i>Item Name</i>	79
	<i>Expression</i>	254
	<i>Trigger</i>	11
	<i>Format</i>	11
	<i>File Name</i>	253
	<i>No. Files</i>	4
	<i>Comment</i>	254

Rule	Field	Value
	<i>Period</i>	32
	<i>Storage Method</i>	64
	<i>Type</i>	32
	<i>Eng Units</i>	8

For advanced alarm tags:

Rule	Field	Value
Maximum field length	<i>Custom1... to Custom8...</i>	64
	<i>Paging Group</i>	80

For digital alarm tags:

Rule	Field	Value
Maximum field length	<i>Comment</i>	254
	<i>Custom1... to Custom8...</i>	64
	<i>Paging Group</i>	80

For reports (and facets assigned to **ServerEvents** containers):

Rule	Field	Value
Maximum field length	<i>Report Name</i>	79
	<i>Output Device</i>	16
	<i>Report Format File</i>	253
	<i>Comment</i>	48
	<i>Area</i>	16
	<i>Action</i>	16
	<i>Trigger</i>	254
	<i>Time</i>	32
	<i>Period</i>	32

For events (and facets assigned to **ClientEvents** containers):

Rule	Field	Value
Maximum field length	<i>Name</i>	16
	<i>Trigger</i>	254
	<i>Event (Action)</i>	64
	<i>Comment</i>	48
	<i>Time</i>	32
	<i>Period</i>	32

Supervision Fields Special Characters

The table describes the rules that the software validates during generation with regard to the contents of the following field.

Field	Rule
<i>Name</i>	<p>Tag names need to have the following syntax:</p> $[<\text{alpha}> '_']^* [<\text{alpha}> <\text{digit}> '\' '_']$ <ul style="list-style-type: none"> • Tag names need to begin with either an alpha character (A-Z or a-z) or the underscore. • Any following characters need to be either of: <ul style="list-style-type: none"> ◦ alpha characters (A-Z or a-z) ◦ digits ◦ \ ◦ _ (underscore) ◦ (vertical bar) <p>If you use any other character, the compile process cannot complete successfully.</p> <p>NOTE: Tag names can only begin with a digit only if the <i>[General] TagStartDigit</i> parameter of the citect.ini file is set to 1 (the default value is 0).</p>

Using the Help

What's in This Chapter

Software Participant Help.....	93
Using Help for Different Versions of Schneider Electric Templates.....	94
Help on User-Created Templates.....	96

Software Participant Help

Overview

You can open the HTML help of a Participant by clicking **Help** or the **?** button in the menu of the Participant when the Participant window is open (for example, after you click **Configure** in the context menu of a topological entity).

You can open the same help also from the **Help EcoStruxure Process Expert** dialog box without having to open the Participant. To access the dialog box, click the **?** button in the toolbar of a client.

Help is available for these Participants:

- The Control Participant
- The Advantys Configuration Software (ACS)
- The Supervision Participant

NOTE: ACS is used to configure STB islands during the configuration stage. For simplicity, in this manual ACS is assimilated to the Control Participant.

NOTE: The dialog box lets you also open:

- The help describing how to use the Modicon IEC 61850 Configuration Tool.
- The help of OPC Factory Server.
- The help of OPC UA Server Expert.

Opening the Help of a Participant

To open the help of a Participant from a client window, proceed as follows.

Step	Action
1	In the toolbar, page 101 of the client window, click the help button (?). Result: The Help EcoStruxure Process Expert dialog box opens.
2	In the Software Participant Help section, double-click a Participant name. Result: The HTML help of the Participant opens in a separate window.

Using Help for Different Versions of Schneider Electric Templates

Overview

For the templates that are included with the software, each software version contains only one version of each template (the latest) and installs the help only for this version. This overwrites the help of an existing previous version if you are upgrading the software. The help is provided in compiled HTML (CHM) format.

However, your application can contain several versions of a same template, for example, if you have migrated your database and have imported a later version of the template afterwards.

Starting from V4.3 of the software (Process Expert), you have the possibility to retain the help that is included with any version of the software to continue using it after you upgrade to a later version.

By verifying the version of a template when you use the context-sensitive help button, the software opens the help for this version if available.

This topic provides recommendation when using several versions of the help and describes how to proceed.

NOTE: For more information on how to obtain help on templates, refer to the topic describing how to .

Considerations When Using Several Versions of the Help

The help of a given version of the software documents the templates and functionalities provided by this version only.

If you open the help for a given template and then, within the same help window, you look up another template or a platform functionality, it may be that the version of the template or the functionality that is documented differs from the version/ functionality that you are using.

This can be the case, for example, if:

- You have migrated your database and have not updated the template used by instances of the Application. In this case, the help that is provided may document a later version of templates.
- You are opening the help of an earlier version of the software. In this case, the templates that are available in the library and the functionalities provided by the platform may be of a later version than those documented.

Schneider Electric recommends the following:

- Open the help for a template only by using the context-sensitive help button, which takes into consideration the version of the template that you are selecting.
- Do not view the help for another template or platform functionality within the help window that you had opened for a template.
- Verify the version of the help you are viewing and which is indicated in the title of the help window.

Location of the HTML Help Files

On the computer running the engineering client, the help files are located in the User Guides folder at the path C:\Program Files\Schneider Electric\EcoStruxure \Process Expert.

The User Guides folder also contains ContextualHelpRepository.* files, which are required for the context-sensitive help on templates to work.

Preserving the Current Help Files Before Upgrading the Software

So that the help of the installed version of the software remains available after you upgrade to a later version, before proceeding with the upgrade, you need to copy the complete set of help files and their auxiliary files to a specific location on the computer running the engineering client.

Proceed as follows.

Step	Action
1	<p>Create the following folder structure if it does not exist yet: %localappdata%\Schneider Electric\Process Expert\User Guides.</p> <p>NOTE: By default, the value of the %localappdata% environment variable is C:\Users\username\AppData\Local.</p> <p>A Process Expert x folder, where x corresponds to the installed major version (the R⁺ version suffix does not appear), may already exist in the Schneider Electric folder but do not use it.</p>
2	<p>In the User Guides folder, create a subfolder with a name containing the number of the current version. For example, <i>Template Help Files Vx</i>.</p> <p>Mentioning the version helps you identify for which version of the software you are opening the help in case the same template version is provided and documented in different versions of the software.</p>
3	Copy all the files contained in the User Guides folder, page 94 to the subfolder that you have created in step 2.

Opening the Help for Templates of an Earlier Version

To open the help of a template for which several versions exist, proceed as follows.

Step	Action
1	<p>Click the context-sensitive help button in the toolbar, page 101 of the engineering client and select a template, an instance of a template, or a facet of an instance of a template.</p> <p>Result: The software verifies the version of the template and opens:</p> <ul style="list-style-type: none"> The help of the current version from the default location if the template is provided in the current version of the software. The help of the earlier version of the software in which the template was provided (given you have copied the necessary help files to the required location, page 95). <p>Only one help window opens at a time.</p>

NOTE:

- If the version of the template was provided with a version of the software for which the help is not present on the computer, not at the required location, or incomplete, no help is shown.
- If the version of the template was provided in both the current and the earlier version of the software, the **Contextual Help Conflict** dialog box opens, which lets you select which version of the help you want to open.

Help on User-Created Templates

Overview

You can configure the software to open the help in CHM format of a user-created template.

You can also manage the HTML help files of several versions of the same template so that clicking a given version of the template opens the corresponding help.

Only one help window opens at a time.

This topic describes the prerequisites and the steps you need to perform. It does not describe how to create HTML help files in CHM format.

NOTE: By using this feature, you can also associate your own HTML help to a Schneider Electric global template in addition to the existing help. You can then select which one you want to open.

NOTE: You can also associate help files to templates and their instances that operators can open from the operation client when using runtime navigation services (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).

Prerequisites

The following are the prerequisites to be able to open help for user-created templates:

- You need to have help files documenting user-created templates in the Microsoft® compiled HTML format (.chm).
For more information on creating a help project, refer to the Windows® online help.
- You need to know the topic ID, page 97 of the topic documenting the template, which you want to open when you use the context-sensitive help button of the engineering client. If the help file (.chm) documents several templates, you need to have the topic ID that corresponds to each template.
For more information on topics, refer to the Windows® online help.

Creating a Repository for Help Files of User-Created Templates

So that the help of user-created templates remains on the computer and can be opened after you upgrade the software to a later version, you need to copy your complete set of help files and the ContextualHelpRepository.xml file to a specific location on the computer running the engineering client.

Proceed as follows.

Step	Action
1	Browse to %localappdata%\Schneider Electric\Process Expert\User Guides. NOTE: By default, the value of the %localappdata% environment variable is C:\Users\{username}\AppData\Local.
2	In the User Guides folder, create a copy of the folder named <i>Sample-Tobecopied</i> in the same location and rename it by using a meaningful name describing its contents and the version of the software that was used to create the template. For example, <MyReactor Template Help Files for V2020>. NOTE: You can create as many copies of this folder as needed in case you want to keep the help files of different templates in distinct folders.
3	Copy all your help files to the subfolder that you have created in step 2.

NOTE: Uninstalling the software removes the folder named *Sample-Tobecopied* but not the copies of it.

Identifying the ID of a Topic Inside a CHM Help File

The procedure describes how to identify the ID of a topic inside a CHM help file.

Step	Action
1	Open the user-created HTML help,
2	Select the topic (page) that you want to open when using the context-sensitive help button on the template and right-click somewhere on the page. Result: The context menu of the page opens.
3	Select Properties . Result: The Properties dialog box opens.
4	Copy the topic ID. The topic ID of the page is the part of the string that starts after <code>: /</code> in the Address (URL) attribute. For example, if the Address (URL) attribute is <code>mk:@MSITStore:C:\PROGRA~2\XMLNOT~1\Help.chm::/Overview.htm</code> , the topic ID is <code>Overview.htm</code> .

Editing the XML File

To edit the ContextualHelpRepository.xml file, proceed as follows.

Step	Action
1	In the folder, page 96 to which you have copied your help files, open the ContextualHelpRepository.xml file by using Notepad or an XML editor software. Result: The file that opens contains the following entry, which serves as an example only. <pre><ContextualHelp Version="1.0"> <Item HelpKey="MyMotor3" FileName="PELbPrTp" MappingId="PELbPrTp_D-SE-0014236.htm" FileType="CHM" Description="" Version="2.7.2"/> </ContextualHelp></pre>
2	Edit the following parameters of the entry: <ul style="list-style-type: none"> <i>HelpKey</i>: Is the identifier of the user-created template. <i>FileName</i>: Is the name of the CHM file without file extension. <i>MappingId</i>: Is the ID, page 97 of the topic in the CHM file that documents the user-created template. Include the <code>.htm</code> file extension. <i>FileType</i>: Do not edit this parameter. Default value: CHM. <i>Description</i>: Optional. You can enter a short description by using free form text. It is used to describe the entry in the XML file and can appear in the Contextual Help Conflict dialog box, page 95. To use no description, leave it as "". <i>Version</i>: Is the version of the user-created template as it appears in the Global Templates Explorer or in the Templates Browser pane of the Application Explorer.
3	Add a new entry for each user-created template for which you want help to open. To manage the help for different versions of the same template, refer to the next topic, page 97. NOTE: The <code></ContextualHelp></code> element must be the last entry in the file.
4	Save your changes.

Updating the Help for a New Version of a User-Created Template

The table describes the actions that you need to perform when you create a new version of a template and want help to open for the various versions that exist. Two versions of a template are used as an example. *MyReactor 1.0.0* is considered the previous version and *MyReactor 1.0.1* the new version.

An updated help file is available to document the new version	Action	Comment
No	<ol style="list-style-type: none"> In the folder, page 96 to which you have copied your help files, edit the ContextualHelpRepository.xml file. Add a new entry that corresponds to the new version of the template by using the same parameter values as for the previous version; except for <i>Version</i>, which needs to correspond to the version of the new template. 	<p>The same help file opens when you use the context-sensitive help button on <i>MyReactor 1.0.0</i> or <i>MyReactor 1.0.1</i>.</p> <p>If you remove the entry for template version 1.0.0, help cannot open when you select it.</p>
Yes	<ol style="list-style-type: none"> Browse to the User Guides folder , page 96 located at the path %localappdata%\... Create a copy of the folder in which the help files of <i>MyReactor 1.0.0</i> and the ContextualHelpRepository.xml file are located. Rename the copy of the folder by using a name that allows you to distinguish the previous and the new version. For example, <MyReactor 1.0.1 Template Help Files for V2018>. In this folder, edit the entry in the ContextualHelpRepository.xml by modifying the value of <i>Version</i> to be equal to the new version (for example, <i>Version="1.0.1"</i>). 	<p>Clicking either template version when context-sensitive help is enabled opens the help file that is located in the folder, which also contains the XML file with an entry of the same template identifier and version.</p> <p>NOTE: It is assumed that the help file name and the topic ID are identical for both template versions.</p>

Opening the Help for User-Created Templates

To open the help of a user-created template, proceed as follows.

Step	Action
1	<p>Click the context-sensitive help button in the toolbar, page 101 of the engineering client and select the user-created template.</p> <p>Result: The software verifies the version of the template and opens the corresponding help file according to your configuration, page 97.</p>

NOTE:

- If the value of any parameter in the ContextualHelpRepository.xml file is incorrect (except for description), if the ContextualHelpRepository.xml file is missing or its location incorrect, or if the location of the folder containing the help files and the XML file is incorrect, no help is shown.
- If the template identifier and version appear in more than one ContextualHelpRepository.xml file contained in different subfolders of the User Guides folder, the **Contextual Help Conflict** dialog box opens, which lets you select from which location you want to open the help.

Associating Other HTML Help Files to Schneider Electric Global Templates

By using an example, this topic describes how to associate your own help file in CHM format to a sample control module template of a Global Templates library for which the EcoStruxure Process Expert help is already available.

The data used for the example is the following:

- Control module template: \$Motor version 1.0.0.
- Name of your help file: MyMotorHelp.chm
- ID of the topic that you want to open: mymotor_description.htm

The table describes the steps that you need to perform to implement the help and select, which file to open.

Step	Action
1	Create a repository for your help, page 96 and copy the MyMotorHelp.chm file to this folder.
2	<p>Edit the ContextualHelpRepository.xml file, page 97 located in this folder and add the following entry before the </ContextualHelp> element.</p> <pre><Item HelpKey="\$Motor" FileName="MyMotorHelp" MappingId="mymotor_description.htm" FileType="CHM" Description="" Version="1.0.0"/></pre>

Step	Action
3	<p>Click the context-sensitive help button in the toolbar, page 101 of the engineering client and select the \$Motor template version 1.0.0, and instance, or facet of it.</p> <p>Result: The Contextual Help Conflict dialog box opens and shows the two help files that are associated to the template: The EcoStruxure Process Expert help and your help.</p>
4	Select the help file that you want to view and click OK .

User Interface

What's in This Chapter

Engineering Client User Interface Description.....	101
About Dialog Box.....	104
Engineering Client and Explorer Work Areas.....	105
Organizing Work Areas.....	108
Working with Panes.....	111
Using Menus.....	115
Software Participant Windows.....	116
Filtering and Sorting Data.....	118
Keyboard Shortcuts.....	121

Overview

This chapter describes the engineering client user interface, its main components, and the common tools that allow you to engineer and navigate a system.

Engineering Client User Interface Description

Overview

When you start the engineering client, it opens one main work area in which the various explorers and editors are displayed.

The engineering client user interface is composed of five main elements:

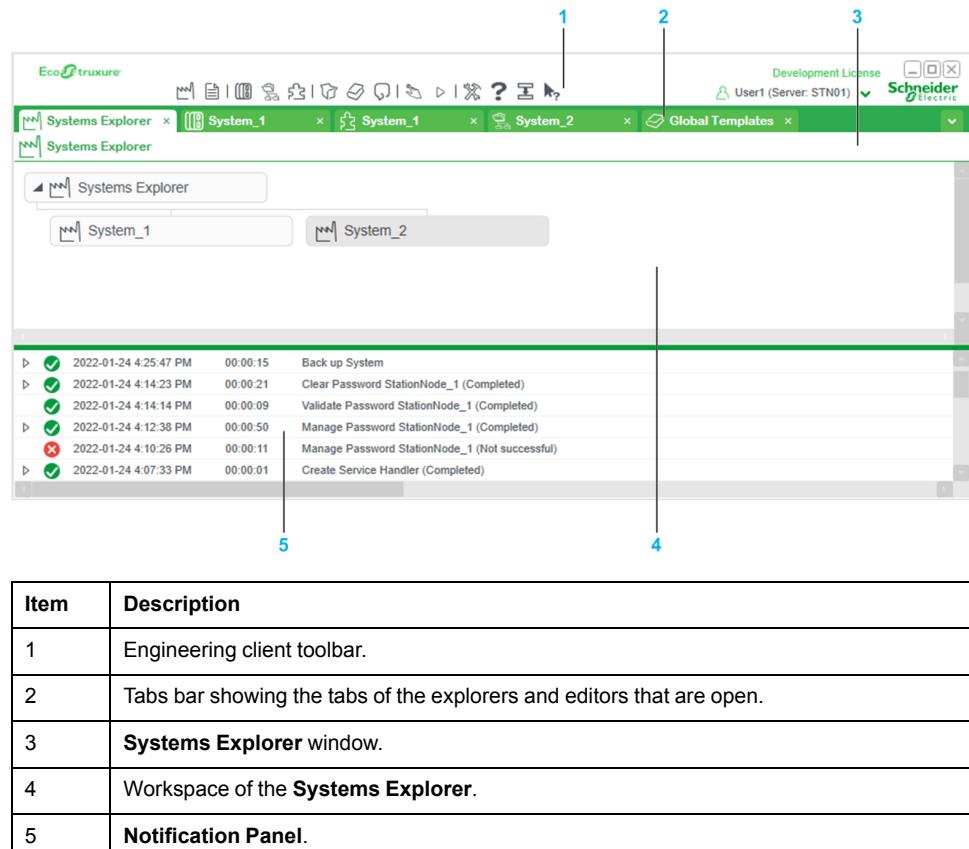
- The client toolbar
- The tabs bar of the main work area
- Explorer windows
- The **Notification Panel**, page 73

You can divide the main work area, page 108 of the engineering client.

The explorer that opens when a client is started depends on the group (see EcoStruxure™ Process Expert, Installation and Configuration Guide) that the logged-in user is a member of.

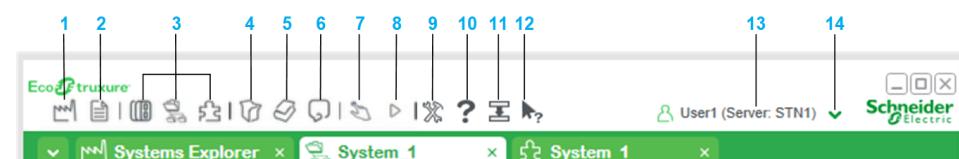
User Interface Description

The following figure shows an example of the engineering client user interface.



Engineering Client Toolbar Description

The following figure describes the toolbar of the engineering client.



Item	Description
1	<p>Opens the Systems Explorer window, page 133, which lets you create, organize, and navigate systems.</p> <p>From the Systems Explorer you can open the explorers that are needed to create and manage system components:</p> <ul style="list-style-type: none"> • Application Explorer • Project Explorer • Topology Explorer
2	<p>Opens the Generate Report window, which lets you create and save to disk a report, page 927 that contains engineering information about the selected system or one or more of its domains (such as, topology, Control projects), including Participant project documentation.</p> <p>All windows in the engineering client, except the Systems Explorer, need to be closed to use the command.</p>
3	<p>Explorer navigation buttons to open or show the explorer of a system.</p> <p>From left to right:</p> <ul style="list-style-type: none"> • Topology Explorer • Application Explorer • Project Explorer <p>Clicking a button opens the corresponding explorer for either:</p> <ul style="list-style-type: none"> • The system that is selected in the Systems Explorer when this explorer is active. • The system for which an explorer or editor is active. For example, if the Assignment Editor for <i>System_1</i> is active, clicking the first button opens/shows the Topology Explorer for <i>System_1</i>. <p>If the explorer is already open, it is shown.</p> <p>NOTE: Navigation does not work from the Global Templates Explorer nor the Contents Repository by using these buttons.</p>
4	<p>Opens the Content Repository explorer, page 936, which contains files related to Global Templates and the systems that you are engineering.</p>
5	<p>Opens the Global Templates Explorer, page 839, which you can use to browse and manage templates of the Global Templates library.</p> <p>NOTE: In this example, the button is shown unavailable because the role of the logged-in user, page 60 does not have the permission to use this explorer.</p>
6	<p>Opens the Notification Panel, page 73.</p>
7	<p>Lets you release the Supervision Participant so that you can open it (see EcoStruxure™ Process Expert, Supervision Participant Services User Guide) outside EcoStruxure Process Expert.</p> <p>Releasing is not required to launch the Plant SCADA runtime from the engineering client.</p>
8	<p>Launches the Supervision runtime (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide) with the Supervision project that is set as active on the local computer after you confirm the command.</p> <p>If the runtime is already launched, you can restart it by using the command.</p> <p>The runtime is not launched if the Supervision Participant is being used by EcoStruxure Process Expert (for example, if a Supervision project is being refined or its executable built).</p>
9	<p>Opens the About EcoStruxure Process Expert dialog box, page 104.</p>
10	<p>Opens a dialog box, which lets you access:</p> <ul style="list-style-type: none"> • The global software HTML help. It describes how to install, license, and use the software as well as the Global Templates that are included with it. • The global help of each software Participant, page 93. • Other help files. <p>Double-click an entry to open the corresponding help.</p>
11	<p>Opens the HTML help at the topic describing the system engineering life cycle and the system creation workflow.</p> <p>It gives you an overview of the various stages of the life cycle and helps you identify what is the next step that you need to complete.</p>
12	<p>Opens context-sensitive HTML help.</p>

Item	Description						
	<p>Clicking the button enables context-sensitive help and a question mark appears next to the cursor.</p> <p>Selecting an item opens the help when available. If no help is available for the selected item, the functionality is disabled.</p> <p>To disable the functionality while the question mark is shown next to the cursor, press Esc.</p> <p>For an overview of UI items for which context-sensitive help is available, refer to the topic describing how to open context-sensitive help.</p>						
13	Full name of the logged-in user and name of the computer running the system server to which the client is connected. If the full name is not defined, the username appears instead.						
14	<p>Opens the User menu, which contains the following commands.</p> <table border="1" data-bbox="605 601 1464 822"> <tr> <td data-bbox="605 601 759 669">Lock</td><td data-bbox="759 601 1464 669">Locks, page 62 the components to which you are logged in on the local computer.</td></tr> <tr> <td data-bbox="605 669 759 759">User Information</td><td data-bbox="759 669 1464 759">Opens the User Information dialog box (see EcoStruxure™ Process Expert, Installation and Configuration Guide).</td></tr> <tr> <td data-bbox="605 759 759 822">Log In/ Log Out</td><td data-bbox="759 759 1464 822">Lets you log in or out, page 60 of the software.</td></tr> </table>	Lock	Locks, page 62 the components to which you are logged in on the local computer.	User Information	Opens the User Information dialog box (see EcoStruxure™ Process Expert, Installation and Configuration Guide).	Log In/ Log Out	Lets you log in or out, page 60 of the software.
Lock	Locks, page 62 the components to which you are logged in on the local computer.						
User Information	Opens the User Information dialog box (see EcoStruxure™ Process Expert, Installation and Configuration Guide).						
Log In/ Log Out	Lets you log in or out, page 60 of the software.						

About Dialog Box

Opening the About Dialog Box

To open the dialog box, click:

- The  icon in the toolbar of the engineering or operation client, page 101.
- **Help > About** in the menu bar of the system server.

General Tab

The **General** tab displays information related to:

- Installed EcoStruxure Process Expert and third-party software versions.
 - The activated application size license and the total instance count for your database.
 - The installation and expiration dates of entity certificates (see EcoStruxure™ Process Expert, Installation and Configuration Guide) installed on the local computer and of the corresponding root CA.
- If no entity certificate is installed, no information is shown.

System Details Tab

The **System Details** tab displays the instance count by system, page 174 for the mounted database.

Client Details Tab

The **Client Details** tab displays information about the clients that are currently connected to the system server with a user being logged in. It shows the computer name, user name, and the type of client.

Support Tab

Contains the contact information of the Schneider Electric support.

The **Logs Folder** link lets you access the activity log files, page 80.

Hotfix Details Tab

The **Hotfix Details** tab displays information about installed hotfixes.

If no hotfix is installed, the tab is hidden.

Engineering Client and Explorer Work Areas

Overview

By default, in an engineering client only one work area is visible at a time.

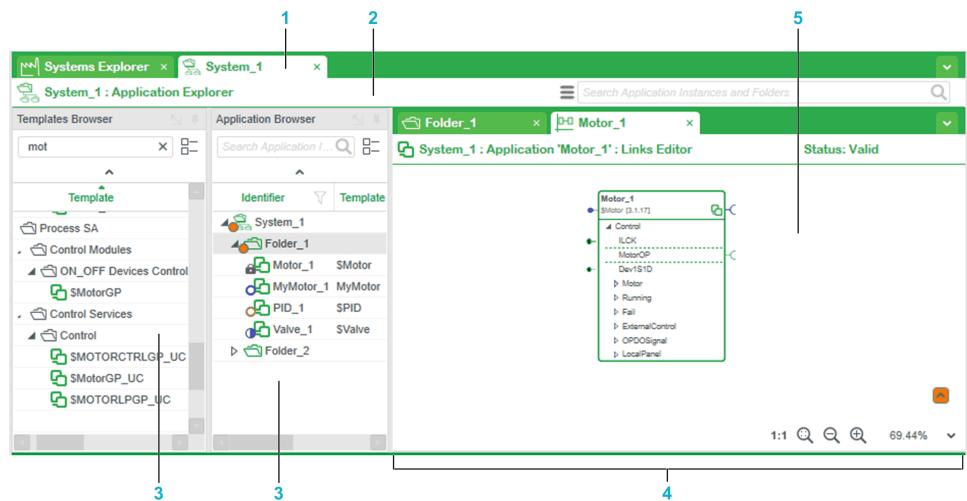
The work area is the display area that is below the tabs bar, page 101 where explorers and certain editors open, such as the **Systems Explorer** and **Composite Editor**.

Certain explorers (for example, the **Application Explorer** or **Project Explorer**) also feature a work area in which various windows and editors open, such as **Properties** windows or the **Instance Editor**.

You can create additional work areas, page 108 and organize them.

NOTE: When no window opens in the work area, it is called a *workspace*. This applies to editors and certain explorers (for example, the **Global Templates Explorer**).

The following figure shows an example of the work area of the **Application Explorer**.



Item	Description
1	Tab of the Application Explorer .
2	Toolbar of the explorer.
3	Panes of the explorer, which are open in the work area.
4	Remaining work area of the explorer in which a window and an editor are open. You can select them by using their tab.
5	Workspace of the editor.

Tabs Bar Description

For each window, explorer, or editor that you open in a work area, a tab is displayed in the tabs bar, which features:

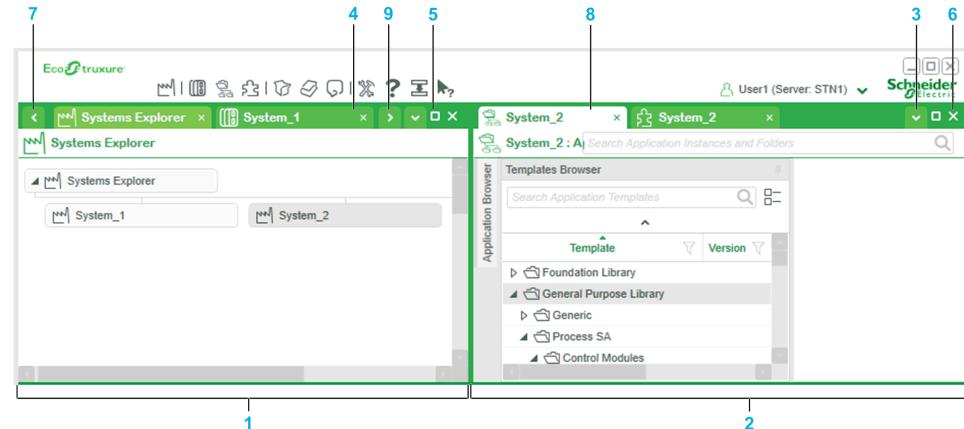
- An icon representing the type of window, explorer, or editor (for example, **Properties** window, **Application Explorer**, or **Links Editor**).
- The identifier of the system that is displayed in the corresponding window (except for the **Contents Repository**, **Global Templates**, and **Systems Explorer**).
- A button to close it.

You can rearrange the order of tabs on the tabs bar by dragging them to a new position.

NOTE:

- Closing the last tab in a tabs bar closes the work area (except if the work area is the last one to be open).
- You cannot close the last tab of the last work area that is open.
- To close all windows except one, right-click the tab of the window that you want to leave open and select **Close Other Tabs**.

The following example describes the tabs bars of the engineering client in which two work areas are open.



Item	Description
1	Main work area.
2	Additional work area.
3	Display all Tabs button. Displays a list of the tabs that are open in this work area.
4	Inactive tab.
5	Maximize/Restore Down button. The button is available when more than one work area, page 108 is open. It maximizes the work area in the window or reverts the display to a multiple work area view. NOTE: When you maximize a work area, the other open work areas are hidden. Use the Restore Down button to show them again.
6	Button to close the work area. The button is available when more than one work area, page 108 is open. It allows you to close the other work areas. You cannot close the main work area. NOTE: Closing the last tab in a work area closes it.
7	Scroll left button. Click the button to view hidden tabs that are to the left of the first tab. The scroll increment is one tab.
8	Active tab.
9	Scroll right button. Click the button to view hidden tabs that are to the right of the last tab. The scroll increment is one tab.

Moving the Workspace (Panning)

In editors (for example, the **Links Editor**), click an empty area of the workspace and hold down the left mouse button. The pointer turns into a hand icon. You can now drag the workspace in any direction by moving the pointer. This is useful to view items that are outside the visible workspace.

In certain explorers (for example, the **Global Templates Explorer**), you can use the following methods to move the workspace.

Action	Direction in which the workspace is moved
Mouse wheel up	Down
Mouse wheel down	Up
Shift + Mouse wheel down	Left
Shift + Mouse wheel up	Right

Magnifying and Reducing the Workspace View

To magnify or reduce the workspace view gradually, use the following method.

Ctrl + Mouse wheel up	Zoom in. The position of the pointer becomes the center of the magnification.
Ctrl + Mouse wheel down	Zoom out. The position of the pointer becomes the center of the reduction.

NOTE:

- A zoom selection menu may be available at the bottom of some workspaces. You may also be able to use the radar view.
- In some workspaces, starting out from the default zoom level, you can only zoom in.
- Zooming out beyond a certain level may switch the display to simplified view, page 878.

Selecting Multiple Items

The method to select multiple items varies depending on the editor or explorer you are working in:

- In editors (for example, template editors), you can select multiple items in the workspace by using either or a combination of the following methods:
 - Select an item with the pointer, press and hold **Ctrl**, and select other items.
 - Press and hold **Ctrl**, click the workspace, and hold down the mouse button. Then, drag the pointer to draw a rectangle. Items that overlap with the rectangle are selected when you release the mouse button. Last, release **Ctrl**.
- In explorers, browsers, and various containers, you can select multiple items by selecting an item with the pointer, pressing and holding **Ctrl**, and selecting other items. To select contiguous items, select the first one, press and hold **Shift** and select the last item. Release **Shift**.

To unselect an item, press **Ctrl** and click the item.

NOTE: While you are still holding **Ctrl**, click a selected item again to unselect it.

Organizing Work Areas

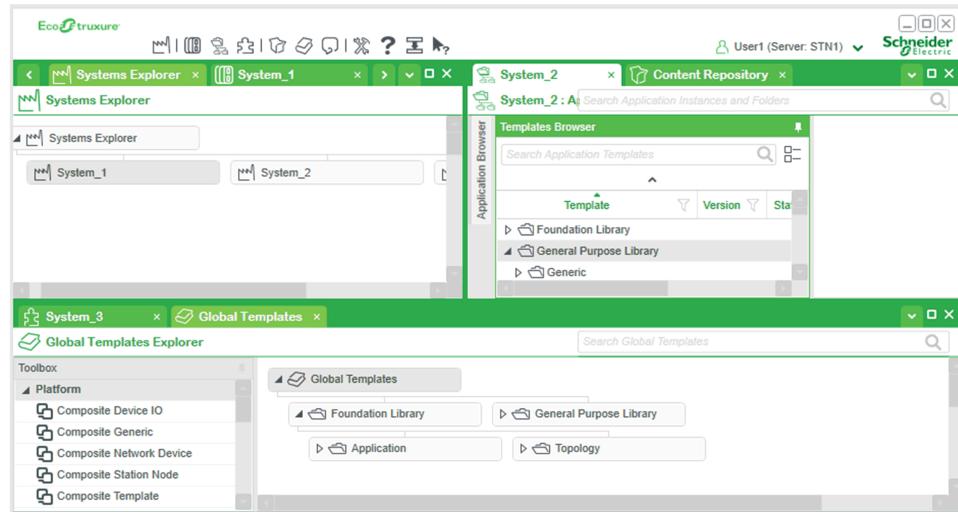
Overview

To work more effectively, you can customize the client UI by creating additional work areas, page 108 left, right, or below the main one. Inside each work area you can open various explorers and editors.

Each work area has its own tabs bar, page 105 and toolbar (if available).

NOTE: You can also divide the work area of explorers, page 109.

The following figure shows an example of an engineering client UI that is divided into three work areas.



Viewing Client Windows on Separate Monitors

To open a new window for the same client, so that you can view it on a separate monitor, proceed as follows.

Step	Action
1	Verify that, at least, two windows are open in the same or separate work areas. For example, the Systems Explorer and the Project Explorer windows.
2	Right-click the tab of a window in the tabs bar and select Open in new Window . Result: The tab opens in a second, separate client window.
3	Drag any of the two client windows to the other monitor.

NOTE:

- You can repeat the procedure from any tab of either client window given that two tabs are open in the work area. (The functionality is not available for tabs that are open inside the **Application Explorer** or **Project Explorer**.)
- Notifications about actions that you perform in the additional client windows appear in the **Notification Panel** of the main client window.

Creating Additional Work Areas

You can create a new work area with an already open explorer by using the docking tabs. The new work area opens on the left, right, or at the bottom of the existing one depending on which docking tab you use.

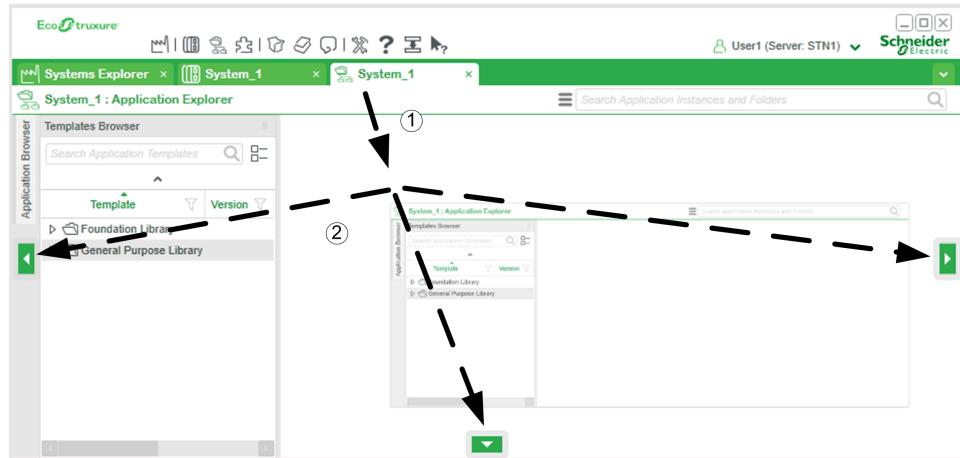
Once you have created several work areas, you can:

- Drag tabs from one work area to the other.

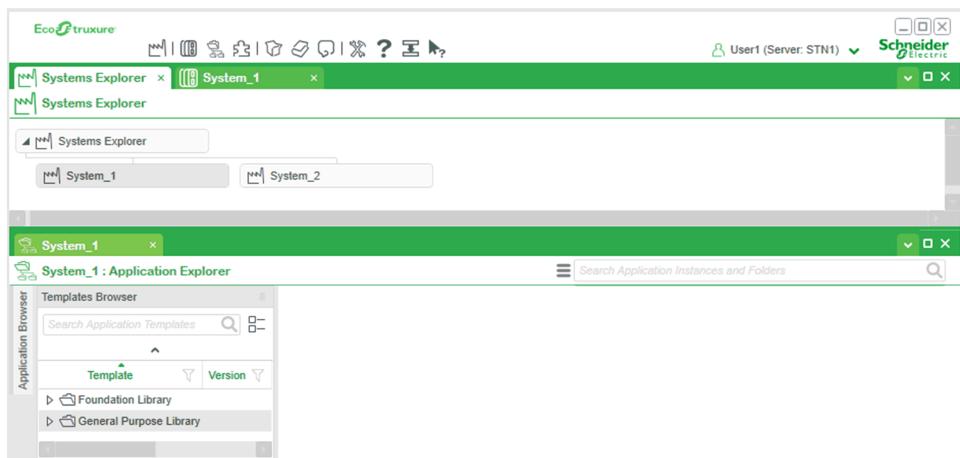
- Reposition work areas by using the docking tabs.

Refer to the topic describing the tabs bar, page 105 for information on how to manage windows inside a work area.

The following figure illustrates the use of the external docking tabs to create a new work area.



Example of result when you use the bottom docking menu.



To open a new work area, proceed as follows.

Step	Action
1	Verify that at least two explorers are open in the work area.
2	Drag a tab from the tabs bar towards the side of the work area where you want to create the new one. Result: The explorer follows the pointer and the external docking tabs appear.
3	Keep dragging the pointer over the docking tab where you want the new work area to open and release the mouse button. Result: A new work area, which contains the explorer opens at the selected location. NOTE: To cancel the operation, drag the pointer away from the docking tab and release the mouse button.

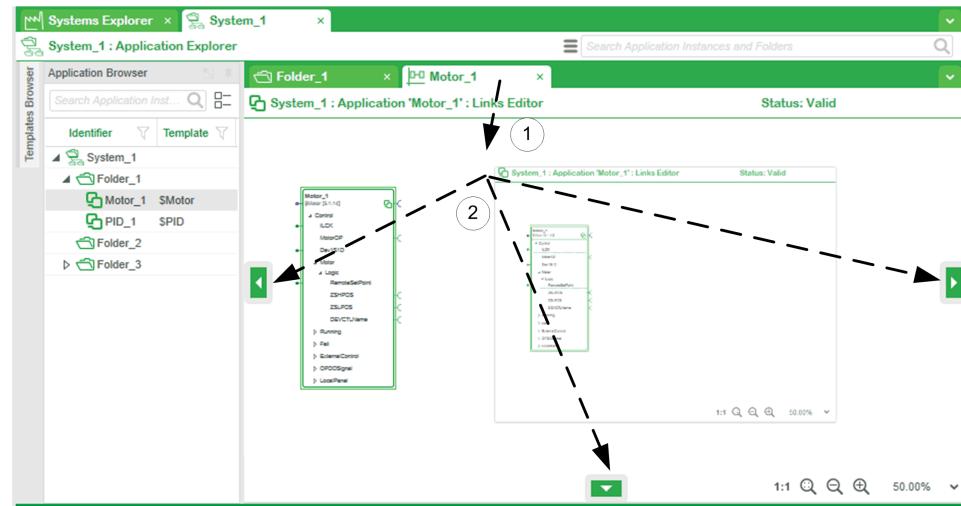
Creating Additional Work Areas in Explorers

When several windows are open in the work area of the **Application Explorer** or **Project Explorer**, you can create additional work areas by dragging either of the windows to a docking tab.

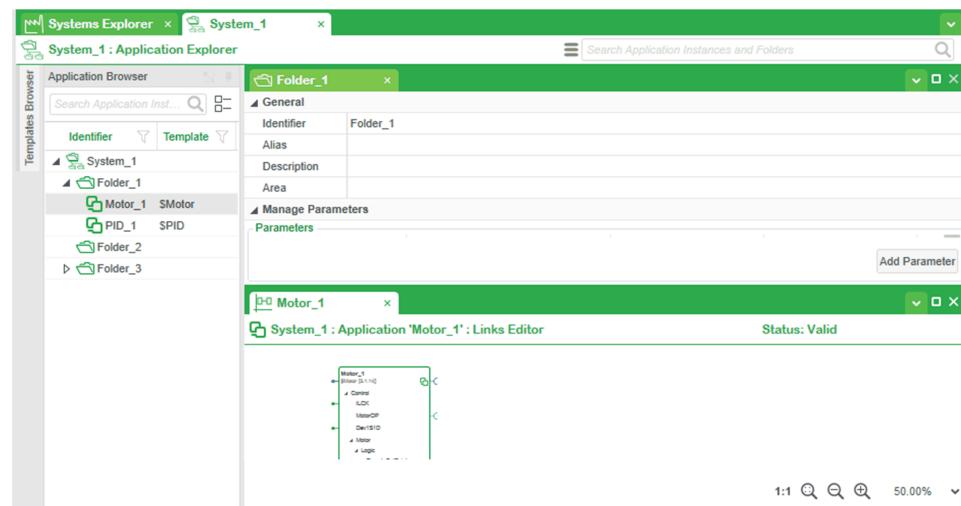
Refer to the topic describing the tabs bar, page 105 for information on how to manage windows inside work areas.

NOTE: You cannot create a new work area with the pane of an explorer. Refer to the topic describing how to work with panes, page 111.

The following figure illustrates the use of the external docking tabs to create a new work area inside the **Application Explorer**.



The following figure shows an example of the **Application Explorer** with two work areas that are open.



To open a new work area, proceed as follows.

Step	Action
1	Verify that at least two windows or editors are open in the work area of the explorer.
2	Drag a tab towards the side of the work area where you want to create the new one. Result: The item follows the pointer and the external docking tabs appear.
3	Keep dragging the pointer over the docking tab where you want the work area to open and release the mouse button. Result: A new work area, which contains the window opens at the selected location. NOTE: To cancel the operation, drag the pointer away from the docking tab and release the mouse button.

Working with Panes

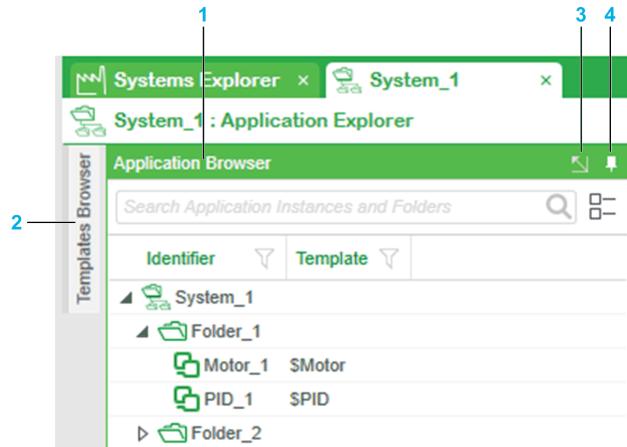
Overview

The software displays panes in various explorers and editors.

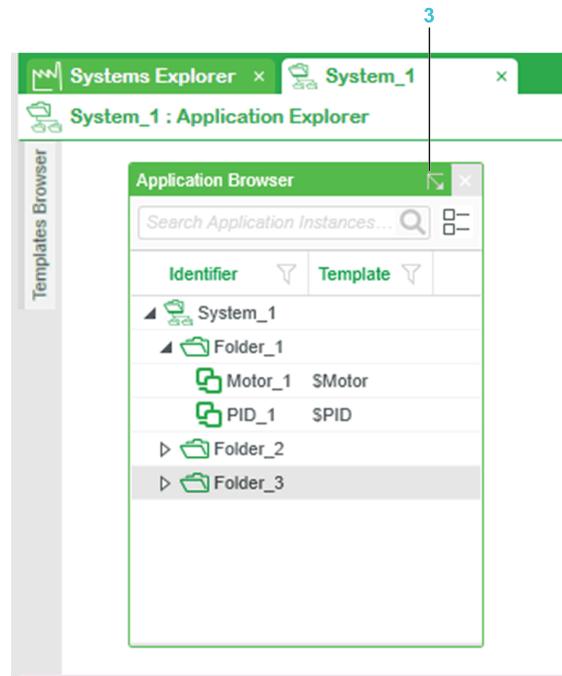
You can minimize them automatically when you do not need to see them, resize, and move them as required.

Pane Description

The following figure shows a partial view of two panes in the **Application Explorer**.



The following figure shows a floating pane. The pin button is not available.



Item	Description
1	Pane that is pinned and always visible.
2	Pane that is minimized to a tab.
3	Extract/retract button. Turns it into a floating pane that you can dock or returns a pane to its previous position (docked or tabbed).
4	Pin button. Lets you show and hide the pane.

Minimizing, Showing, and Restoring Panes

To minimize, show, and restore a docked or tabbed pane, proceed as follows.

Step	Action
1	<p>Click the pin button in the title bar of the pane.</p> <p>Result: The pin button is displayed in a horizontal position and the pane will be minimized to the edge of the screen as a tab.</p>
2	<p>Move the pointer over the tab.</p> <p>Result: The pane is shown entirely while the pointer is inside the pane.</p> <p>NOTE: Click inside the pane so that it remains visible even you move the pointer away from it. Click outside the pane to minimize it again as a tab.</p>
3	<p>Click the pin button again.</p> <p>Result: The pin button is displayed in a vertical position and the pane is restored. If there are several overlapping panes, it may be shown only partially.</p>

Extracting Panes

To make a pane floating, click the extract button or drag it by the title bar.

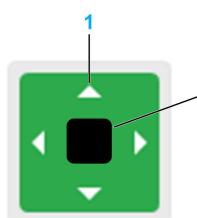
When panes are docked on top of each other so that their tabs appear at the bottom, dragging either pane by its title bar extracts the complete group. To extract only one pane, drag it by its tab or use its extract button.

Docking Panes

You can move docked panes to another area of the workspace or dock floating panes by using the external docking tabs, page 108.

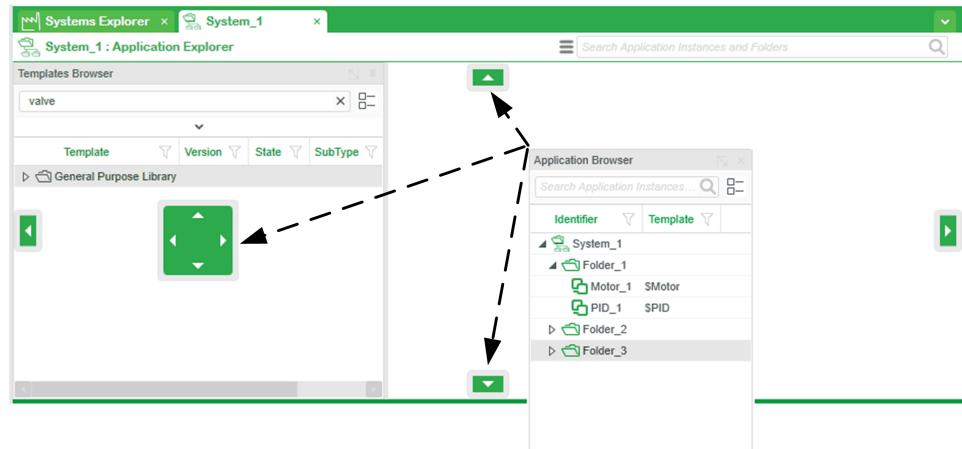
When the workspace contains more than one pane and you drag one pane on top of the other, a docking menu appears in the center of the other pane. It lets you position panes in a relative position to each other.

The following figure shows the docking menu.



Item	Description
1	The up, right, down, and left arrows let you position the pane on top, to the right, below, or to the left of the other one respectively.
2	The center pad positions the pane on top of the other one as a tabbed pane. The center pad appears when you move the pointer over the square docking menu but outside one of the arrows.

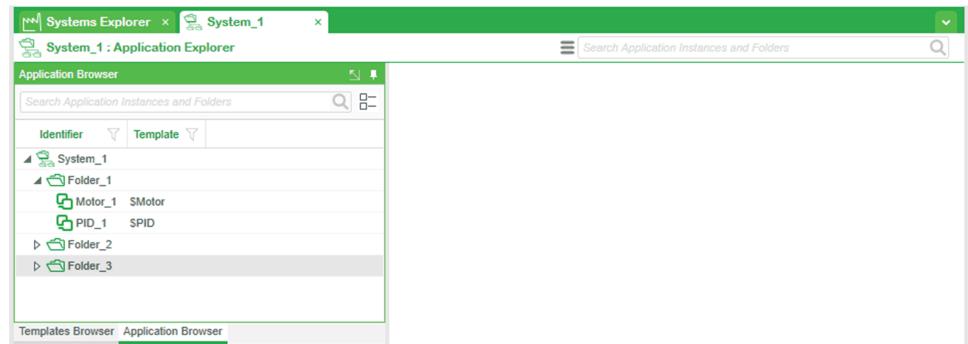
The following figure illustrates the use of the docking menu or external docking tabs to reposition a pane (docked) inside the work area of an explorer.



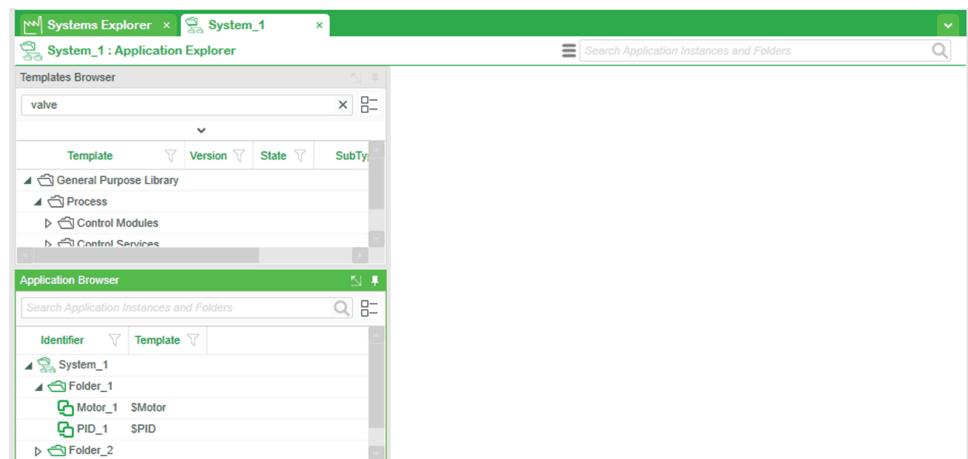
To dock a pane, proceed as follows.

Step	Action
1	Drag the pane by its title bar. If the pane is already docked, extract it, page 112 first. Result: The docking menu and/or the external docking tabs, page 108 appear.
2	Keep dragging the pointer over the docking menu or tab where you want the pane to be positioned and release the mouse button. Result: The pane is docked.

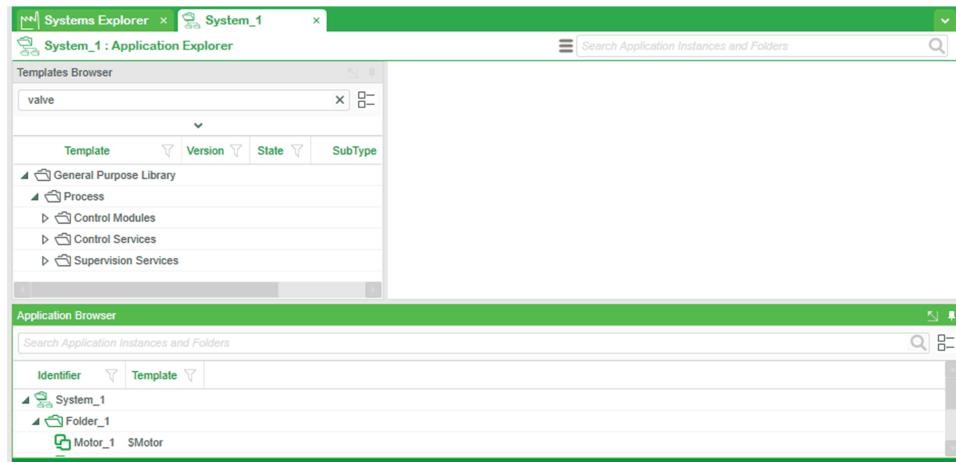
The following figure shows an example of panes that have been tabbed by using the center pad of the docking menu.



The following figure shows an example of a pane that has been docked below another one by using the down arrow of the center pad of the docking menu.



The following figure shows an example of a pane that has been docked to the bottom edge of the engineering client window by using the bottom external docking tab.



Resizing Panes

Typically, panes do not feature a maximize button. Instead, resize them to occupy the entire work area.

To resize a pane, proceed as follows.

Step	Action
1	Move the pointer over an edge of the pane. Result: The pointer turns into a double-ended arrow.
2	Drag the pointer away from the edge. Result: The pane is resized.

Closing Panes

To close a pane, click the close button in the title bar of the pane (when available). To open it again, select the corresponding command from the menu of the explorer or editor.

If no close button is available or if it does not allow closing the pane, minimize the pane, page 112 to a tab.

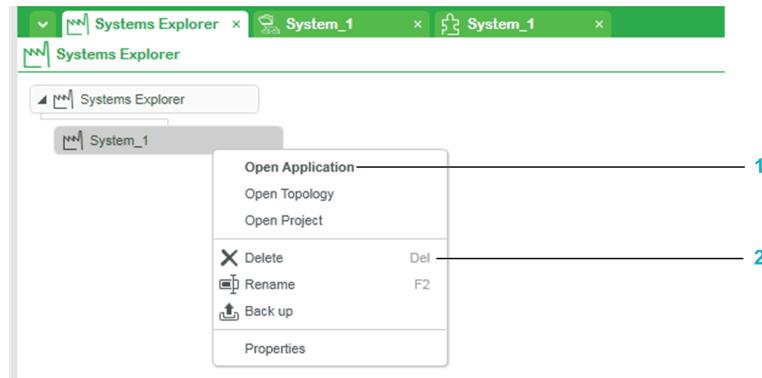
Using Menus

Context Menus

For many items of the UI (for example, folders, instances, elements of projects), right-clicking the item opens a context menu containing several commands.

The commands are described in the topic documenting the item.

The figure shows an example of the context menu that opens when you right-click a system in the **Systems Explorer**, which lets you perform actions on the system.



Item	Description
1	You can execute commands that appear in bold also by pressing Enter when the item is selected or double-clicking the item itself (in this example, the item is System_1).
2	For certain commands, you can also use the keyboard shortcut, page 121 that is indicated to the right of the command when the item is selected.

NOTE: To close a context menu, press **Esc**.

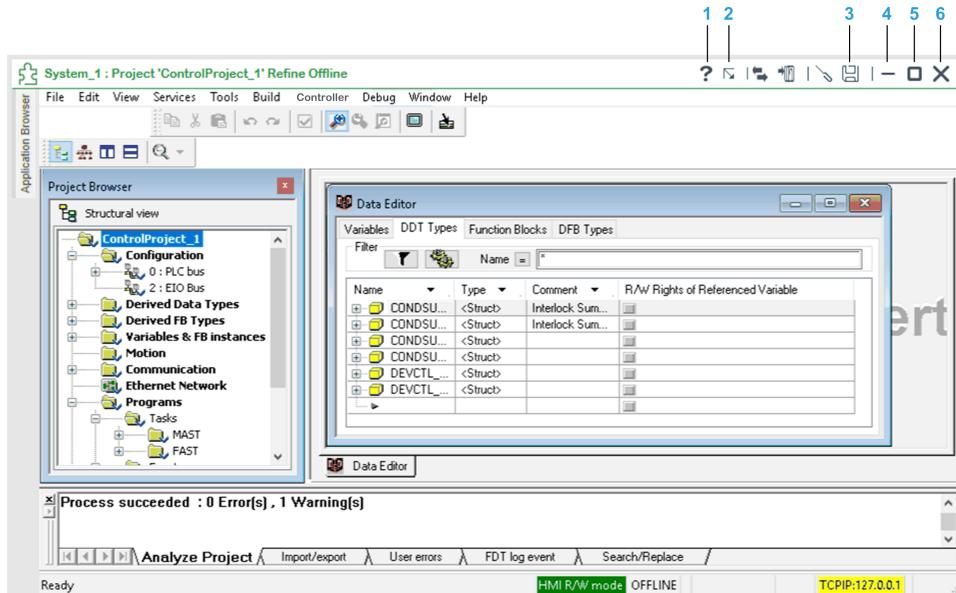
Software Participant Windows

Floating Software Participant Windows

When you open a Participant window, it opens as a floating window and maximized.

You can resize this window, display it on another monitor, or embed it in the client window (except in the **Topology Explorer** where Participant windows cannot be embedded).

You can open several floating windows simultaneously.



The following figure shows an example of a floating Control Participant window. The window is shown restored down.

Item	Description
1	Open the HTML help of EcoStruxure Process Expert
2	Embed the floating Participant window in the client
3	Save changes that were made in the Participant
4	Minimize the floating window
5	Maximize the floating window (or restore down the floating window when it is maximized)
6	Close the floating Participant window

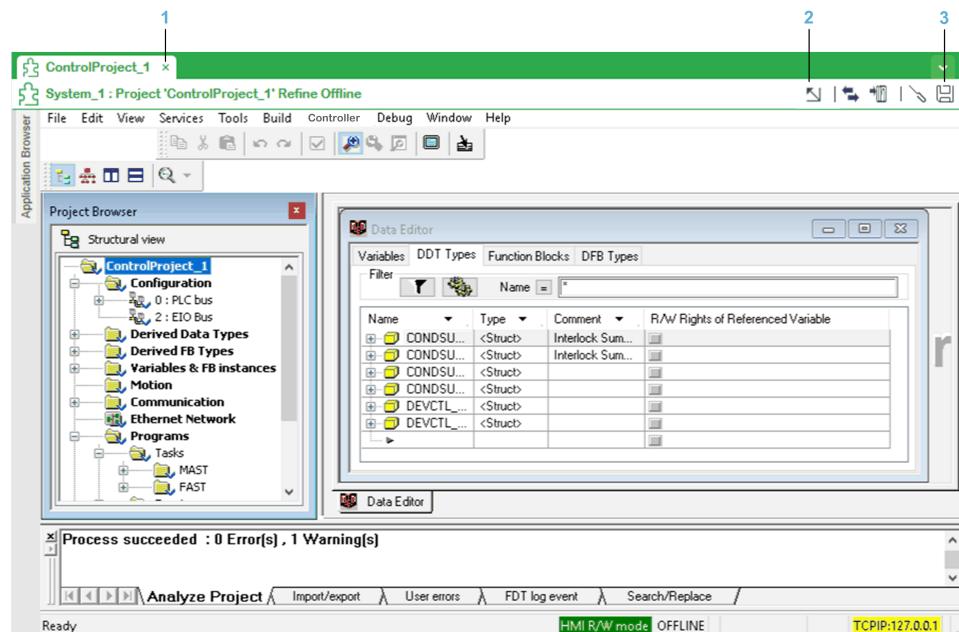
NOTE: The other buttons of the Participant window toolbar are task-specific and are described in the topic documenting the task.

While the floating Participant window is open:

- The embedded Participant window is not refreshed and displays the view that was shown when you extracted the floating Participant window.
- You cannot work in the embedded Participant window nor close it.
- Use the icon in the taskbar to display the client user interface when the floating Participant window is maximized.
- The Windows® snap feature is enabled.

Embedded Software Participant Windows

The following figure shows an example of a Control Participant window, which is embedded in the engineering client by using the restore button. The functionality is not available in the **Topology Explorer**.



Item	Description
1	Close the Participant window
2	Extract and maximize the Participant window
3	Save changes that were made in the Participant

Showing Software Participant Windows on Separate Monitors

To show a software Participant window on another monitor, extract it, page 117 and drag it to the other monitor.

If you experience issues viewing the extracted Participant window on an extended monitor (for example, if the window turns black), try disabling hardware acceleration for Windows® Presentation Foundation (WPF). The required setting may already be present in the registry and you can modify it; otherwise, you need to add it.

To disable hardware acceleration for WPF, proceed as follows. Follow this procedure only if you are experiencing issues viewing an extracted Participant window on an extended monitor.

Step	Action
1	By using the Windows® Registry Editor of the computer on which the client is installed, go to HKEY_CURRENT_USER > SOFTWARE > Microsoft > Avalon.Graphics
2	Add a new registry key by right-clicking the Name column and selecting New > DWORD (32-bit) Value . NOTE: If the DisableHWAcceleration registry key is already present, proceed to step 4 if necessary.
3	Enter DisableHWAcceleration as name.
4	Right-click DisableHWAcceleration , select Modify , and enter 1 as data value.
5	Restart the computer.

Filtering and Sorting Data

Overview

For most columns inside folders and panes, you can open a filter menu that lets you refine data that is displayed according to criteria and/or conditions that you define.

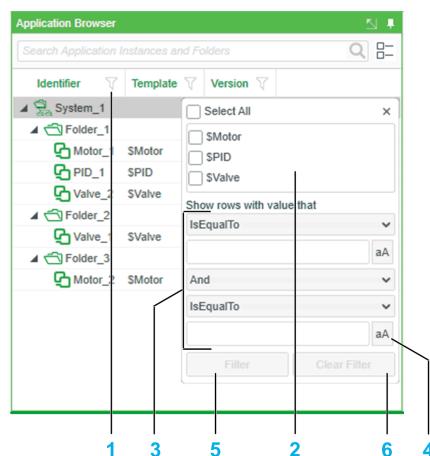
In addition, you can sort data in ascending or descending alphabetical order.

Filters and sorting are reset when you close the pane, window, or editor in which they appear.

NOTE: When a filter is active, commands that you execute at a parent node level may also execute on data that is not displayed. This can be the case, for example, when you execute a command at the folder level in the **Application Explorer** while a filter is applied on the **Identifier** column of the folder and, as a result, not all the instances contained in the folder are displayed.

Filter Menu Description

The following figure shows an example of the filter menu.

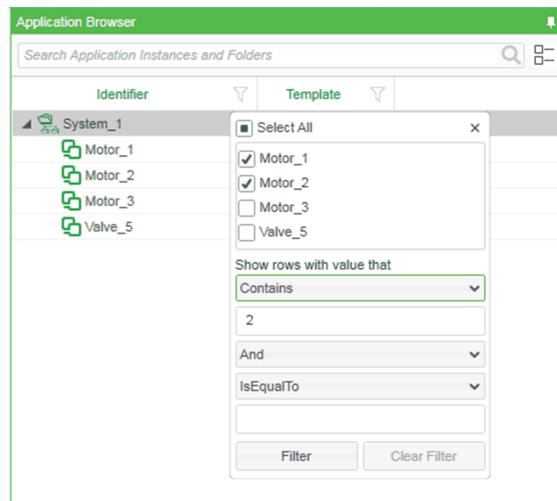


Item	Description
1	<p>Column filter icon.</p> <p>Click the icon to open the filter menu.</p> <p>The icon turns dark to indicate that you have applied a filter on the column.</p>
2	<p>Data field.</p> <p>The field displays in a flat list one of each of all the entries that the column contains. That is, if an entry appears more than once in the column, it is displayed only once in the data field. If the column contains only partial data because, for example, a keyword search has been performed, the data field displays also only the partial data.</p> <p>Select one or more entries. Only entries that you select will be displayed when you apply the filter.</p> <p>NOTE: If the column contains a tree view (for example, the Templates Browser pane of the Application Explorer), the content of the data field depends on the type of lowest-level child elements shown in the column:</p> <ul style="list-style-type: none"> • A single type of child elements (for example, only templates): The data field displays only child elements that are not part of the tree structure (for example, only templates but no folders). • Several types of child elements (for example, the Name column of the Import window in the Projects Explorer for a Control project, which contains services and sections as lowest-level child elements): The data field displays only the parent nodes (for example, the system root folder).
3	<p>Expression builder.</p> <p>Lets you build custom filter expressions by selecting one or two conditions and an operator from a list.</p>

Item	Description
	<p>The conditions apply to any data contained in the column even if it is not shown in the data field. The list of available conditions varies and is adapted to the data type of the column.</p> <p>NOTE: If you build an expression and select an entry in the data field, your selection is used as an additional condition in the expression with the AND operator. If you select several entries in the data field, each one is treated as OR condition before the result is used as an additional condition.</p>
4	<p>Match case button.</p> <p>Click the button to enable case-sensitive filtering as an additional condition.</p> <p>The button appears only for columns showing data of String type.</p>
5	<p>Click the button to apply the filter to the column.</p> <p>NOTE: When the filtered data is modified (for example, new objects have been created or imported, objects have been renamed), the filters do not apply to the modified data. To apply them, open the filter menu again and click Filter.</p>
6	<p>Click the button to clear the filter for the column.</p> <p>NOTE: To clear filters for all columns, right-click any column header and select Clear Filters.</p>

Example

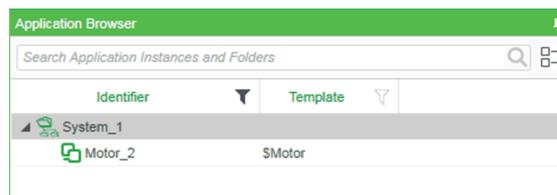
The following figure shows a filter example consisting of two items selected in the data field and one expression. It is used to filter the contents of the **Identifier** column.



The resulting filter expression is:

(Motor_1 OR Motor_2) AND Identifier contains 2.

The following figure shows the search result once the filter is applied; instances for which the expression is true.



Using Filters

To use filters, proceed as follows.

Step	Action
1	Click the filter icon of the row that contains the data that you want to filter. Result: The filter menu opens.
2	Select entries in the data field or create a filter using the expression builder.
3	Click Filter . Result: The column displays data that matches the criteria and the filter icon turns dark. NOTE: When columns contain tree views, the parent structure of filtered items is shown.
4	Close the filter menu.

NOTE: You can use filters for several columns simultaneously. Repeat the above procedure on each column.

Sorting Data

The sorting feature is represented by a green triangle inside the column header.

Click a column header to enable sorting of the data that the column contains.

Clicking the column header repeatedly toggles between the following states:

- Arrow up: Sorts in ascending alphanumerical order.
- Arrow down: Sorts in descending alphanumerical order.
- No arrow: No sorting. Data is displayed by using the default display order defined for the column.

NOTE:

- Hold **Shift** to enable sorting of multiple columns.
- To clear sorting on all columns, right-click any column header and select **Clear Sorting**.
- If a rule has been defined to display contents of a column, enabling the sorting feature does not override this rule. For example, in the **Application Browser** of the **Application Explorer** folders are shown first, followed by instances. Applying sorting sorts only items within their respective category. It does not change the display order of categories.

Clearing All Filters and Sorting

To clear filters and sorting for all columns, right-click any column header and select **Clear All**.

Keyboard Shortcuts

Available Keyboard Shortcuts

You can use the following keyboard shortcuts in place of the command in the context menu or the button.

Keyboard shortcut	Corresponding command or button
Alt+A	Navigation button, page 101. Opens or shows the Application Explorer for the system.
Alt+P	Navigation button, page 101. Opens or shows the Project Explorer for the system.
Alt+T	Navigation button, page 101. Opens or shows the Topology Explorer for the system.
Ctrl+A	No corresponding command. Use this keyboard shortcut to select the entire contents of a folder or container, such as instances or facets.
Ctrl+C	Copy
Ctrl+F	Find (Global Templates) Places the cursor in the search field of the Global Templates Explorer.
Ctrl+L	Edit Links. Opens the Links Editor of the Application Explorer .
Ctrl+V	Paste
Del	Delete
F2	Rename
F4	Settings (Global Templates)
F12	Read-only (Global Templates)
Shift+F12	Edit (Global Templates)

Engineering a System

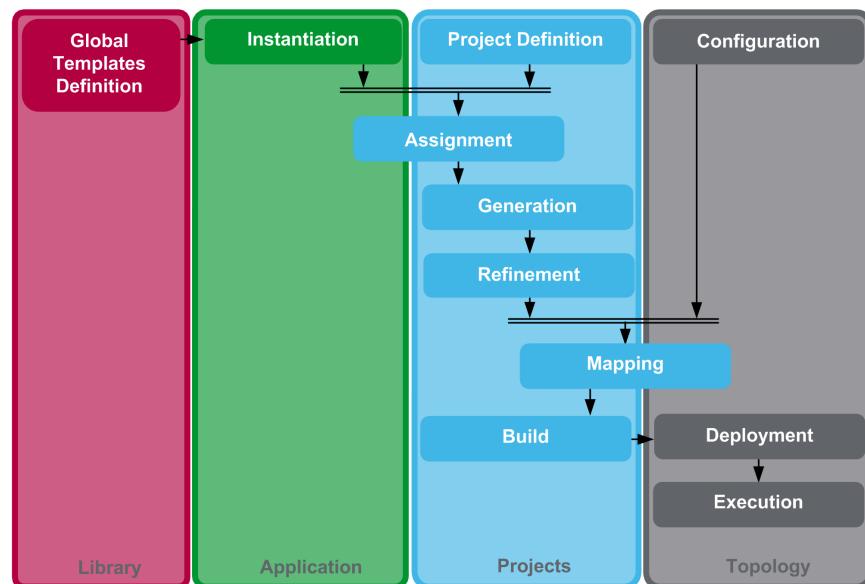
What's in This Part

Quick Reference Guides	123
Managing Systems	132
Instantiation Stage	152
Project Definition Stage	254
Assignment Stage	328
Generation Stage	383
Refinement Stage	434
Configuration Stage	498
Mapping Stage	576
Build Stage	628
Deployment Stage	648
Execution Stage	706
Engineering M580 Safety Systems	759
Managing Communication	779

Overview

This part describes how to complete the different stages of the system engineering life cycle to engineer a system.

The following figure shows the stages of the system engineering life cycle.



Quick Reference Guides

What's in This Chapter

System Creation Workflow	124
Task Quick Reference.....	128

System Creation Workflow

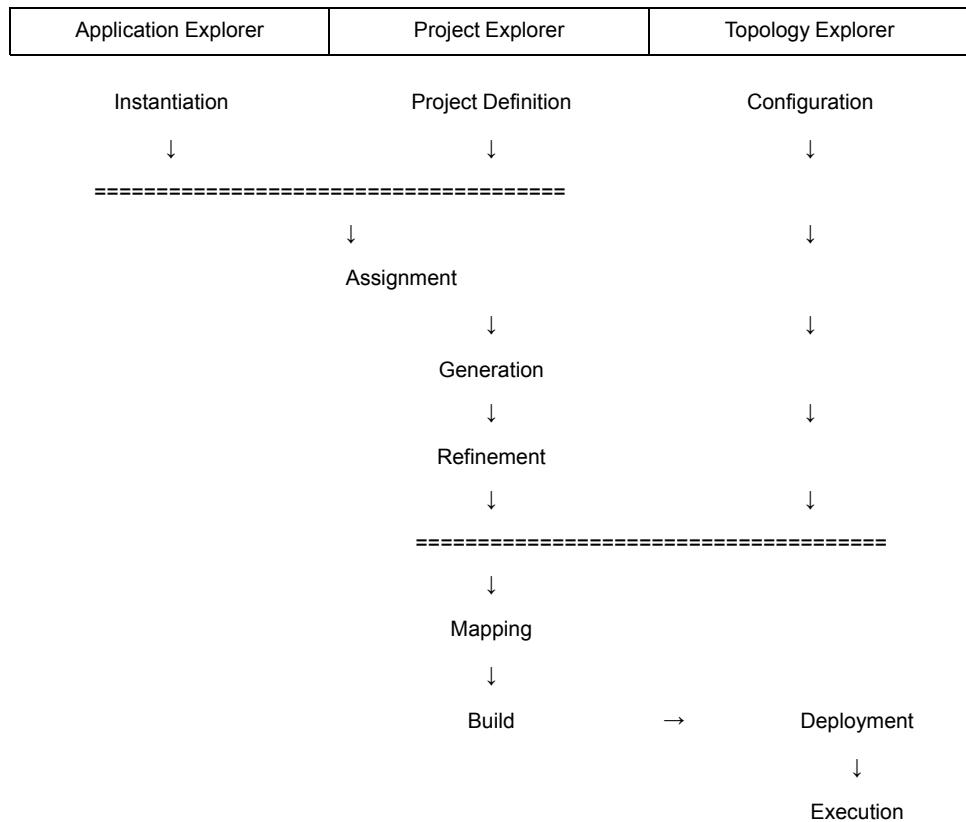
Navigation

This page shows the various stages of the system engineering life cycle and which Explorer is used to complete them.

- Click an Explorer to view the main tasks of the workflow.
- Click a stage for detailed information on how to complete it.

NOTE: In the HTML help, use the **Back** button in the toolbar of the help window to revert to this page.

System Engineering Life Cycle



NOTE: Specific transversal workflows are also provided describing how to implement communication, page 779.

System Creation Workflow

The following workflow shows the main tasks that you need to perform to complete the system engineering life cycle. Click a task to navigate to the topic describing how to complete it.

Systems Explorer

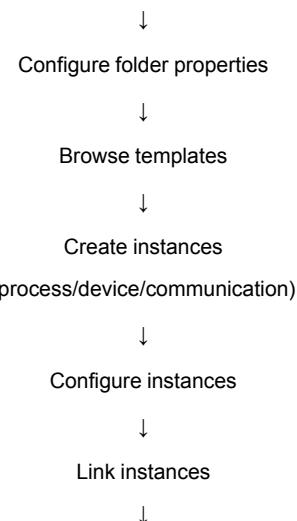
Create system



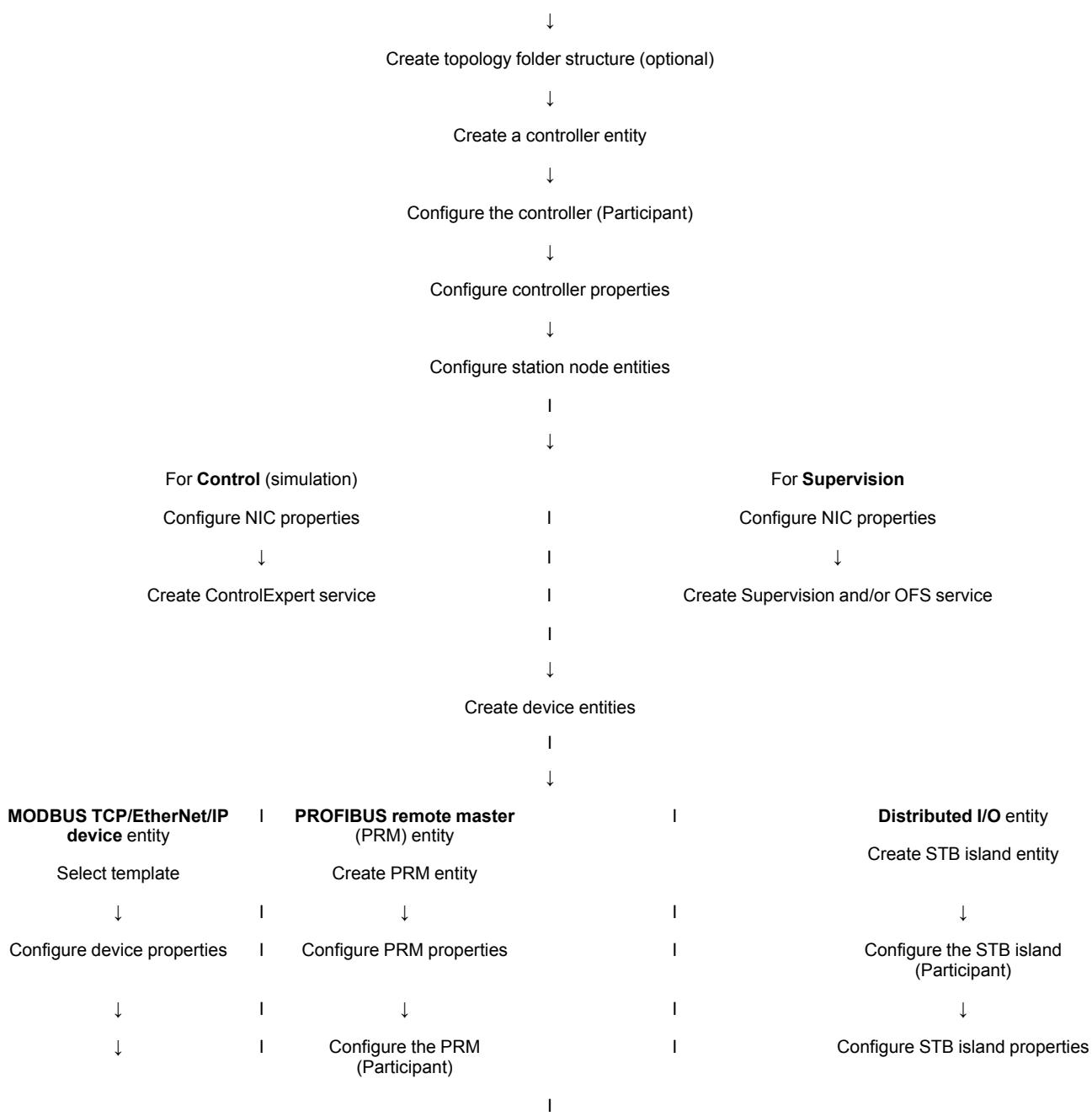
Application Explorer

Create application folder structure





Topology Explorer





Create an Ethernet network



Create physical connections



Project Explorer



Control

Create project



Create sections in task folders



Assign facets to sections



Generate project or sections



Refine project or sections (Participant)



Create executable



Perform service mapping



Perform communication mapping



Perform hardware mapping



Build project



View the built project

Supervision

Create project



Assign facets to project container



Create Supervision pages



Edit pages / assign genies (Participant)



Generate project



Refine the project (Participant)



Create executable



Perform service mapping



Perform communication mapping



Build project



Deploy the project

(from the executable)



Topology Explorer



Control

(deployment and execution)



Start the simulator (if applicable)



Select the project and executable

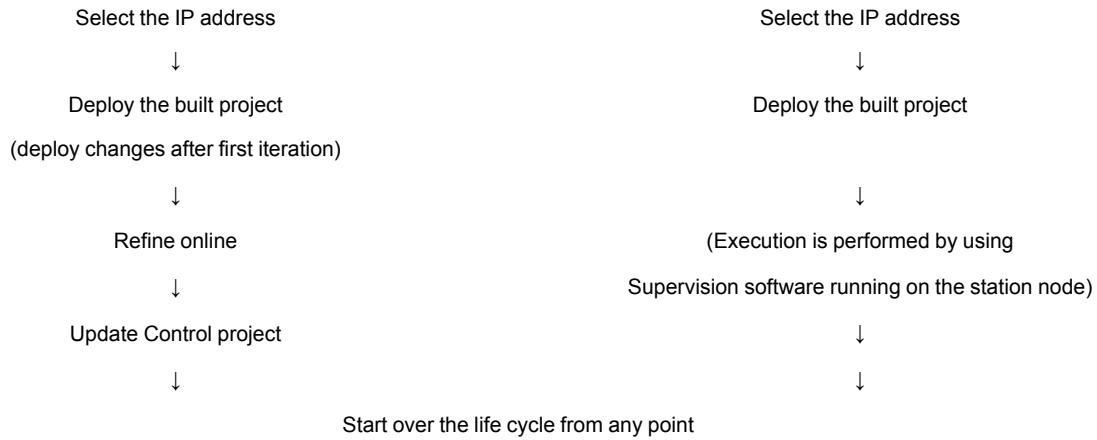
Supervision

(deployment from the station node)



Select the project and executable





NOTE: Specific transversal workflows are also provided describing how to implement communication, page 779.

Task Quick Reference

Overview

The tables in this topic provide an overview of the main tasks that you can perform by using the software at each stage of the system engineering life cycle, page 33 and where to find information explaining how to complete the task.

Stages are grouped by explorer.

Systems Explorer Task Quick Reference

Task
Creating systems, page 132 (and configuring runtime navigation service (RTNS) parameters)
Creating system backups, page 142
Scheduling system backups, page 145
Restoring systems, page 147
Generating system engineering documentation, page 927
Enabling and managing the System Access Password for a system., page 137

Application Explorer Task Quick Reference

System Engineering Life Cycle Stage	Task
Instantiation stage, page 152	Browsing templates, page 158 Creating instances, page 171 Browsing instances and folders, page 161 Counting instances, page 174 Configuring instances, page 182 (and associating documents and URLs to instances for use with RTNS) Linking instances, page 193 Viewing facet assignment, page 379 Managing instances, page 203 (copy/paste, move, update/replace templates) Exporting and importing application objects, page 219

Project Explorer Task Quick Reference

System Engineering Life Cycle Stage	Control Projects	Supervision Projects
Project Definition stage, page 254	Creating projects, page 260 and M580 safety projects, page 759	Creating projects, page 290
	Managing projects and project sections, page 265 (order, export, import, manage data backup files)	Creating project components, page 294
		Managing Projects, page 306 (export, import)
		Adding and using Supervision page templates, page 346
Assignment stage, page 328	Creating sections, page 335 (automatic and manual)	Creating containers (automatic and manual)
	Assigning facets, page 362	Assigning facets, page 364
	Viewing the application structure in projects, page 371	
	Managing assigned facets, page 376 (view assignments, unassign, move, change order)	
Generation stage, page 383	Generating for the first time, page 388	
	Enabling automatic generation, page 263	Generating equipment names, page 425
	Generating after changes, page 416 (auto-assign and generate, regenerate)	
	Managing generated facets, page 431 (unassign, reassign, unlink, move)	
	Managing versions of types, page 396	–
Refinement stage, page 434	Refining projects, page 436 (viewing the application structure, viewing code generated by facets, managing device DDTs, creating LL984 segments)	Refining projects, page 494
	Checking consistency, page 449	Editing Supervision pages, page 495
	Managing Peer to Peer communication, page 462	Editing advanced project settings, adding included projects, attaching user files, page 495
	Creating network variables, page 452 (legacy)	
Mapping stage, page 576	Creating Control executables, page 578	Creating executables, page 619
	Mapping services, page 580	Mapping services, page 622
	Mapping communication, page 582	Mapping communication, page 624 (and connecting Modbus devices to Supervision)
	Mapping network variables, page 594 (legacy)	
	Mapping hardware, page 602	
	Exporting/Importing hardware mappings, page 609	
Build stage, page 628	Building projects, page 630 (also, generate and build)	Building projects, page 643 (also, generate and build)
	Viewing the built Participant project, page 634	Downloading project files, page 646
Deployment stage, page 648	Refer to the Topology Explorer task quick reference.	Deploying built projects, page 694 and deploying changes, page 703 (to all the station nodes that are service mapped to the project executable)

Topology Explorer Task Quick Reference

System Engineering Life Cycle Stage	Control Projects	Supervision Projects
Configuration stage, page 498	Configuring controllers, page 507 and M580 safety controllers, page 759 Configuring station nodes, page 516 (for Supervision/simulation) Configuring Modbus TCP and EtherNet/IP devices, page 531 Configuring PROFIBUS PRM Masters and Decentralized Peripherals, page 527 Configuring STB Islands, page 523 Configuring Ethernet networks, page 538 Connecting to Ethernet networks, page 540 Managing the Topology, page 548 (export, import, update templates) Exporting and importing I/O devices to/from CSV file, page 564 Managing communication, page 779 (Modbus TCP Ethernet implicit (I/O scanning, peer to peer communication)/explicit, EtherNet/IP explicit, Modbus serial)	
Deployment stage, page 648	Using the emulated simulator, page 654 Managing controller passwords for deployment and execution operations, page 658 Deploying built projects, page 649 Re-deploying projects, page 672 Deploying project changes, page 682 (and undoing online changes) Deploying controller data, page 677	Deploying built projects, page 694 and deploying changes, page 703 (to specific station nodes)
Execution stage, page 706	Starting/stopping controllers, page 713 Synchronizing redundant controllers, page 719 Refining online, page 727 (and importing generated changes, viewing the application structure) Updating Control projects, page 748 Backing up controller data, page 733	Refer to Executing the Deployed Supervision Project (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide)

Global Templates Explorer Task Quick Reference

System Engineering Life Cycle Stage	Tasks
Global Templates, page 837	Managing Global Templates, page 881 (save, copy, export/import, purge) Updating, duplicating, and replacing Global Templates, page 898 Creating Global Templates (see EcoStruxure™ Process Expert, Global Templates Reference Manual) Templatizing Supervision animated graphics (see EcoStruxure™ Process Expert, Global Templates Reference Manual) Enabling and managing the Control facet template password, page 849

Content Repository Task Quick Reference

Tasks
Creating content containers (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide) (and editing, copying, moving, exporting and importing them to set up runtime navigation services (RTNS))

Software Participant Task Quick Reference

Tasks
Configuring the number of Control Participant instances (see EcoStruxure™ Process Expert, Installation and Configuration Guide) that you can open per computer .
Installing extensions in the Control Participant (see EcoStruxure™ Process Expert, Installation and Configuration Guide) (DTMs, GSD and EDS files, hotfixes)
Opening the help of a Participant, page 93, OPC Factory Server, and OPC UA Server Expert

Managing Systems

What's in This Chapter

Organizing Systems	133
Creating Systems	135
Managing System Access Passwords.....	138
Creating System Backups	142
Scheduling System Backups	145
Restoring Systems	147

Overview

This chapter describes how to manage systems by using the **Systems Explorer**. Creating a system is prerequisite to starting the system engineering life cycle.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Organizing Systems

Overview

Systems that you create are stored in the database of the system server.

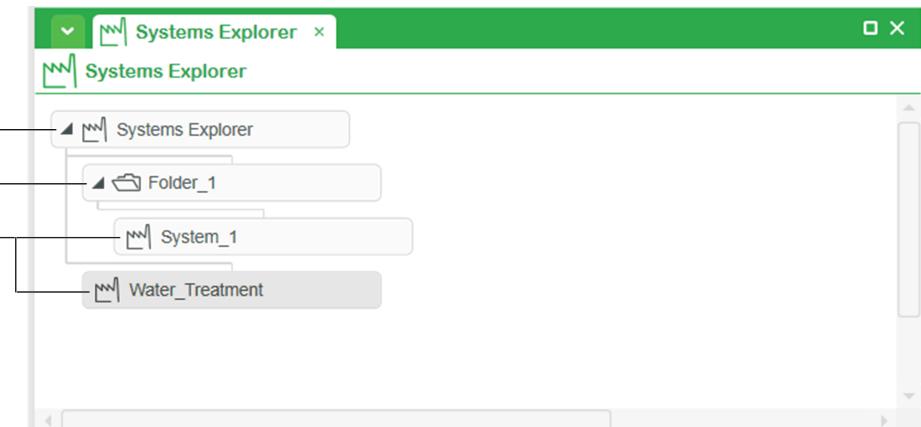
You can access a system from any client that is connected to the server of this infrastructure.

Accessing the Systems Explorer

To access the **Systems Explorer**, click the **Systems** icon in the toolbar, page 101.

Systems Explorer Window

The following figure shows the **Systems Explorer** window.



Item	Description
1	Systems Explorer root folder
2	System folder
3	Systems

Creating System Folders

To organize systems, you can create system folders, which can contain one or more systems.

To create a system folder, proceed as follows.

Step	Action
1	Right-click the Systems Explorer root folder and select Create Folder . Result: The software creates a system folder with the default name in edit mode, and displays it in the tree view. NOTE: To create a subfolder, repeat step 1 at the system folder level.
2	To change the name, type it; otherwise click outside the folder to accept the default name.

System Folder Actions

Right-click the root folder or any system folder to open a context menu with the following commands.

When you select...	The software...
Create Folder	creates a system folder or subfolder.
Create System	creates a system.
Delete	deletes the folder. The command is not available from the root folder. NOTE: The command is not available if the folder is not empty.
Rename	allows you to enter a new folder name. The command is not available from the root folder. NOTE: You can also click the folder twice to edit its name.
Restore	opens the Restore window, which allows you to select a system backup file (.sbf) to restore a system, page 147 in the database.

Creating Systems

Creating a System

To create a system, proceed as follows.

Step	Action
1	<p>Right-click the Systems Explorer root folder or a system folder and select Create System.</p> <p>Result: The software creates a new system with the default identifier in edit mode, and displays in the tree view.</p>
2	To change the identifier, type the new one; otherwise click outside the system to accept the default name.

System Actions

Right-click a system to open a context menu with the following commands.

Command	Description	
Open Application ⁽¹⁾	Opens the Application Explorer , which allows you to start the Instantiation stage, page 152 or access the application of an existing system.	
Open Topology ⁽¹⁾	Opens the Topological Explorer , which allows you to start the Configuration stage, page 498 or access the topology of an existing system.	
Open Projects ⁽¹⁾	Opens the Projects Explorer , which allows you to start the Project Definition stage, page 254 or access the Control and Supervision projects of an existing system.	
System Access Password	<p>Opens a submenu containing the following commands.</p> <p>The menu is only available if the System Access Password property, page 137 of the system is enabled.</p>	
	Manage Password	Opens the Manage Password dialog box, which lets you set and change the system password, page 139.
	Clear Password	Opens the Clear Password dialog box, which lets you remove the system password, page 140.
	Forgot Password	Opens the Forgot Password dialog box, which contains an authentication code that you need to reset the forgotten password, page 140.
Delete	<p>Deletes the system including all of its contents.</p> <p>The command is inactive if either an application, topology, or project window of the system is still open.</p>	
Rename	<p>Allows you to enter a new system identifier.</p> <p>NOTE: You can also click the system slowly twice to edit its identifier</p>	
Back Up	Opens the Back Up window, page 142, which allows you to create a backup file of the system.	
Generate Report	<p>Opens the Generate Report window, which lets you create and save to disk a report, page 927 that contains engineering information about the selected system or one or more of its domains (such as, topology, Control projects), including Participant project documentation.</p> <p>All windows in the engineering client, except the Systems Explorer, need to be closed to use the command.</p>	
Settings	Opens the Settings window, page 136, which lets you view and configure various system-related settings.	
Properties	Opens the Properties, page 137 window of the system.	
(1) You can also use the navigation buttons, page 101 located in the toolbar of the engineering client.		

System Settings Window

The **Settings** window opens when you select the **Settings** command in the context menu of a system.

Section	Description
Document Viewers	Lets you enter the command line that the software executes for a given document type. The command is executed when you open, in the operation client, the document that is associated to an instance by using the HyperLink service, page 184.

System Properties

To open the **Properties** window of the system, right-click the system and select **Properties**.

You can view and/or edit the following items.

Item	Description
Identifier	The identifier must be unique on the system server. Default value: System_n where n is an incremental number starting at 1.
Description	Optional. You can enter a description of the system with free form text. The description you enter appears in a tooltip when you move the pointer over the system. Default value: Blank
Market	Optional. You can enter a description of the market that the system pertains to with free form text. Default value: Blank
Customer	Optional. You can enter a description of the customer that the system pertains to with free form text. Default value: Blank
Site	Optional. You can enter a description of the site that the system pertains to with free form text. Default value: Blank
Created By	Logged-in user who has created the system.
Creation Date Time	Date and time of the system creation.
Modified By	Logged-in user who has modified the system properties.
Modification Date Time	Date and time of the last modification to the system properties.
Change Description	Optional. You can enter a description of the changes to the system with free form text. Default value: Blank

Item	Description
Security	<p>Opens a submenu containing the following properties.</p> <p>The properties are enabled and read-only when the Control application and facet template password protection setting (see EcoStruxure™ Process Expert, Installation and Configuration Guide) is enabled. The setting can be configured in the Control Passwords section of the System Server Configuration Wizard.</p>
Controller Access Password at Creation (formerly Optional Security Services By Default)	<p>The setting has an impact on the default value of the Controller and Simulator password property of controllers, page 509 and station nodes acting as simulator, page 520 that you create in the system:</p> <ul style="list-style-type: none"> Possible values: Enabled (default): Sets their property to Enabled by default. Disabled: Sets their property to Disabled by default. <p>NOTE: Modifying the setting does not change the value of the password property of existing station nodes and controllers of the system.</p> <p>NOTE: This setting does not impact the Safety password property of M580 safety controllers, page 770.</p>
System Access Password	<p>Lets you enable the System Access Password context menu of the system, page 135, which lets you set and manage a password, page 138 at the system level.</p> <p>Possible values:</p> <ul style="list-style-type: none"> Enabled: After enabling the setting, you are required to set a password, page 138 for the system in the Manage Password dialog box to open one of its explorers or generate a report. Disabled (default): You cannot set a password for the system. <p>To disable the property when a password has been set, you must enter the password, which also clears it, page 140. To perform this operation, all other clients connecting to the system server must be closed and no other tab must be open in this engineering client.</p>

Managing System Access Passwords

Overview

The system access password that you manage for each system by using its context menu has the following purpose:

- It is used as application password in the Control Participant projects of the system (Projects and Topology domains).
- It enables file encryption for these Control projects.

NOTE:

For an overview of the entire Control project password management functionality, refer to the topic describing the system and Control facet template passwords (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Prerequisites

The following are prerequisites to setting and managing the system access password:

- The **System Access Password** property, page 137 of the system is enabled.
- No other tab than the **Systems Explorer** is open in the engineering client.
- No other client is connected to the system server.

Scope of the System Access Password Protection

You are required to enter the password to perform the following actions:

- Open an explorer of the system.
- Generate a system documentation report.

NOTE: After you have entered password once, you are not required to enter it again until you close the engineering client or restart the system server.

System Access Password Requirements

To be valid, the system access password must contain the following:

- Between 8 and 16 characters.
- At least one uppercase letter from the classical Latin alphabet (A...Z).
- At least one lowercase letter from the classical Latin alphabet (a...z).
- At least one base-10 digit (0...9).
- At least one special character (~, !, @, \$, %, ^, &, *, _, +, -, =, ` , \, (,), [,], :, ;, ', <, >, {, }, :, #).

Setting the System Access Password

You can set the system access password either way:

- By using the **Manage Password** command of the **System Access Password** context menu of the system.
- By opening an explorer of the system (for example, the **Application Explorer**).

Changing the System Access Password

To change the system access password, proceed as follows.

Step	Action
1	In the Systems Explorer , right-click the system and click System Access Password > Manage Password . Result: The Manage Password dialog box opens.
2	Enter the current password in the Current Password text box.
3	Enter the new password and confirm the new password in the corresponding text boxes.
5	Click OK . Result: The new password is applied to the Control Participant projects of the system.

Clearing the System Access Password

Clearing the system access password has the following impact:

- Clears the application password of Control Participant projects of the system.
- Disables file encryption for these Control Participant projects.

NOTE: You cannot clear the password when the **Control application and facet template password protection** setting (see EcoStruxure™ Process Expert, Installation and Configuration Guide) is enabled at the system server level.

To clear the system access password, proceed as follows.

Step	Action
1	In the Systems Explorer , right-click the system and click System Access Password > Clear Password . Result: The Clear Password dialog box opens.
2	Enter the current password in the Current Password text box.
3	Click OK . Result: The Confirm Clear Password dialog box opens.
4	Click OK . Result: The password is cleared.

Resetting a Forgotten System Access Password

To reset the system access password, proceed as follows.

Step	Action
1	In the Systems Explorer , right-click the system and click System Access Password > Forgot Password . Result: The Forgot Password dialog box opens and an authentication code is displayed.
2	Copy the authentication code, contact Schneider Electric support, page 104, and provide the code. Result: You receive a temporary password.
3	Right-click the system and click System Access Password > Manage Password.. Result: The Manage Password dialog box opens.
4	Follow the procedure to change the password, page 139 and enter the temporary password in the Current Password text box. NOTE: If you attempt to enter the temporary password when prompted for the actual password (for example, when opening an explorer), you are asked to change the password first. You cannot clear the forgotten password by using the temporary password.

Password Management When Exporting and Importing Control Projects

When you export a Control project, its application password is removed and file encryption is disabled.

When you import a Control project, an application password is set and file encryption is enabled if the **System Access Password** property, page 137 of the system in which you import is enabled.

Password Management When Importing Control Expert Projects

When you import a Control Expert project, an application password is set and file encryption is enabled if the **System Access Password** property, page 137 of the system in which you import is enabled.

Password Management When Importing and Exporting Application and Topology Objects

When you export instances or topological entities, if the export file contains Control constituents (.stu), their application password is removed and file encryption is disabled.

When you import instances or topological entities, an application password is set and file encryption is enabled for any Control constituents (.stu) if the **System Access Password** property, page 137 of the system in which you import is enabled.

File Encryption Management for Already Deployed Control Projects

If you enable the **System Access Password** property for the first time and the **Controller** password has already been set for a controller, the already deployed Control project is not impacted. That is, file encryption is not enabled.

This can be the case, for example, after restoring the database of a previous version.

To apply file encryption to the deployed project, you need to deploy the Control project again after setting the system access password. This requires stopping the controller.

Password Management When Backing Up and Restoring Systems

If you backup a system, page 142 for which a system access password is set, the setting and the password are exported.

When you restore this system, the system access password protection remains in effect.

If you restore a system for which no system access password is set (setting disabled), system access password protection will be enabled, and you will be prompted to set a password, if the setting is enabled at the system server level.

Password Management When Saving Control Participant Project Files from the Content Repository

When you save to disk, page 939 a Control Participant project file (.stu) from the content repository (of a Control project or controller) and you have set a system access password for the corresponding system, the application password protection and file encryption remain in effect for this project file.

You can change its application password but you cannot remove it. Also, you cannot disable file encryption.

Creating System Backups

Overview

You can create a backup file (.sbf) of any system that exists in the **Systems Explorer**. This allows you to do the following:

- Save your engineering work on demand so that your data is preserved, for example, in case of a disk problem on the computer on which the system server is installed (refer to the topic describing the contents of a system backup for details).
- Restore your work to an earlier point in time.
- Deliver a system to a site.

You can create system backups in the following ways:

- Manually, by using the **Back Up** command from the context menu of the system in the **Systems Explorer**.
- By using the **System Backup Scheduler**, page 145 command from the **Settings** menu of the system server (requires system server access rights (see EcoStruxure™ Process Expert, Installation and Configuration Guide)).

A backup of a system cannot be created if a component of the system is open in an engineering or operation client (for example, if the **Application Explorer** is open or if data of the system is accessed by using runtime navigation services). Also, during a system backup, no user can access it from an engineering or operation client.

You can start a manual backup while another system is being backed up. You can do this from the same or separate engineering clients.

Version Compatibility

Backup files are version dependent. You cannot use them to restore systems in a different version of the software.

When you back up a system, the version of the software is automatically recorded and is displayed when you restore the system.

NOTE: Backup files are forward compatible with subsequent releases within the same year (*R** versions) and service packs of a same version of the software only if the Participant version is unchanged.

System Backup File Contents

A system backup file contains:

- The definition of all topological and application instances used in a system.
- Participant project and executable files with the related mapping information.
- Controller data backup files, page 286.
- Supervision included projects, page 646 added by users.
- User files, page 497 added to Supervision projects.
- OPC Factory Server and OPC UA Server Expert configuration files, page 646.
- The setting of the **System Access Password** property, page 137 and the password (if set).
- Password protection properties, page 509 for topological entities and their value.
- Content containers, page 937 that you have created in the **User Contents** folder structure of the system.

It does not contain:

- Global templates, that is, templates that are used by the instances of the application and topological entities.
- The default Supervision included project.
- Password information, page 658.

NOTE: To be able to restore a backup file, page 147, the items listed above must be present in the database of the system infrastructure where you want to restore it. For example, templates that you have modified or created may not present in the database of a newly installed or different EcoStruxure Process Expert infrastructure.

System Backup File Name

A system backup consists of a file with the default name <system identifier> <(time stamp)>.sbf where *time stamp* is the date and time when the backup file was created, displayed by using the format YYYYMMDDHHMMSS. Time is indicated by using the 24-hour format.

For example, System_1 (20160625144852).

You can edit the file name when creating the backup.

NOTE: For scheduled backups, page 145, the software automatically adds the *time stamp* suffix to the file name that appears in the **Backup Path** text box. The time stamp information corresponds to the point in time when the backup file is created.

Aborting System Back Up Tasks

After clicking **Save** in the **Back Up** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The backup file is not created.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are backing up a small system, the task may complete before you are able to click the icon.

NOTE: Scheduled backups cannot be aborted once started.

Creating a System Backup Manually

To create a system backup manually, proceed as follows.

Step	Action
1	Right-click the a system in the Systems Explorer and select Back Up . Result: The Back up window opens.
2	If you want to create the back up in a different location and/or with a different file name, click the browser button; otherwise, proceed to step 6. Result: A Save As dialog box opens.
3	Select a location where to create the back up file and edit the file name if needed.
4	Enter a description (optional). NOTE: The description appears when you restore the system by using the back up file.
5	Click Save . Result: The Save As dialog box closes and the full path to the file is displayed in the Back up window.
6	Click Save . Result: The software creates a backup file of the system and displays information in the notification panel. NOTE: To cancel the backup task, click the abort icon in the notification panel.
7	If necessary, export templates, page 220 from the root folder of the Global Templates explorer because the templates that are used in the system are not included in the backup file but required in the target infrastructure where the system is restored.

Creating Scheduled System Backups

To create system backups on a schedule, proceed as follows.

Step	Action
1	On the system server computer, open the , select a system, and enable backups, page 145.
2	Configure the schedule and select a backup file location.
3	Click Save and close the scheduler. Result: The software creates a backup file of the system at the scheduled dates and time. Relevant information is displayed in the notification panel of engineering clients that are connected to the system server.

Scheduling System Backups

Overview

The **System Backup Scheduler**, which opens from the system server menu bar (see EcoStruxure™ Process Expert, Installation and Configuration Guide) lets you schedule the automatic backup of systems, page 142.

Each system can have its own backup schedule.

The system server must be running to configure schedules and perform backups.

Backup Notifications

10 min. before the scheduled backup time, a notification about the upcoming backup operation is displayed in the notification panel of engineering clients that are connected to the system server and in the server console.

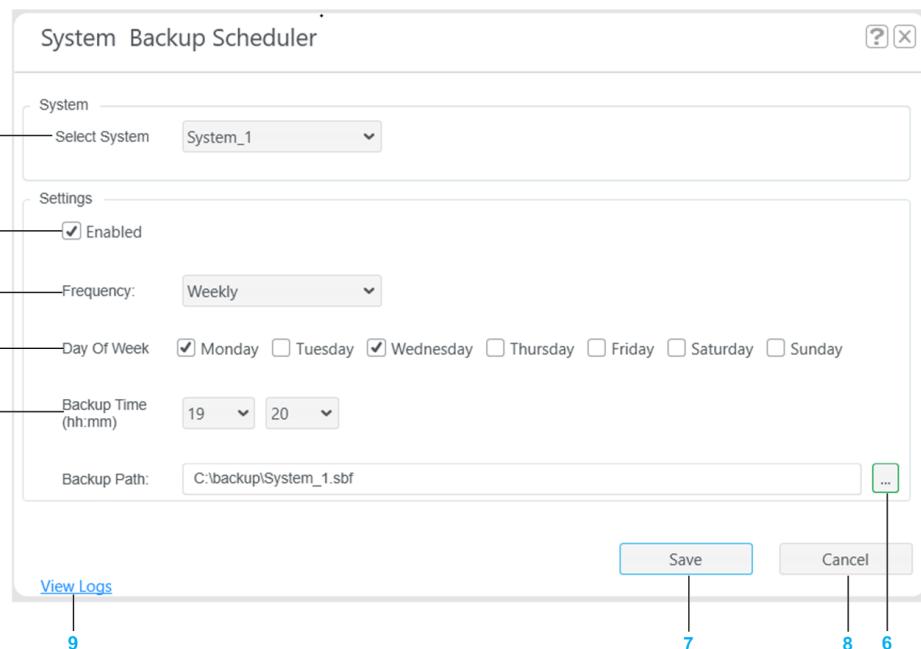
Additional notifications about the successful or unsuccessful backup creation are also displayed.

Managing Backup File Storage Location

Depending on the selected schedule and the size of the systems, backup files can use up a lot of disk space. Manage storage folders regularly to make sure sufficient disk space remains available for new backup files.

System Backup Scheduler Description

The following figure shows an example of the **System Backup Scheduler** window.



Item	Description
1	Lets you select an existing system to configure its backup schedule and file location.
2	Enabled: A backup schedule can be configured and backups are performed accordingly once you save changes. Disabled (check box cleared, default value): No backup is performed. If a schedule has been configured, disabling backups and saving the change discards the schedule. The setting applies to the system that is selected.
3	Backup frequency: Possible values: <ul style="list-style-type: none"> • Daily • Weekly • Monthly
4	• For daily backups: No setting is available. • For weekly backups: You must select at least one day of the week and a time. • For monthly backups: Select a date. For months with less than 31 days, the backup is performed the last day of the month.
5	Time (hour and minutes in 24-hour format) when the backup is performed based on the time setting of the system server computer. NOTE: If the backup cannot be performed, for example, because an editor of the system is open, the software retries two times at 10 min. intervals.
6	Opens the Save As dialog box, which lets you select a folder where backup files are created and a backup file name, page 143. Default value: <System identifier>.dbf in the <i>Documents</i> folder of the logged-in user on the system server computer. If you select a network folder, verify that the system server computer can access the folder at the scheduled times.
7	Lets you save the backup settings for the selected system.
8	Discards unsaved changes.
9	Opens a window that lists the last 20 automatic backup attempts, which did not complete successfully. The list is for all systems and contains the description of the detected issues.

Restoring Systems

Overview

In the **Systems Explorer**, you can restore a system from a system backup file that you had created by using the **Back Up** command or the **System Backup Scheduler**. This requires that templates that are used by the instances of the application and topological entities are present in the database of the EcoStruxure Process Expert infrastructure.

You cannot overwrite an existing system that has the same identifier or merge system data. If a system with the same identifier already exists, the system is restored with a new identifier.

You cannot start a restore task while another system is being restored within the same infrastructure.

Version Compatibility

Backup files are version dependent. You cannot restore system back up files created with another version of the software. The version that was used to create the backup file is displayed when you try to restore a system.

NOTE: Backup files are forward compatible with subsequent releases within the same year (*R*• versions) and service packs of a same version of the software only if the Participant version is unchanged. For example, you can restore a backup file created with version 2018 in version 2018 R2. In such case, a dialog box informs you of the version shift and the software makes the restored systems compatible with the installed version.

Detection of Missing Data Before System Restoration

When you restore a system, if the software detects that an item (for example, a template) is missing in the database, page 143, it opens the **Restore** dialog box. Missing items are reported in the **Status** column of this dialog box.

To be able to restore the system, first, you need to add the missing item to the database where you want to restore it.

If an include project is missing, you can add it to the content repository by following the steps describing how to add a project file (see EcoStruxure™ Process Expert, Global Templates Reference Manual). Start from an existing Supervision genie facet template or create a new one from the **Toolbox** of the **Global Templates Explorer**.

Password Management After Restoring a System

After you restore a system that contains a controller and/or a station node acting as simulator for which password protection, page 509 has been enabled, you need to do the following to be able to deploy a Control Participant project:

- If no password was set before creating the backup file, you need to set one.
- If a password had been set before creating the backup file, you need to enter this password when prompted.

For more information, refer to the topic describing how to manage passwords for Control Participant projects, page 658.

Aborting System Restoration

After clicking **Restore** in the **Restore** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The system is not restored.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are restoring a small system, the task may complete before you are able to click the icon.

Restoring a System

When you restore a system, its name in the database may be different from the name it had when the backup was created. If this occurs and you had already deployed Supervision projects contained in this system when you created the backup file, the system name that was used as reference by runtime navigation services now refers to a different system. As a result, runtime navigation services may not work or display data of another system.

NOTICE

NAVIGATION TO WRONG SYSTEM

Redeploy Supervision Participant projects after restoring a system.

Failure to follow these instructions can display information of an unexpected system.

NOTE: For details on the possible name change, refer to the topic describing the impact of restoration on system names in the database, page 149.

To restore a system from a backup file, proceed as follows.

Step	Action
1	<p>Right-click the system root folder or any system folder in the Systems Explorer and select Restore.</p> <p>Result: The Restore window opens.</p>
2	<p>Browse to the system backup file (.sbf) and click Open.</p> <p>Result: A confirmation dialog box opens.</p>
3	<p>Review the information that is displayed and click Restore.</p> <p>Result: The system is restored. If a system with the same identifier already exists, a dialog box opens, which indicates the new identifier that will be used to restore the system.</p> <p>NOTE: Click Cancel to close the confirmation dialog box without restoring the system.</p>
4	<p>Click OK.</p> <p>Result: The system is restored.</p> <p>NOTE: To cancel the restore operation, click the abort icon in the notification panel.</p>
5	If the system that you have restored contains Supervision Participant projects that you had deployed before creating the backup, redeploy each one of them.

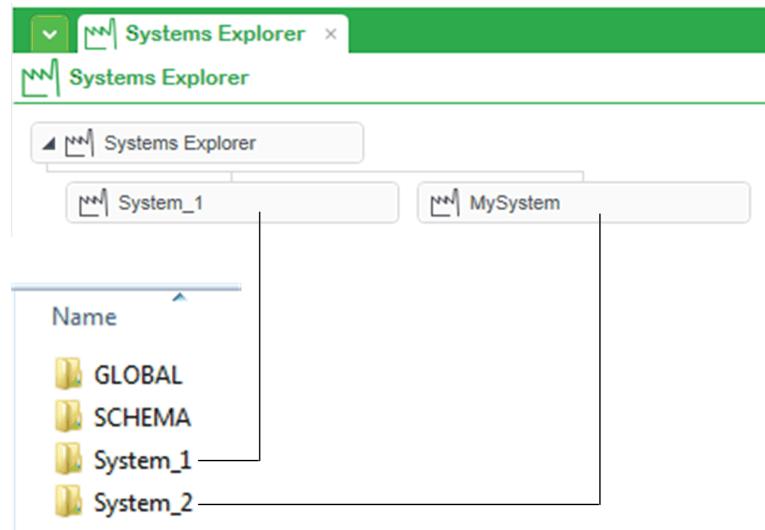
Impact of Restoration on System Names in the Database

Each system that you create has an identifier in the **Systems Explorer** and also a name in the database, which may be different from the system identifier. Names of systems in the database are assigned by the software incrementally and used in the background by software services for system identification purposes, for example, by runtime navigation services provided by the operation client. The name in the database does not change when you rename a system in the **Systems Explorer**.

When you restore a system in the same or another EcoStruxure Process Expert infrastructure, if the name that was originally assigned to it is already used, the system is restored with same identifier in the **Systems Explorer** but with the next available name in the database.

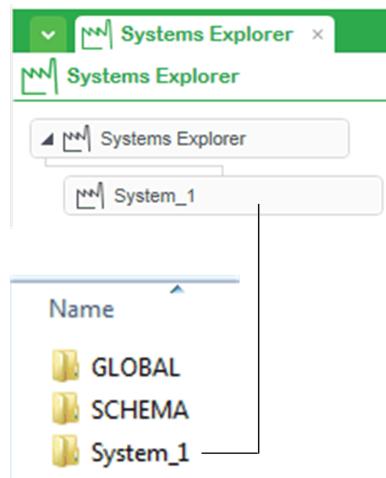
The following example illustrates the scenario where the name of a system in the database changes after restoring it.

The figure shows two systems that exist in the **Systems Explorer** and their respective name in the database folder.

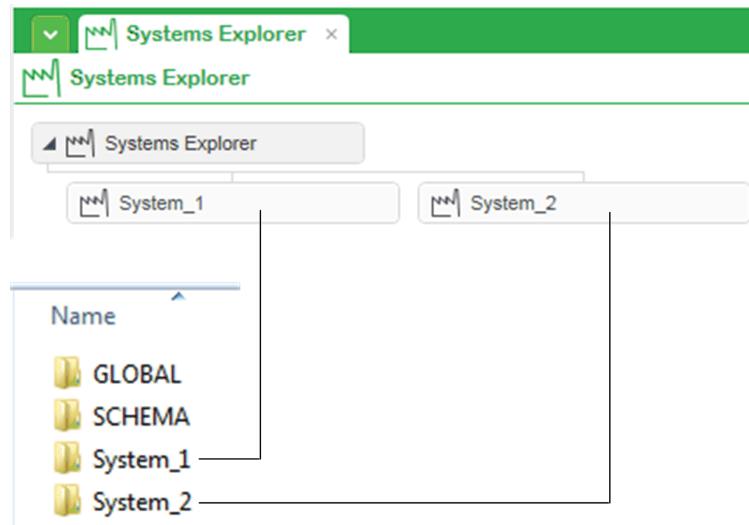


NOTE: The path to the database folder is C:\ProgramData\Schneider Electric\Process Expert x\Db, where x corresponds to the installed major version (the R• version suffix does not appear).

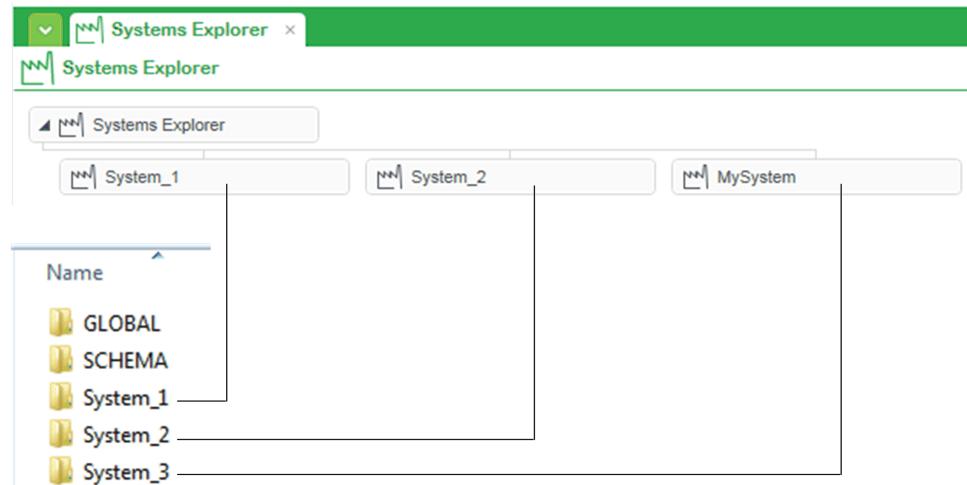
MySystem is backed up and deleted from the **Systems Explorer**. Its name in the database is also removed and becomes available for assignment to a new system.



Then, a new system (*System_2*) is created. The software assigns it the next available name in the database. In this case, it corresponds to the name that was originally assigned to *MySystem*.



Later, *MySystem* is restored. The software assigns it the next available name in the database (*System_3*), which is different from the name it originally had.



As a result, in this example, the name of *MySystem* in the database:

- Was *System_2* before restoration.

- Is *System_3* after restoration.

Instantiation Stage

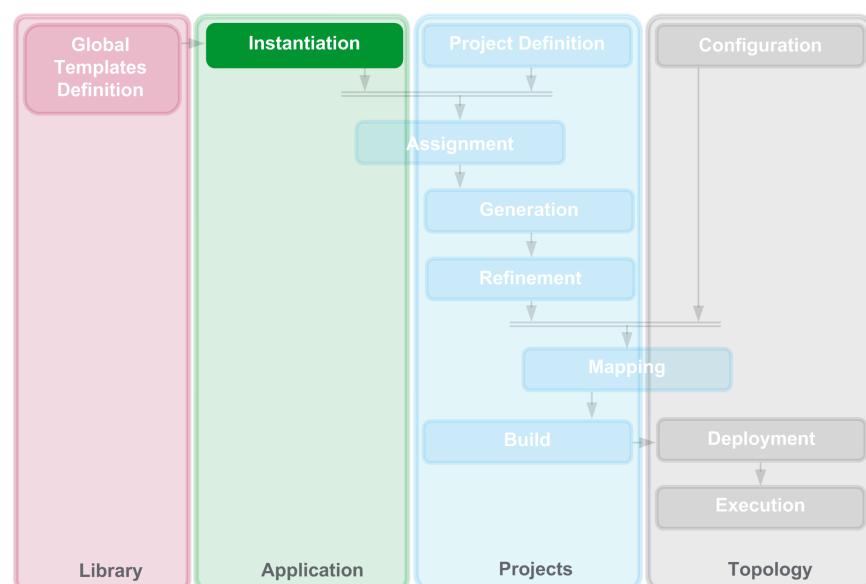
What's in This Chapter

Instantiating Templates	153
Instance Indicators	188
Linking Instances by Using the Asset Workspace Editor	193
Managing Application Folders and Instances	203
Exporting and Importing Application Objects to/from CSV and XML Formats	219

Overview

This chapter provides information to help you model the entire application of the system according to the process and instrumentation diagram (PID).

The following figure shows the position of the **instantiation** stage within the system engineering life cycle.



Refer to the Instantiation stage, page 37 for a description of the purpose of this stage.

Instantiating Templates

Overview

This section describes how to use the **Application Explorer** to:

- Create a folder structure representing the process hierarchy of the application.
- Create and configure instances by instantiating application templates to model the functional components of the application.
- Link instances to each other to propagate data.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Organizing the System Application

Before Starting

Before creating the folder structure for your system application, define the appropriate segmentation, hierarchy, and naming convention for your process.

The **Application Explorer** lets you use templates of the Global Templates library to create instances that model typical components and functions of the application of a system. For example:

- Process objects: Represent the functions of process hardware, for example, motors and valves. This includes signals from the field and/or variables.
For example, the *\$Motor* template can model the functions of the electrical motor of a pump.
- Device objects: Provide functions required to communicate with, control, and/or diagnose a range of electrical devices and instrumentation that are connected to field buses.
For example, the *\$TesysTEM* template can model the functions of a TeSys T variable speed drive communicating through Modbus TCP explicit messaging.
- Communication objects: Provide functions to manage communication with device objects connected to field buses.
For example, the *\$EMPortQ* template can model the management of communication through the port of a Quantum controller through Modbus TCP explicit messaging.

Refer to *Browsing Application Templates*, page 158 for additional information on templates that you can instantiate.

Creating an Equipment Hierarchy

When you create instances and folders in the **Application Explorer**, you create a structure, which is defined by:

- The identifier and location of the application folders.
- The identifier and location of the instances.

By default, this structure is used in the Supervision project to create an equipment hierarchy, page 425, which allows you to view alarms in the Supervision page in a structure identical to the folder structure of the application.

Therefore, changes to the structure, page 206 of the application are reflected on the facets, page 366 of the impacted instances once these facets are assigned to a Supervision project and generated.

Viewing the Application Structure in Control Projects

You can view the folder structure of the application also in relation to Control Participant project sections, page 330 once you have instantiated templates and assigned facets to the Control project.

The functionality is available also when you refine a Control project, page 438 and make changes online, page 728.

Application Explorer

Overview

The work area, page 105 of the **Application Explorer** features:

- The **Templates Browser** pane, which lets you search installed libraries and find the application templates that you need to create the instances of your application.
- The **Application Browser** pane, which lets you view, search, sort, and filter folders and instances of your application. You can also perform other tasks on folders and instances by using this pane.
- The **Asset Workspace** pane, which lets you create asset workspaces. These are structural elements that let you group instances in order to create links between them.
- In the form of tabs:
 - The properties of folders.
 - The **Instance Editor**, which lets you view and edit properties of instances.
 - The **Asset Workspace Editor**, which lets you view, create, and manage links between instances.

Opening the Application Explorer

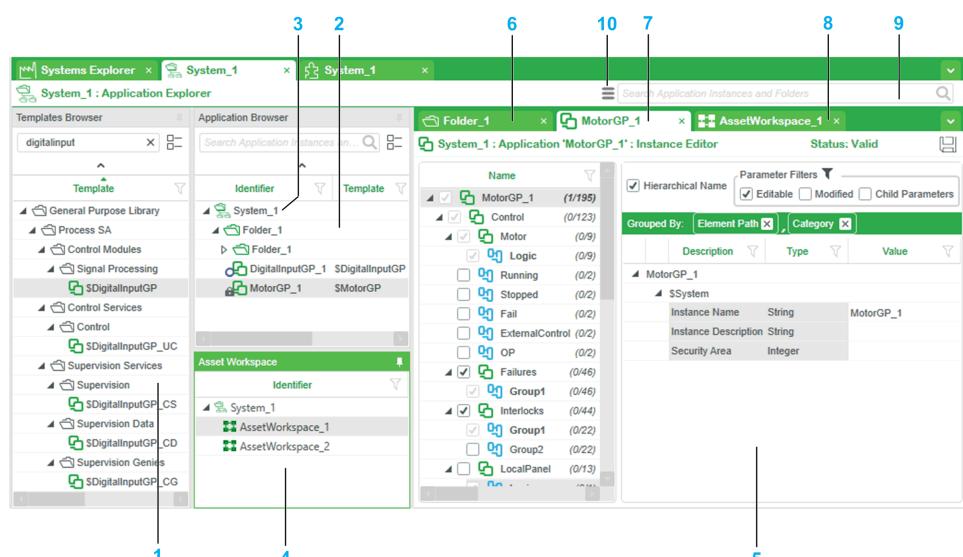
To open the **Application Explorer**, double-click the system in the **Systems Explorer**.

NOTE: You can also open the **Application Explorer** in the following ways:

- Drag the system from the **Systems Explorer** tree view to the tabs bar, page 101 or to a docking tab, page 108.
- Right-click the system in the **Systems Explorer** and select **Open Application**.
- Select the system and click the **Open Application Explorer** button in the toolbar, page 101.

Application Explorer Window

The following figure shows an example of the **Application Explorer** window.



Item	Description
1	Templates Browser pane, page 158
2	Application Browser pane, page 161

Item	Description
3	System root folder
4	Asset Workspace pane, page 186
5	Work area, page 105 of the Application Explorer . It is empty by default.
6	Properties of a folder
7	Instance Editor tab
8	Asset Workspace Editor tab
9	Search field, page 156 for folders and instances
10	Button to open the folder Settings pane, page 170

Searching Instances and Folders by Using the Application Explorer Search Field

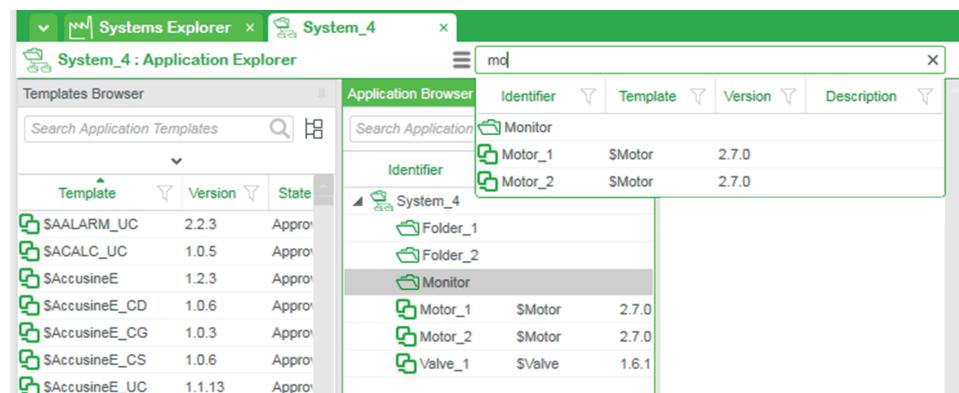
The search field of the **Application Explorer** lets you find instances and folders in the **Application Browser** by typing the entire or part of their identifier. You do not need to use wildcards. The search is not case-sensitive. Results are displayed as you type in a pane with columns. Your last three matching search keywords are stored and appear in a keyword menu. The stored keyword entries are system-independent and persist even after the system server is restarted.

For example, `to` returns instances and folders that contain `to` in their identifier, such as instance `Motor_1`, and folder `Reactor`.

By default, results are displayed in ascending alphabetical order with folders shown first, instances next.

The results pane provides basic information on instances, page 171 and remains open until you close it by clicking the close button. A tooltip provides additional information.

The following figure shows an example of the search results that are displayed when you enter `mo` in the search field.



Right-click a search result in the results pane to open a context menu with the following commands.

Command	Description
Navigate	<ul style="list-style-type: none">For an instance: Expands the containing folder and highlights the instance.For a folder: Expands the folder structure and selects the folder. You can also double-click the item. <p>If the item is open in a window in a work area, the corresponding tab is also selected.</p> <p>NOTE: If the instance is hidden because a filter is applied in the Application Browser, the instance is not shown and a dialog box opens to inform you.</p>
Edit	<ul style="list-style-type: none">For an instance: Opens the Instance Editor tab, which lets you view and edit parameters of the instance, page 176.For folder: Opens the Properties tab, which lets you view, edit, and create parameters of the folder, page 166.

You can sort and filter, page 118 search results.

NOTE: Search results are not updated automatically if instances or folders were created, modified, or deleted in the meantime. You need to start over the search by clicking the search field and pressing **Enter**.

Browsing Application Templates

Overview

The **Templates Browser** pane of the **Application Explorer** lets you find the application templates of the Foundation library and other libraries that are installed in the database of the system server. These templates let you model the application of the system.

The **Templates Browser** pane also displays templates that you have created and/or imported.

To be displayed, a template needs to meet the following criteria:

- Template type: **Application**
- **Usability State** (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of either:
 - **Approved**
 - **Deprecated**

By default, quick filters are applied to display only templates that are:

- Composite templates, page 28.
- The latest version of the template.
- In the approved state.

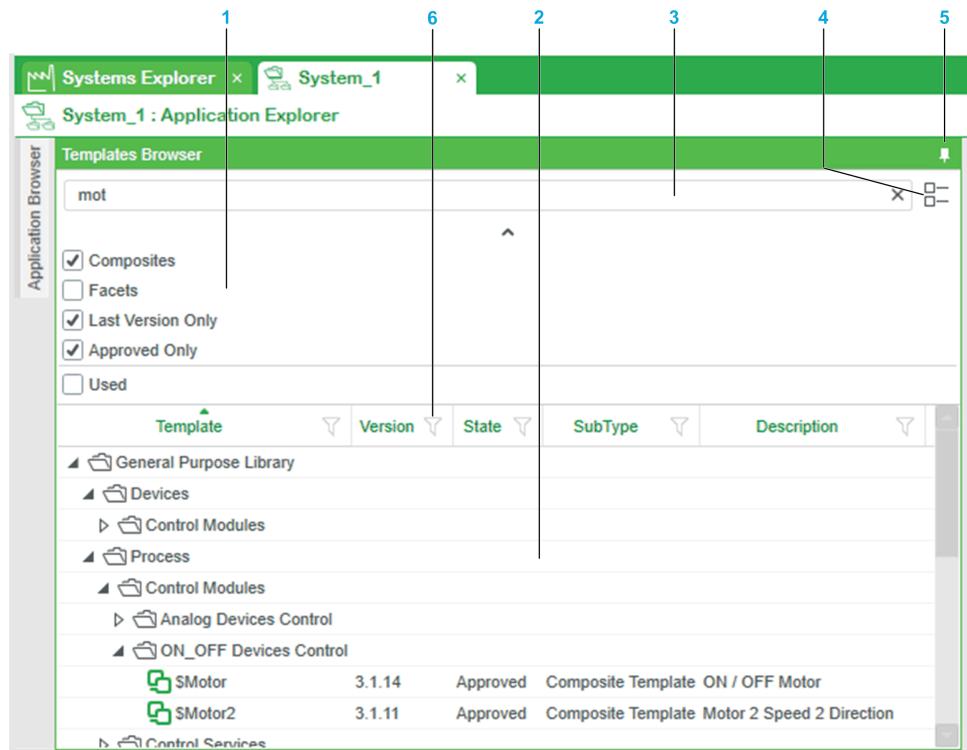
You can display additional templates or refine your search by:

- Using the search field.
- Modifying quick filters.
- Using column filters.

NOTE:

- For details on available Schneider Electric application templates, refer to the help, page 101 of the Schneider Electric libraries for EcoStruxure Process Expert.
- For information on the naming conventions followed by Schneider Electric templates, refer to the topic describing *Global Templates libraries*, page 840.

Templates Browser Pane



Item	Description
1	Quick filters, page 160 (the check mark indicates that the filter is applied).
2	Search results. By default, in tree view, folders are shown first followed by instances. In grid view, instances are shown in a flat list. In both view modes, items are listed in ascending alphabetical order.
3	Search field.
4	Button to toggle between grid and tree view for the display of search results.
5	Pin button, page 111. Lets you minimize the Templates Browser pane to the left edge of the screen. Move the pointer over the Templates Browser tab to display the pane. Click the pin button again to restore the pane.
6	Filters and sorting, page 118.

Searching Templates by Using the Search Field

The search field of the **Templates Browser** pane lets you search templates by typing the entire or part of the template identifier. You do not need to use wildcards or type the \$ prefix. The search is not case-sensitive. Your last three matching search keywords are stored and appear in a keyword menu. The stored keyword entries are system-independent and persist even after the system server is restarted.

The search field acts like a filter for the active view mode as soon as you enter a string. Results are displayed as you type. To show the entire content, clear the field by clicking the close button or deleting your entry.

For example, val returns templates that contain val in their identifier, such as \$HandValve_UC, \$MValve, and \$ControlValve.

The table describes the scope of the search and how results are displayed depending on the view mode that is selected.

View mode	What you can find	What is displayed
Tree view	<ul style="list-style-type: none"> • Templates • Folders 	Templates and folders matching the search criteria with their entire parent folder structure.
Grid view	Templates	Templates matching the search criteria in a flat list.

NOTE: Click the toggle button to switch between display modes. Search keywords and search results are retained.

Searching Templates by Using Quick Filters

The table describes the filters that are available in the **Templates Browser** pane to refine the template search.

Filter	Description	Selected by default
Composites	Displays only templates that are composites, which also include control module templates.	Yes
Facets	Displays only templates that are facets.	No
Last Version Only	Displays only the latest version of templates.	Yes
Approved Only	Displays only templates that have the approved (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) state.	Yes
Used	Displays only the templates that are already used by instances of the application. Using this filter disables the other filters.	No

NOTE:

- When you select multiple filters, only templates that match the selected filters are displayed.
- When you select a filter, the software automatically refreshes the result. The other way around, when the filtered data is modified, the filters are not applied to the modifications.
- Filters are reset when you close the **Application Explorer**.

Additional Filters and Sorting

You can apply additional filters and sort the results, page 118 to refine your search.

Using the Application Browser Pane

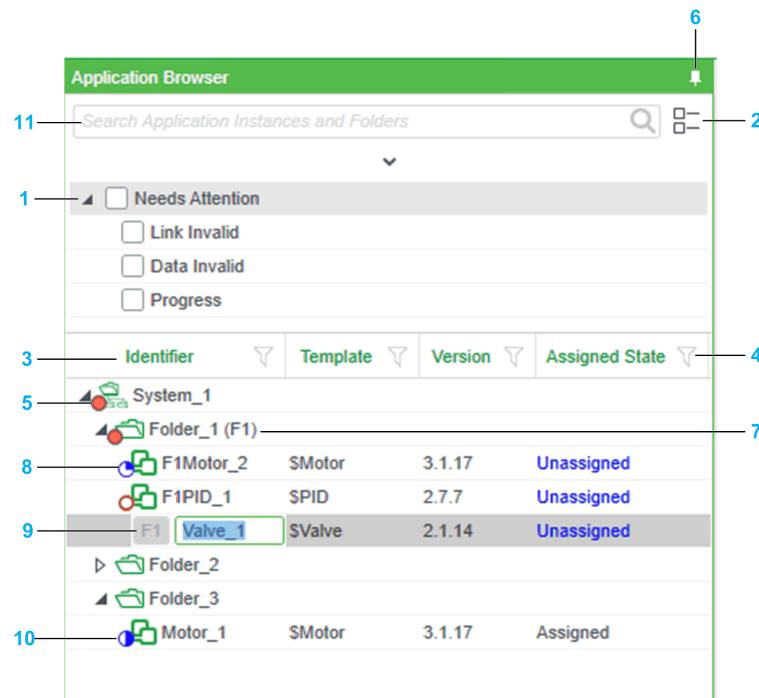
Overview

The **Application Browser** pane lets you:

- Create, delete, move, copy-paste, and rename folders and instances.
- Search, sort, and filter folders and instances
- Toggle between tree and grid view mode, the latter showing only instances.
- Customize the display and the order of columns.
- View and edit the properties of folders and instances.
- Add instances to an asset workspace that is open in the **Asset Workspace Editor** to create interface links.
- View assignments.
- Export and import application objects.

Application Browser Pane

The following figure shows an example of the **Application Browser** pane.



Item	Description
1	Status filters, page 162 (a check mark indicates that the filter is applied).
2	Button to toggle between tree and grid view mode.
3	You can customize, page 163 which columns appear and in which position.
4	Filters and sorting, page 118.
5	System root folder. By default, the system root folder is expanded to display its immediate children. Folders are shown first in the order you have arranged them, followed by instances in ascending alphabetical order.
6	Pin button, page 111. Lets you hide the pane and restore it when it is docked.
7	If the folder has an alias, page 166, it is shown in brackets after the folder identifier.
8	For instances for which the hierarchical naming function is enabled, the alias is shown in front of the instance name, page 171 (if applicable).
9	When you edit the identifier of an instance for which the hierarchical naming function, page 183 is enabled, the alias appears to the left of the identifier in read-only mode (if applicable).

Item	Description
10	Indicator icons, page 188.
11	Search field, page 163.

Using Instance Status Filters

The table describes the impact on the search result when you select instance status filters in the **Application Browser** pane.

Filter	Description
Needs Attention	Shows only instances that satisfy the three subfilter criteria.
Link Invalid	Shows only instances whose Link status, page 197 is invalid.
Data Invalid	Shows only instances whose Data status, page 171 is invalid.
Progress	Shows only instance for which a progress icon, page 189 appears.

NOTE:

- When you select multiple filters, only templates that match the selected filters are displayed. You can combine column filters, page 118 and use the search field to refine your search further.
- When you select a filter, the software automatically refreshes the result. The other way around, when the filtered data is modified, the filters are not applied to the modifications.
- Filters are reset when you close the **Application Explorer**.

Toggling Between Tree and Grid View

Tree view is the default view mode. You can switch to grid view any time by clicking the toggle button. The item selection in either view mode is retained when you switch and revert to this mode.

In grid view, when no sorting and filter is applied, all the instances of the application are shown in a flat list sorted in ascending alphanumerical order.

The table describes the impact on various features when switching from one view mode to the other.

Feature that is used	Impact when switching between view modes
Search	The search criteria is carried over.
Filtering	Is not applied to the other view.
Sorting	

NOTE: In grid view mode:

- Commands of context menus related to folders are not available.
- You cannot create instances by dragging a template to the **Application Browser** pane.

Navigating Inside the Browser Pane

The table describes the various methods that you can use to navigate inside the browser pane.

Key or action	Result
Up/down arrow key	Moves the selection to the previous/next item.
Left/right arrow key ⁽¹⁾	Expands/collapses the next level of the selected node.
Enter⁽¹⁾	
Double-click ⁽¹⁾	On a node, expands/collapses the next level of the node.
Plus (+)/minus (-) key	Expands/collapses the entire hierarchy from/up to the node.

- (1)** The next time you expand the node by using one of these methods, the display reverts to the last expanded view.

Customizing Column Display and Order

You can choose which columns you want to see by right-clicking a column header and selecting **Customize**. The **Identifier** and **Template** columns are always shown.

The **Path** column is available only in grid view.

When you hide a column for which a filter is applied, the filter is cleared.

When the **Customize** dialog box is closed, you can change the order of columns by dragging a column to its new position (except for **Identifier**).

You can configure the column display and order independently in tree and grid view.

For a description of each column, refer to the topic describing the creation of instances, page 171.

NOTE: The order of columns is reset each time you open the **Customize** dialog box and click **Apply**.

Searching Instances and Folders

The search field of the **Application Browser** pane lets you search folders and instances by typing the entire or part of their identifier. You do not need to use wildcards. The search is not case-sensitive. Your last three matching search keywords are stored and appear in a keyword menu. The stored keyword entries are system-independent and persist even after the system server is restarted.

For example, entering `to` returns instances that contain `to` in their identifier, such as folder `Torrefier` and instance `Motor_1`.

The search field acts like a filter for the active view mode as soon as you enter a string. Results are displayed as you type. To show the entire content, clear the field by clicking the close button or deleting your entry.

The table describes the scope of the search and how results are displayed depending on the view mode that is selected.

View mode	What you can find	What is displayed
Tree view	<ul style="list-style-type: none"> • Instances • Folders 	<p>Instances and folders matching the search criteria with their entire parent folder structure.</p> <p>NOTE: When searching for folders, if several folders within the same hierarchical structure match the search criteria, only the highest level parent folder is shown in a collapsed state. You can see whether any child folders match the search criteria by entering the search criteria in the search field of the Application Explorer. A tooltip indicates the path to folders in the search results.</p>
Grid view	Instances	Templates matching the search criteria in a flat list

Search keywords and search results are retained when you switch between view modes.

By default, results are displayed with folders shown in the order you have arranged them followed by instances in ascending alphanumerical order.

If sorting and/or filters, page 118 are applied, they are also applied to the search results.

NOTE: Search results are not updated automatically if instances or folders are created, modified, or deleted. For example, if you have entered `va` in the search field and create a folder, which has `Folder_n` as default identifier, it will be shown.

Creating Application Folders

Overview

You can create application folders to structure your application by using the **Application Browser**.

The software lets you structure the application according to the ISA-S88.01-1995 standard for batch control. You can create and rename the application folders according to the physical assets that you want to represent and assign icons to the folders, page 170 to define a hierarchy of equipment.

Folders that you create may inherit properties of their parent folder, such as the **Area**.

You can also:

- Create folders also by copying and pasting or import, page 219.
- Move folders, page 166 with or without changing the folder hierarchy.
- Assign icons to application folders, page 170.

Creating Application Folders

To create an application folder, proceed as follows.

Step	Action
1	<p>Right-click the system root folder and select Create Folder.</p> <p>Result: An application folder is created at the end of existing folders of the same level. The identifier is in edit mode.</p>

Application Folder Actions

Right-click the root folder, any application folder, or multiple folders to open a context menu with the following commands.

When you select...	The software...
Create Folder	creates an application folder or subfolder. The command is not available when more than one folder is selected.
Open Topology	opens the Topological Explorer , page 261, which allows you to start the configuration stage or access the topology of the system. The command is available from the root folder only.
Open Project	Opens the Projects Explorer , which allows you to start the Project Definition stage, page 254 or access the Control and Supervision projects of an existing system. The command is available from the root folder only.
Update Template , page 212	opens the Update Instance Template window. It lets you update the template that is used by instances inside the folder and any subfolder with the latest version of that template, which is available in the Global Templates library on the system server.
Replace Template , page 212	opens the Replace Instance Template window. It lets you select another template to be used by instances contained in the folder and any subfolders. The template needs to be available in the Global Templates library on the system server.
Copy , page 208	copies folders, their subfolders, and the instances they contain. The command is not available from the root folder.
Paste , page 208	pastes the copied items to the selected locations in the system or in another system of the server. The command is available if the copied content is valid for the selected target locations.
Export , page 219	opens an Export dialog box, which allows you to save the parameters and configuration data of application objects to a file in CSV or XML format.

When you select...	The software...
Import, page 246	opens an Open dialog box, which allows you to select an export file in CSV or XML format, display its compatible contents, and select instances to be imported into the application. The command is not available when more than one folder is selected.
Delete	deletes any selected folders and their contents including any subfolders and instances. The command is not available from the root folder. NOTE: After confirming the command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. Already deleted instances and/or folders are restored. Only the user who has selected the command is allowed to abort it. If you are deleting only a few items, the task may complete before you are able to click the icon.
Rename	lets you enter a new folder identifier. NOTE: <ul style="list-style-type: none"> Renaming a folder may affect the equipment hierarchy, page 154. The command is not available from the root folder and when more than one folder is selected. If the folder has an alias, it is not shown while the identifier is in edit mode.
Properties	opens the Properties, page 166 tab, which allows you to edit the properties of the application folder. The command is not available from the root folder and when more than one folder is selected.

Moving Folders

You can move one or more folders by dragging them to a new position. A tooltip indicates the result of the action, which can have an impact on instances if the folder hierarchy is changed.

The table describes the tooltips that can appear and the corresponding result.

Tooltip	Result
Move X before Y	The folder and its content is repositioned before or after the other folder.
Move X before Y	The folder hierarchy is not changed.
X	The folder that you are moving and its content becomes a child of the other folder. The equipment hierarchy, page 425 may be impacted. The instance identifier, page 183 and assignment status, page 411 of generated facets may be impacted.
Where: X Is the identifier of the folder that you are moving. Y Is the identifier of another folder of the application.	

NOTE: When you move several folders, you cannot reposition them and maintain the same hierarchy.

Application Folder Properties

To open the properties of the application folder, right-click the folder and select **Properties**. A tab opens inside the work area, page 105 of the **Application Explorer**.

You can also open the properties by dragging the folder to the work area, the tabs bar of a work area, or to a docking tab, page 109 within the **Application Explorer**.

You can edit the following items.

Tab	Item	Description						
General	Identifier	<p>Designation of the folder.</p> <p>The identifier must be unique for each subfolder of the same parent folder and satisfy naming rules, page 88.</p> <p>Default value: Folder_n where n is an incremental number starting at 1 for each folder level.</p> <p>NOTE: Changing the identifier of an application folder may affect the equipment hierarchy, page 154.</p>						
	Alias	<p>Optional.</p> <p>You can assign an abbreviated name to the folder.</p> <p>The alias is shown in brackets next to the folder identifier.</p> <p>The software uses the alias for the hierarchical naming function, page 183.</p> <p>You cannot create or modify the alias value if it causes any instance identifier to be non-unique.</p> <p>Default value: Blank</p>						
	Description	<p>Optional.</p> <p>You can enter a description of the folder with free form text. The description you enter appears in a tooltip when you move the cursor over the folder.</p> <p>Default value: Blank</p>						
	Area	<p>Optional.</p> <p>You can assign a folder to an area.</p> <p>Range: 1...255 (integer)</p> <p>Default value: Blank</p> <p>The software attempts to propagate the area value to subfolders and any instances contained in the hierarchy to which no area value has been assigned manually.</p> <p>For details, refer to the topic describing area value propagation, page 167.</p> <p>NOTE: You can also enter a string instead of an integer value. In such case, you must refine the Supervision project after generating it and use the same string as label to name the area; otherwise, the Supervision project does not build successfully. Refer to the help of the Supervision Participant for details.</p> <p>NOTE: The area value is also propagated to Supervision elements of instances such as genies.</p>						
Manage Parameters	Parameter_1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; padding: 2px;">Identifier</td> <td style="padding: 2px;">You can use as many folder parameter sets as you require.</td> </tr> <tr> <td style="padding: 2px;">...</td> <td style="padding: 2px;">Default value: No parameter</td> </tr> <tr> <td style="padding: 2px;">Parameter_n</td> <td style="padding: 2px;">Value</td> </tr> </table>	Identifier	You can use as many folder parameter sets as you require.	...	Default value: No parameter	Parameter_n	Value
	Identifier	You can use as many folder parameter sets as you require.						
...	Default value: No parameter							
Parameter_n	Value							
	NOTE: Click Add Parameter to add a parameter.							

Area Parameter Value Propagation

The software automatically propagates the value (including blank (NULL value)) of the **Area** parameter of a folder to:

- Existing subfolders and instances that the folder and its subfolders contain unless you have assigned an area value to them manually.
- Subfolders and instances that you create afterwards.
- Subfolders and instances that you add afterwards either by copying or moving them unless an area value had been assigned to them either automatically or manually.

If you change an existing area value of a folder, only the area value that was propagated automatically to the next level of subfolders and their instances is updated. The change is propagated down the folder structure, level after level, until a manual folder area entry is detected.

If the software is not able to propagate the value of the **Area** parameter to any item because the item is locked, it:

- Displays a notification to inform you which item is locked.
- Rolls back any propagated area value.
- Reverts the **Area** parameter value that you have entered in the folder properties to its previous value.

NOTICE

NO CHANGE PROPAGATION

When the **Area** parameter value of a folder is not propagated:

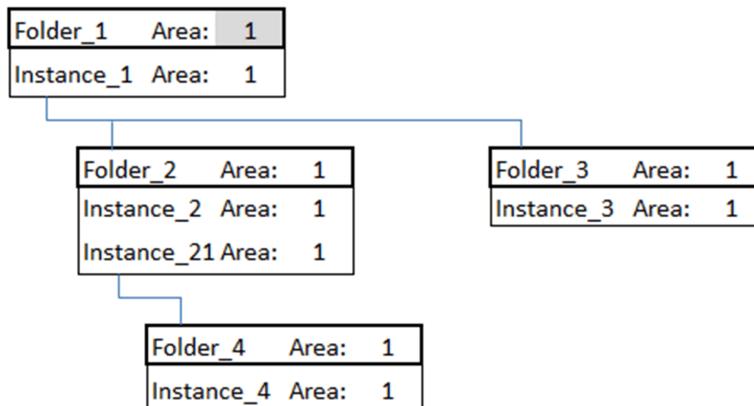
- Reenter the **Area** parameter value in the properties of the folder.
- Verify in the **Notification Panel** that parameter value propagation completed successfully.

Failure to follow these instructions can result in alarms being signaled in the wrong area during operation.

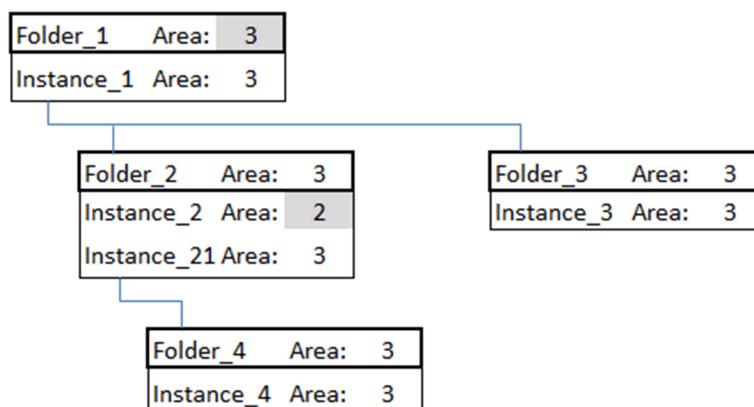
Examples

The following three examples illustrate the area value propagation to existing subfolders and instances in the **Application Explorer**. The changes described in these examples are performed in steps inside the same folder structure.

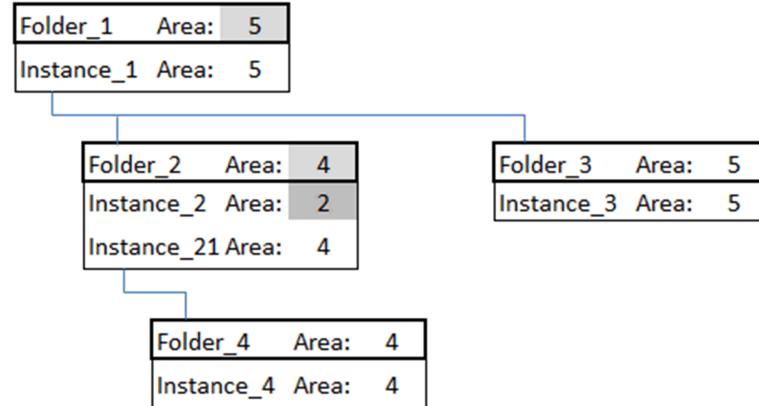
In a first step, starting from a folder structure and instances without area value, if you change *Area* of *Folder_1* to 1, the value is propagated to *Instance_1*, subfolders, and their instances.



In a second step, if you change *Area* of *Instance_2* to 2 and then *Area* of *Folder_1* to 3, value 3 is propagated to *Instance_1*, subfolders, and their instances except for *Instance_2* because the area was entered manually.



In a third step, if you change *Area* of *Folder_2* to 4, value 4 is propagated to *Instance_21*, the *Folder_4* subfolder and its instance. *Instance_2* remains unchanged because the area was entered manually. If you then change *Area* of *Folder_1* to 5, value 5 is propagated only to *Instance_1*, subfolder *Folder_3*, and its instance because the area of other subfolders and instances have either been entered manually (for example, *Folder_2*) or are manual entries that have been propagated (for example, *Folder_4*).



Assigning Icons to Application Folders

Overview

By using the **Settings** pane, you can assign icons to each level of application folders. The icons appear in the tree view of the **Application Browser** pane, page 155.

You can use this feature, for example, to represent graphically the physical hierarchy of equipment used in a batch process according to the ISA-S88.01-1995 standard.

The settings apply to the system for which you configure them, to any user who opens the system in an engineering client, and persist even if you restart the system server.

NOTE: You cannot assign an icon to the system root folder.

Available Icons

You can choose from the following selection of icons.

Icon	Description
	Default folder icon
	Site
	Area
	Process cell
	Unit
	Equipment module

NOTE: Control modules are represented by the instances of the application.

Assigning Icons to Folder Levels

To assign an icon to a folder level, proceed as follows.

Step	Action
1	Click the Explorer Layout Settings button, page 101. Result: The Settings pane opens.
2	In the Appearance section, select the folder level to which you want to assign an icon.
3	In the Style menu, click the icon that you want to use. Result: The icon is assigned to each existing folder of the selected level and to each new folder that you create in the selected level.
4	Repeat the procedure for other folder levels.

Creating Instances

Overview

Instances that you create in the folder structure, page 165 of the application are created with the default configuration of their template. Basic information on each instance is available at the folder level.

Once you have created an instance by using the **Application Browser**, you can perform various actions by using the pointer, accessing the context menu of the instance, or using keyboard shortcuts.

Order of Instantiation

The order in which you create instances inside an application folder may affect the order of the referenced Control facets in a section during assignment, page 362.

Instantiating Application Templates

To instantiate a template in the root folder or in any application folder, proceed as follows.

Step	Action
1	Verify that the folder in which you want to create the instance is displayed in the Application Browser .
2	In the Templates Browser , select a template.
3	Drag the template to the folder. Result: The folder opens or expands and an instance of the template is created in last position with the default identifier in edit mode. NOTE: If the folder has an alias, it is shown in a separate cell on the left of the identifier.
4	To change the identifier, type it; otherwise, press Enter to accept the default name. NOTE: To keep the default name and create a new instance, you do not need to press Enter ; you can drag another template while the identifier of the instance is still in edit mode.

NOTE: Refer to *Managing Application Folders and Instances*, page 203 for information on features that are available when working with many application folders and instances.

Instance Description

The following information of an instance is displayed in the **Application Browser** when the corresponding column is shown, page 163.

Header	Description
Icon	Displays the icon corresponding to the type of template that the instance uses.
Identifier	For instances, displays the instance identifier (see EcoStruxure™ Process Expert, Global Templates, Reference Manual), which must be unique for each instance of the same template. The software calculates the identifier by using the parameter \$Name of the instance and the applicable naming conventions. For folders, displays the folder identifier, which you can edit in the Properties tab or by using the Rename command. NOTE: <ul style="list-style-type: none"> You can edit the \$Name parameter of an instance in the Instance Editor, page 176 or by using the Rename command. If the resulting identifier is not unique, the \$Name field is outlined in red. In such case, clicking outside of the field reverts the name to its previous value. The hierarchical naming function, page 183 can modify the identifier of an instance. Two instances using two different templates cannot have the same identifier if both instances have an element in their composition in common.

Header	Description
	(for example, the <i>Logic</i> element, which is created by the <i>\$DEVCTL_UL</i> facet). In such case, the identifier of the element that both instances have in common would be the same; this is not allowed.
Template	Indicates the identifier of the template that the instance uses. The field is not used for folders.
Version	Indicates the version of the template that the instance uses. The field is not used for folders.
Data	Indicates the status of the instance. Possible values: <ul style="list-style-type: none">• Valid• Invalid The value is updated when you save changes in the Instance Editor . The field is not used for folders. NOTE: You can obtain information about the invalid status, page 180 of an instance in the Validity Diagnosis window: <ul style="list-style-type: none">• By clicking Invalid.• When viewing its properties, page 176.
Link	Indicates the status of the application interface link of the instance. Possible values: <ul style="list-style-type: none">• Valid• Invalid The value is updated when you make a link by using the Asset Workspace Editor, page 193 or save changes in the Instance Editor that impact the link requirements of the instance. The field is not used for folders. NOTE: You can obtain information about the invalid status, page 197 of an instance in the Validity Diagnosis window: <ul style="list-style-type: none">• By clicking Invalid.• In the Asset Workspace Editor.
Assigned State	Indicates the assignment status, page 366 of the facets of the instance. Values: Assigned: At least 1 facet of the instance is assigned to any project of the system. Unassigned: No facet of the instance is assigned to any project of the system. The field is not used for folders.
Description	Displays the content of the <i>\$Description</i> parameter the folder or the instance. You can edit the field in the Properties or the Instance Editor , page 176 tab respectively.
Area	Displays the value of the <i>\$Area</i> parameter for the folder or the instance. You can edit the field in the Properties tab or the Instance Editor , page 176 tab respectively. NOTE: If the value was assigned automatically, page 167 by the software, it appears between angle brackets. For example, <5>.
Path	Indicates the path to the instance in the application starting from the system root folder. NOTE: The column is shown only in grid view mode.

Instance Actions

Right-click an instance to open the instance context menu with the following commands.

Command	Description
Properties⁽¹⁾	Opens the Instance Editor , page 176 in the work area, which allows you to view and configure parameters, and select services of the instance.
Navigate to Workspace⁽¹⁾	Opens the Show Workspace window, which lists the asset workspaces that the instance belongs to. Double-clicking an asset workspace opens it in the Asset Workspace Editor and highlights the instance. This lets you create and manage links between instances.
Edit Links⁽¹⁾	Opens the Links Editor , page 943, which allows you to view the interface links that exist for the instance.
View Assignments, page 379⁽¹⁾	Opens the View Assignments window, which lets you verify if the facets that are referenced by the instance are assigned to Participant projects: <ul style="list-style-type: none"> If a facet is not assigned, you can select an existing project and open the Assignment Editor to assign it. If a facet is assigned, you can access the container to which the facet is assigned. For Supervision genie facets, you access the Pages node in either case. NOTE: The command is not available if the Data or Link status of the instance is Invalid .
Update Template, page 212	Opens the Update Instance Template window. It lets you update the template that is used by the instance with the latest version of that template, which is available in the Global Templates library on the system server.
Replace Template, page 212	Opens the Replace Instance Template window. It lets you replace the template that the instance is using by another template, which is available in the Global Templates library on the system server.
Inspect Instance⁽¹⁾, page 204	Opens the Inspect Instance window, which provides information on the entire instance configuration, including interface links.
Copy, page 207	Copies the selected instances for pasting within the application of the system or any other system on the server.
Export, page 220	Opens an Export dialog box, which allows you to save the instance data to a file in CSV or XML format.
Rename⁽¹⁾	Sets the identifier of a single instance to edit mode. Type the new identifier and press Enter . NOTE: The instance identifier is split in two and only the instance name (\$Name) part is editable if the following conditions are satisfied: <ul style="list-style-type: none"> You have defined an alias for the folder containing the instance or any of its parent folders. The hierarchical naming function, page 183 for the instance is enabled. The concatenated, non-editable aliases are shown on the left of the instance identifier.
Delete	Deletes the instance from the application. You can select multiple instances within the same folder. Not available while you edit the instance. If you have assigned facets of the instance to projects, the assignment state, page 366 of these facets changes to <i>Deleted</i> . NOTE: After confirming the command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. Already deleted instances are restored. Only the user who has selected the command is allowed to abort it. If you are deleting only a few instances, the task may complete before you are able to click the icon.
(1) The command is not available when multiple instances are selected.	

Counting Instances

Overview

The software includes a mechanism, which counts the instances that exist in the application of any system on the system server and compares their number to the maximum number of instances that the application size license that is activated allows.

At any time during the system engineering life cycle, you can view the instance count for each system, the total count, and the maximum number of instances that is allowed.

The software does not allow actions that cause the maximum number of instances of the application to be exceeded.

Viewing Instance Count Information

Instance count and software license information is displayed in the **About EcoStruxure Process Expert** dialog box, page 104.

The table describes the information that is displayed in the **License Details** section of the **General** tab.

Label	Description
Maximum Instances Allowed	Total number of instances that you can create with the current license.
Current Instance Count	Total number of instances in the systems that exist on the system server.

The table describes the information that is displayed in the **Systems Details** tab.

Header	Description
System Name	Identifiers of the systems that exist on the system server.
Instance Count	Total number of instances of the system.

Actions Affecting Instance Count

The following actions that you can perform in the **Application Explorer** increase and/or reduce the instance count in the application:

- Instantiate
- Paste
- Import
- Delete

The following actions that you can perform in the **Systems Explorer** increase and/or reduce the instance count in the application:

- Restore
- Delete

Exceeding the Allowed Number of Instances

If one of your actions causes the maximum number of instances that is allowed by the activated license to be exceeded, the software cancels the entire action. It also displays a message to advise you that you have reached the limit of instances allowed by the license.

In such case you have the choice to:

- Delete instances (you can export the instances, page 220 before deleting them to reuse them later.)

- Delete systems (you can back up the system, page 142 before deleting it to reuse it later.)
- Upgrade the application size license (see EcoStruxure™ Process Expert, Licensing Guide).

Accessing Instance Properties

Overview

The **Instance Editor** tab opens in the inside the work area, page 105 of the **Application Explorer** and provides access to:

- Control, Supervision, and documentation services, page 28, which are provided by the template that the instance is using.
- Parameter values of the elements, page 26 of the instance.
- The number of parameters that have been modified.
- The status of the instance.

NOTE: More information on the instance configuration is available in the **Inspect Instance** window, page 204 in read-only mode.

Read-Only Mode

You can open the **Instance Editor** for an instance in read-only mode if either of the following is true:

- The instance is being edited, for example, from another client connected to the same system server.
- The instance is displayed in the **Links Editor**, either as active or inactive., page 943

The read-only mode allows you to view instance parameters but not to modify them.

Opening the Instance Editor

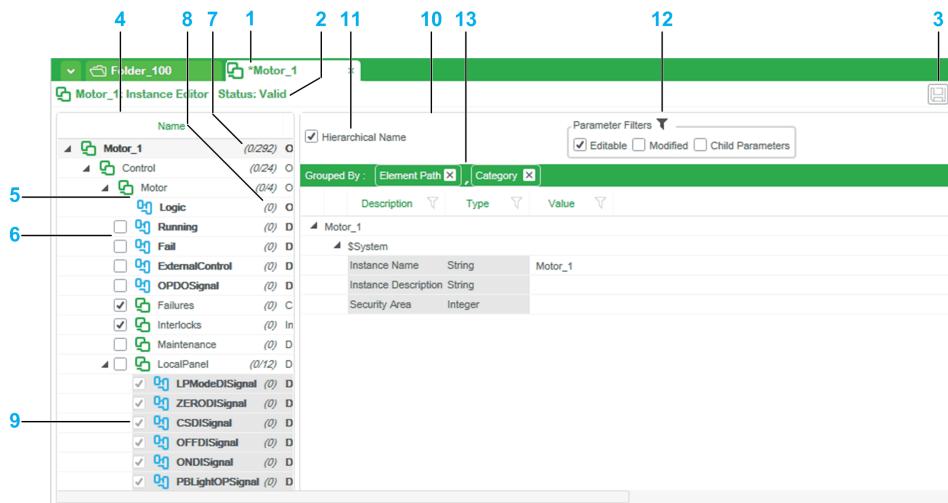
To open the **Instance Editor**, double-click the instance that you want to configure in the **Application Explorer**.

You can also open the **Instance Editor** in the following ways:

- Right-click the instance and select **Properties**.
- Drag the instance to either area within the **Application Explorer**:
 - The work area, page 105.
 - The tabs bar of a work area.
 - A docking tab, page 109.

Instance Editor

The following figure shows an example of instance *Motor_1*, which uses template *\$Motor* open in the **Instance Editor**.



Item	Description
1	Instance identifier. It is prefixed with * if you have made a change and not saved it.
2	Instance validity status indicator. The status is set to Invalid and details are available by clicking the information button if either the Data or Link status, page 171 is invalid.
3	Save button. Click to save parameter values and propagate changes, page 366.]
4	Instance pane, which allows you to view the services that the instance provides and to enable/disable optional ones. By default, nodes are shown expanded.
5	Element displayed in bold indicating that the element contains editable parameters.
6	Check box for element selection.
7	Counter indicating the number of modified parameters for a node out of the total number of parameters that exist for that node.
8	Counter indicating the number of modified parameters for an element.
9	Elements of optional nodes of the instance that are not selected and that have configurable parameters are displayed on a light gray background. This allows identifying easily which are the optional services of the instance that provide additional customization capabilities. Selecting the node removes the gray background.
10	Parameters pane, which allows you to configure the parameters of the elements of the instance that you have selected.
11	Optional Hierarchical naming function, page 183 selection box. The function is enabled by default.
12	Parameter Filters section, page 180.
13	Grouped by sorting function for parameters.

The table describes the fields of the instance pane.

Header	Description
Name	<p>Shows the composition of the instance by displaying the identifier of the instance and:</p> <ul style="list-style-type: none"> • Its core and optional composite elements (nodes) • Core and optional elements thereof (facets) <p>Elements (facets) provide the services of the application template, which depend on the template that the instance uses.</p> <p>For details on the services that are provided by Schneider Electric application templates, refer to the help, page 101 of the Schneider Electric libraries for EcoStruxure Process Expert.</p> <p>An item in bold indicates that you can configure parameters of the element.</p> <p>Move with pointer over an element to see a description of the element in a tooltip.</p>
Icon	Indicates the type of template that is used by the instance and its elements.
Check box	<p>Indicates that the element is optional and not mandatory for the control module to function. Typically, optional elements correspond to additional functionalities that are built into the template and that you can select to customize the instance according to your needs.</p> <p>An optional element may contain optional subelements, which you can select individually.</p> <p>Selecting an element enables it and creates the corresponding facets. If applicable, it allows you to configure parameters of the service that the element provides.</p> <p>Selecting an optional Control element automatically selects the corresponding Supervision element if it is available. For example, when you select the Interlocks element of the \$Motor template, the InterlockTags Supervision element is selected.</p> <p>The other way around, clearing a selected Control element automatically clears the corresponding Supervision element.</p> <p>The following Supervision elements are selected by default:</p> <ul style="list-style-type: none"> • Elements corresponding to core Control elements, which are not optional. • Elements corresponding to optional Control elements, which are selected by default.
(x / y)	<p>Node counter (7). For the instance and for its composite elements, the first number indicates the number of parameters that you have changed in the node (x) compared to the default value and the second, the total number of parameters (y) that you can change for this node.</p> <p>Changes to the Instance Name are not counted.</p> <p>Counters are updated as soon as you confirm your entry by pressing Enter. For boolean types, when you select or clear them.</p> <p>NOTE: The counter turns bold to indicate that you have changed a parameter of the element itself. When you change a parameter of any of its child elements, the counter font style remains regular.</p>
(z)	<p>Element counter (8). For child elements, indicates the number of parameters that you have changed compared to the default values.</p> <p>The counter appears even if the element has no editable parameters.</p> <p>Counters are updated as soon as you confirm your entry by pressing Enter.</p>
Description	Description of the instance or element displayed in the Name column.
Template	<p>For instances and composite elements, indicates the identifier of the template that the instance or composite element is using.</p> <p>For facet elements, indicates the identifier of the facet template that encapsulates the constituent provided by the Participant to implement the service.</p>
Version	Version of the template that the instance or element uses.

The table describes the fields of the parameters pane.

Header	Description
Group By	<p>Allows you to sort parameters that are displayed in the parameters pane by the following criteria:</p> <ul style="list-style-type: none"> • Element Path • Category • Type <p>By default, parameters are grouped as follows:</p> <ul style="list-style-type: none"> • For instances based on a composite template: By Element Path, then Category. • For instances based on a facet template: By Category. <p>Click the close button of a criteria to remove it from the grouping. The corresponding column is displayed.</p> <p>Drag a column header to the green bar to group parameters by the corresponding criteria. You can group by multiple criteria simultaneously.</p> <p>When multiple criteria are selected, swap their order by dragging a criteria left or right.</p> <p>When you have selected to group by a criteria, the corresponding column is not displayed anymore.</p>
Element Path	<p>Path to the element, which provides the parameters, as shown in the instance pane.</p> <p>Not available for instances based on a facet template.</p>
Category	Category of the parameter.
Name	<p>Identifier of the parameter.</p> <p>The column is hidden by default.</p> <p>Right-click any column header and select Show Name to display the column. Right-click any column header and select Hide Name to hide the column.</p>
Description	Description of the parameter.
Type	Indicates the data type of the parameter.
Value	<p>Default value of the parameter or the value that you have entered.</p> <p>NOTE: For parameters of the enumerated data type, for each enumerator, both the name and the associated integer are shown (for example, <i>Operator</i> (2))</p>
Resolved Value	<p>Displays the value after formatting according to the data type.</p> <p>For example, when the check box of a boolean parameter is selected, the value that is displayed is True.</p> <p>The column is hidden by default.</p> <p>Right-click any column header and select Show Resolved Value to display the column. Right-click any column header and select Hide Resolved Value to hide the column.</p> <p>NOTE: Open the Inspect Instance window, page 204 to view the resolved value of additional template parameters, which has been calculated by using the value property and any applicable logic defined in the template definition.</p>

Navigating Inside the Instance Pane

The table describes the various methods that you can use to navigate inside the instance pane.

Key or action	Result
Up/down arrow key	Moves the selection to the previous/next item.
Left/right arrow key ⁽¹⁾	Expands/collapses the next level of the selected node.
Enter ⁽¹⁾	On a node, expands/collapses the next level of the node.
Double-click ⁽¹⁾	On a node, expands/collapses the entire hierarchy from/up to the node.
Plus (+)/minus (-) key	Expands/collapses the entire hierarchy from/up to the node.

- (1) The next time you expand the node by using one of these methods, the display reverts to the last expanded view.

Displaying Instance Parameters Using Filters

The table describes the filters that are available in the **Parameter Filters** section to customize the display of parameters of the selected instance or element.

Filter	Description
Editable	Displays only parameters of the selected instance or element that you can edit. A parameter may not be editable because it is optional and you have not selected it. The filter is applied by default.
Modified	Displays only parameters of the selected instance or element whose value you have modified compared to their default value.
Show Child Parameters	Displays parameters of the selected instance or element including those of child elements.
None selected	No filter is applied. Displays the parameters of the selected instance or element.

NOTE:

- When you select multiple filters, the software displays only parameters that match the selected filters.
- When you select or clear a filter, the software automatically refreshes the result.

To apply default filters, click the filter icon, then click **Default**.

To clear any filters, click the filter icon, then click **Clear Filter**.

Instance Data Status

When any of the following conditions apply, the **Data** status of the instance in the **Instance Editor** changes to **Invalid**:

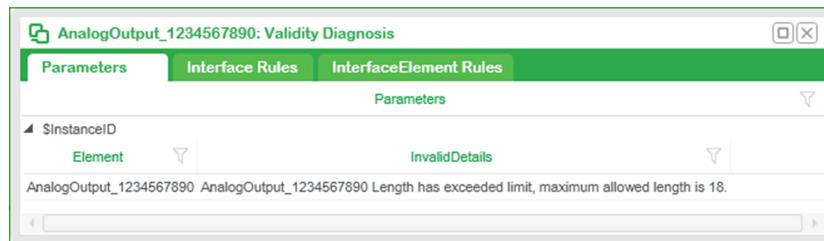
- Entering a character that is not allowed, page 88 in the **\$Name** field.
- Entering a number of characters in the **\$Name** field causing the instance identifier to exceed the length limit that may be defined, page 88.
- Selecting an invalid combination of optional elements in the instance pane of the **Instance Editor**.
- Creating an instance that references an element that has the same identifier as an existing element in the application. For example, if you create two instances with the same identifier from two different templates, which have an element in common, such as **Interlocks**, which uses the **\$CONDSUM1_UC** facet template.

You cannot assign to projects instances that have a **Data** status that is **Invalid**.

Furthermore, when facets of an instance are assigned to a Participant project, and later either status of the instance changes to **Invalid**, the software propagates the invalid status to facets of Participant projects.

While the **Data** status is displayed as **Invalid**, click the validity status information button  to display the **Validity Diagnosis** window, which provides information about the cause of the status.

The following figure shows an example of the **Parameter** tab of the **Validity Diagnosis** window that the software displays when it detects an invalid identifier.



NOTE: For information on the other tabs that are related to the **Link** status of the instance, refer to the topic describing instance links, page 197.

NOTE: If the status of an instance is **Invalid** because of a duplicate element identifier, deleting or modifying the other instance that contains the same element does not change the status back to **Valid**. After you have solved the conflict, edit the invalid instance and save any change (for example, a modification of the description) to set its status to **Valid**.

Configuring Instances

Overview

The **Instance Editor** lets you:

- Enable or disable optional Control, Supervision, and documentation services, which are provided by the template that the instance is using.
- View and configure parameters of the instance.

For more information to help you select and configure instances to meet the requirements of your application, refer to the help, page 101 of the Schneider Electric libraries for EcoStruxure Process Expert.

For information on how to link documents, files, and associate web links to instances for use with runtime navigation services, refer to the topic describing how to add user contents (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).

NOTE: When an instance is being edited by using another client of the same system server, this instance can only be opened with the **Instance Editor** in read-only mode to preserve data integrity, page 70.

Modifying Instance Parameters After Assignment

When you modify and save properties of an instance whose facets are already assigned to a Participant project, the software may display a message to inform you that your modifications affect the assignment status of these facets.

This is a generic message that does not take into consideration whether the assigned facets are generated or not.

In fact, only the assignment status of generated facets can be changed to **Out Of Date** or **Deleted**, page 411.

If facets are only assigned, upon saving your changes:

- Modifications are propagated without affecting the assignment status. (For example, if you change the instance name, the identifier of assigned facets is updated in the Participant project.)
- Entire facets are removed from the container to which they are assigned. (For example, if you clear an element of an instance, the corresponding facet that was assigned is deleted.)

Changing the Element Selection of Instances with Assigned Facets

When you change the element selection of an instance, which has at least one facet that is already assigned and this change results in the creation of new facets, the software can assign candidate facets automatically to the same container.

For more information, refer to the topic describing automatic facet assignment, page 368.

Validating Parameter Values

When you edit parameters of the instance, the **Instance Editor** displays and validates values according to the supported data types, page 86, helping you configure the values in accordance with the data type format.

You cannot save the instance properties if the software has detected that a parameter value is not valid.

Certain supported data types, page 86 use language-specific formatting according to the **Regional Options** that are set for the operating system of the PC. For example, if your regional settings define . (dot) as decimal delimiter and you enter a decimal value in the format 100, 5 by using comma instead of the dot, the software processes this value as 1005.0.

NOTE: Except for changes to the **\$Name** parameter:

- Changes appear in italics until you save them.
- Values that you enter and that are different from the default value are shown in bold at any time.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Verify which **Regional Options** are configured in the operating system of your computer and follow the applicable formatting rules to enter data in the **Instance Editor**.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

In addition, in the **Instance Editor** the following rules apply when you edit the **\$Description** parameter and other element parameters of type String of an instance.

Rule	Example or comment
"" returns an empty string.	“example or example” is not allowed.
Conversion of “ into ^”.	The software converts “example” into ^“example^” to return “example”. ^example or example^ is not allowed.
To return ^ use ^.	^^example^^ returns ^example^.

Naming Rules for Instance Parameters

When you change instance parameter values, verify that they satisfy the Participant specific naming rules, page 88 so that the subsequent generation process can complete successfully.

Further, entering incorrect values may change the **Data** status, page 180 of the instance to **Invalid** when you save the instance configuration.

NOTE: Naming rules may also apply when you use the hierarchical naming function.

Hierarchical Naming Function

You can use the hierarchical naming function to let the software manage the identifier, page 171 of an instance. The function is enabled by default.

When you select the hierarchical naming function, the software creates the identifier of an instance automatically through the concatenation of:

- The alias of each application folder, page 166 in the path of the instance, starting from the system root folder.
- The **\$Name**, page 184 parameter of the instance.

For example, if the **\$Name** of an instance is *Motor_1* and the path to its location is: *System Root Folder\Folder_1\Folder_2*.

Plus, you have assigned the aliases *F1* and *F2* respectively to each folder.

Then, the identifier of the instance becomes *F1F2Motor_1*.

NOTE:

- The length of the instance identifier cannot exceed the maximum length, page 88; otherwise the software displays the status of the instance as **Invalid**, page 180.
- The software excludes folders without alias from the hierarchical naming (for example, if you assign no alias to *Folder_1*, the identifier of the instance becomes *F2Motor_1*).
- You cannot assign an alias to the system root folder.

To use the hierarchical naming function, proceed as follows.

Step	Action
1	Right-click the instance and select Properties . Result: The Instance Editor opens.
2	Verify that Hierarchical Name is selected, enabling the hierarchical naming function; otherwise select it.
3	Close the Instance Editor .
4	Right-click the application folder that is to be considered for the hierarchical naming and select Properties . Result: The Properties tab of the folder opens.
5	Assign an alias to the folder in the Alias field. NOTE: Keep the alias as short as possible to avoid reaching the instance identifier character limit.
6	Repeat steps 4 and 5 for the other folders that you want to include in the hierarchical naming.
7	If required, modify the \$Name of the instance in the Instance Editor to avoid reaching the instance identifier character limit.

NOTE: The software applies the hierarchical naming when you move, copy, page 203, or import, page 246 instances in the **Application Explorer**.

Instance Parameters

You can configure the following instance parameters in the parameter pane of the **Instance Editor**.

Selection in the instance pane	Parameter	Data type	Description
Instance	\$Name	String	<p>Default value: Template identifier <i>_n</i> where <i>n</i> is an incremental number starting at 1. For Schneider Electric templates, the \$ suffix and the _xy prefix are omitted.</p> <p>Within the application of one system, the resulting instance identifier, page 171 must be unique for each instance using the same template; otherwise, the \$Name field is outlined in red. Clicking outside of the field reverts the name to its previous value.</p> <p>Entering a character that is not allowed, page 88 changes the instance Status to Invalid, page 171.</p> <p>NOTE: If you enter the same value for two instances that use a different template and the resulting instance identifiers are identical, the \$Name field is outlined in red if the identifiers of Control facets of the two instances are not unique anymore (for example, when both instances contain the Interlocks element, which is created by the same facet).</p> <p>NOTE: The software changes the \$Name parameter of the instance if the value is not unique anymore within the application following a paste, page 207 or import, page 246 operation.</p>
	\$Description	String	<p>Description of the instance.</p> <p>Default value: Blank.</p>
Instance	\$Area	Integer ⁽¹⁾	<p>Area that the instance is assigned to.</p> <p>Range: 1...255</p> <p>Default value: Blank.</p>

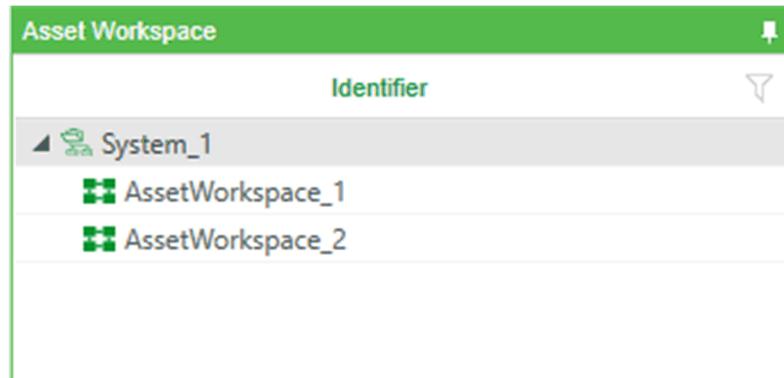
Selection in the instance pane	Parameter	Data type	Description
			<p>NOTE: If you have assigned an area value to the folder, page 167 containing the instance, the software automatically assigns the same value to the instance unless you assign a value manually.</p> <p>If the software is not able to propagate the value of the Area parameter to any item because the item is locked, it:</p> <ul style="list-style-type: none"> • Displays a notification to inform you which item is locked. • Rolls back any propagated area value. • Reverts the Area parameter value that you have entered in the folder properties to its previous value. <p>NOTE: The area value is also propagated to Supervision elements of instances such as genies. However, the change to blank (NULL) from a non-blank value is not propagated to genies, page 494 that are already assigned to a Supervision page (for example, changing the area value from 3 to 4 is propagated but not changing it from 3 to blank).</p>
Control node	–	–	<p>You can configure the values of the parameters of the core and optional Control services that the instance provides through the selected elements.</p> <p>The default values are defined in the respective elements of the template that the instance uses.</p> <p>For details on parameters of Schneider Electric application templates, refer to the help, page 101 of the Schneider Electric libraries for EcoStruxure Process Expert.</p>
Supervision node	–	–	<p>You can configure the values of the parameters of the optional Supervision services that the instance provides through the selected elements.</p> <p>The default values are defined in the respective elements of the template that the instance uses.</p> <p>For details on parameters of Schneider Electric application templates, refer to the help, page 101 of the Schneider Electric libraries for EcoStruxure Process Expert.</p>
HyperLink node NOTE: Requires using templates of a library that supports this functionality.	–	–	<p>You can configure the parameters of the optional Documents service that let you link documents to the instance. You can open these documents on computers running the operation client when using runtime navigation services (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).</p> <p>You can link up to five documents.</p> <p>You must add documents that you want to link to the content repository first.</p> <p>For details, refer to the topic describing how to link documents to instances (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).</p> <p>You can configure the parameters of the optional URL service that let you associate web links and files to the instance. You can open these links on computers running the operation client when using runtime navigation services (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).</p> <p>You can link up to five files or URLs.</p> <p>For details, refer to the topic describing how to associate web links and files to instances (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).</p>
(1) You can also enter a string instead of an integer value. In such case, you must refine the Supervision project after generating it and use the same string as label to name the area; otherwise, the Supervision project does not build successfully. Refer to the help of the Supervision Participant for details.			

Using the Asset Workspace Pane

This topic describes the **Asset Workspace** pane. It lets you manage asset workspaces and open them in the **Asset Workspace Editor** to manage links between instances, page 193.

Asset Workspace Pane

The following figure shows an example of the **Asset Workspace** pane, which contains two user-created asset workspaces. **System_1** is the system root node.



Root Node Actions

The table describes the commands of the context menu of the system root node in an asset workspace.

Command	Description
Create Workspace	Creates a new empty asset workspace. This is a structural element that lets you group assets, which are existing instances of the application in order to create links between them.
Paste	Lets you create a copy of the asset workspaces and their content that you have added to the Clipboard. You can paste within the same system only.
Export	Lets you export, page 219 the data of existing asset workspaces and instances that belong to them to an editable application export file in comma-separated value (CSV) format.
Import	Lets you import, page 219 the data of workspaces and their instances, which are contained in an application export file (.csv).

Asset Workspace Actions

The table describes the commands of the context menu of asset workspaces.

Command	Description
Edit Workspace	Opens the asset workspace, page 193 in the Asset Workspace Editor . Lets you add and remove instances and edit links between them.
Show Instances	Opens the Show Instances window, which lets you see information about the instances that have been added to this asset workspace. Double-clicking an instance opens the asset workspace and highlights the instance.
Copy	Copies the data of the asset workspace and instances that belong to it to the Clipboard.
Export	Lets you export, page 219 the data of this asset workspace and instances that belong to it to an editable file in comma-separated value (CSV) format.
Delete	Deletes the asset workspace but does not delete the instances from the application nor the links between them.
Rename	Lets you rename the asset workspace.

Instance Indicators

Overview

This topic describes the various indicators that appear on folders and instances of the application.

Instance Progress Indicators

Overview

Various icons give an indication of the steps that remain before the instantiation and assignment, page 328 stages for each instance are completed.

The color of progress icons is used to indicate instance validity, page 191.

A progress icon is not visible while the checked-out icon, page 326 is shown.

NOTE: The icons are visible to any user who opens the application.

Meaning of Icons

The following progress information is valid only for instances whose facets are not yet generated.

Icon	Condition description
(0% progress)	The instance is created or has been pasted without further action.
	Either one applies: <ul style="list-style-type: none"> The instance properties, page 182 have been modified at least once but no assignments have been performed. The instance progress is 50% or 75% and existing assignments, page 328 have been removed. The instance has been imported without impact on Participant projects.
	Either one applies: <ul style="list-style-type: none"> Facets of the instance have been assigned but there is still one type of Participant project to which no facet has been assigned (for example, to a Supervision project). At least one facet that is assigned has a status, page 366 other than Assigned. The instance has been imported and has changed the assignment status of at least one facet to Out Of Date or Deleted.
	At least one facet of the instance has been assigned to a Control and Supervision Participant project respectively and each one has the Assigned status.
No icon (100% progress)	All the facets of the instance have the Assigned status.

How Icons Appear

The following figure shows an example of the **Application Browser** displaying several instances with various completion states, which are indicated by progress icons.

Identifier	Template	Version	Data	Link
System_1				
Folder_1				
Valve_1	SValve	2.1.14	Valid	Valid
MyMotor_1	SMotor	3.1.17	Valid	Valid
MyDOutput_1	SDigitalOutput	2.7.10	Valid	Valid
MyDInput_1	SDigitalInput	3.1.10	Valid	Valid
MyAInput1_1	SAnalogInput1	4.1.16	Valid	Valid

Instance Validity Indicator

Overview

A dot or progress icon, page 189 that appears in red color indicates that the configuration of an instance is invalid.

When the icon appears on at least one instance, it appears automatically on folders in the hierarchy of the instance.

NOTE: The indicator icon is visible to any user who opens the application.

Meaning of the Icon

Level	Icon color	Condition description
Instance	Red (progress icon or dot for objects with 100% progress)	The instance Data status, page 171 and/or Link status, page 197 is invalid.
Folder	Red (dot)	The icon appears on at least one instance in the child structure.

Instance Write-Lock Indicator

Overview

A lock icon lets you see that an instance or folder is being used or modified by another process or user and that you cannot work on it.



Certain objects that are locked can be opened in read-only mode.

NOTE: The indicator icon is visible to any user who opens the application.

Meaning of the Icon

The lock icon is shown, for example, when either operation is performed on an instance or folder:

- Properties are open (instances only)
- Paste
- Rename
- Export/import
- Update/replace template

Linking Instances by Using the Asset Workspace Editor

This topic describes how to add instances to asset workspaces and link them by using their exposed interfaces. It also describes how to manage interface links.

Asset Workspace Editor

Overview

The **Asset Workspace Editor** lets you link the exposed application interfaces of an instance to the interfaces of other instances that are compatible and respect the cardinality of the roles (for example, 1 to 1, or 1 to n). The editor lets you create, modify, or delete links between instances, page 199 that belong to a same asset workspace.

You can use the editor, for example, to map a signal to an application object such as a limit switch signal to the appropriate input of a pump motor.

Some instances require that you link them to other instances based on the interface rules of their templates; otherwise the status of the **Link**, page 171 of the instance remains **Invalid** and you cannot proceed further in the system engineering life cycle.

NOTE:

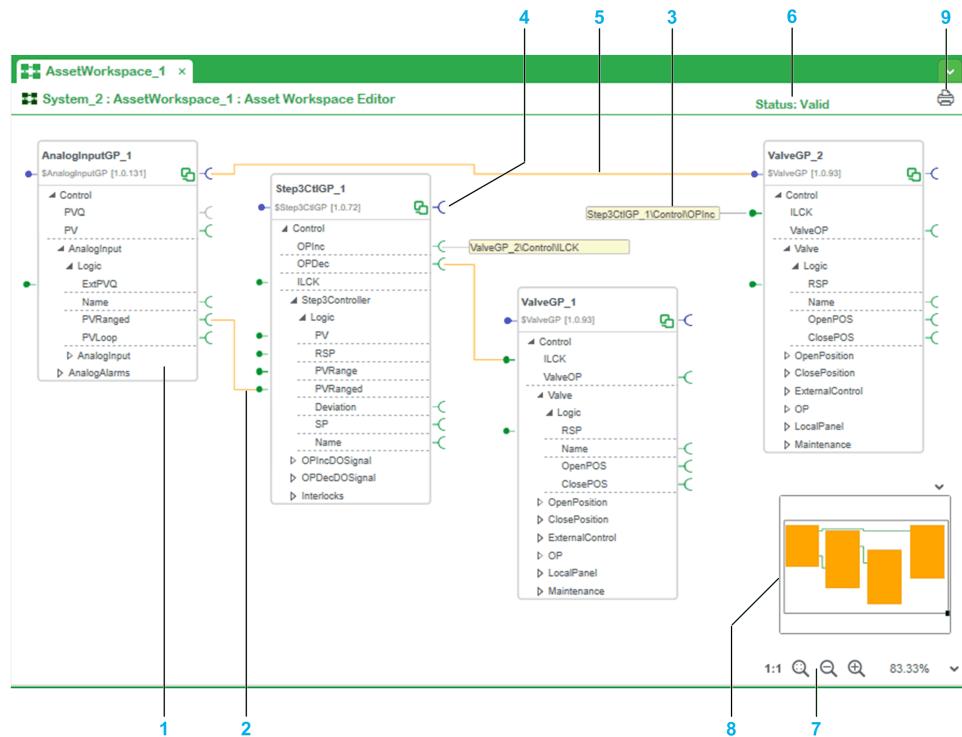
- Linking instances by using the **Asset Workspace Editor** helps avoid having to create such links later by refining the Control Participant project and provides a level of consistency in terms of Control resource names.
- Linking an instance to itself is not allowed. If this is required, edit the template used by the instance or refine the Control project to which facets of the instance are assigned.

Opening the Asset Workspace Editor

To open the **Asset Workspace Editor**, right-click an asset workspace and select **Edit Workspace**.

Asset Workspace Editor

The following figure shows an example of the **Asset Workspace Editor**.



Item	Description
1	Instances that have been added to the asset workspace by dragging them from the Application Browser or importing them. Linked instances that do not belong to the asset workspace are not shown. An instance can belong to an asset workspace even if it is not linked.
2	Existing application interface links and new ones. Only links between instances that belong to the same asset workspace are shown.
3	Interface link that was converted to a connector by using the link context menu command.
4	RTNS interfaces, page 201 (runtime navigation services). Appear in blue and are located in the header section of the instance.
5	RTNS interface links.
6	Indication of the link status of the instance that is selected and button to open the Validity Diagnosis window if the status is Invalid .
7	Zoom tools, page 869.
8	Radar View, page 877.
9	Lets you print the visible portion of the asset workspace.

NOTE: You can also move the editor background, page 106.

Instance Actions

The table describes the commands of the context menu of instances that appear in the **Asset Workspace Editor**.

Command	Description
Show/Hide Disabled	Lets you show or hide interfaces that are disabled (gray) and not available for connection.
Show/Hide Unbound	Lets you show or hide interfaces that are not connected to another instance in the same asset workspace. RTNS interfaces are always shown.
Remove Instance	Removes the instance from this asset workspace but does not delete it from the application. In addition, if the instance is linked to another instance in the same asset workspace, you can perform either action: <ul style="list-style-type: none"> • Delete the links that this instance has with any other instances in the same asset workspace. (Links to instances that are not in this asset workspace are maintained.) • Remove the instance but maintain its links.
View Assignments	Opens the View Assignments window, page 379.
Properties	Opens the instance, page 176 in the Instance Editor .

Link Actions

The table describes the commands of the context menu of links that appear in the **Asset Workspace Editor**.

Command	Description
Convert to Connector	Lets you toggle between .
Convert to Line	
Go to	Opens a submenu that indicates the following: <ul style="list-style-type: none"> • The identifiers of the two instances that are linked. • For each instance, the name of the element that exposes the linked interface and its path.
Delete	Deletes the link.

Impact of Modification of Instance Properties

When you save changes to an instance in the **Instance Editor** applicable changes are automatically reflected on the instance in the **Asset Workspace Editor**. If a change disables an interface that was linked, the link is automatically removed.

This may set the **Link** status, page 197 of either linked instances to **Invalid**.

Viewing Interfaces and Interface Description

Not all interfaces of instances that are displayed in the **Asset Workspace Editor** are visible by default. You may need to expand nodes of the interface to show enabled (green) or disabled (gray) interfaces.

To enable an interface, edit the instance and select the element that contains the interface, and/or configure the element parameter that enables it.

Move the pointer over any interface in the editor to display a description of the interface in a tooltip.

NOTE: Verify that disabled and unbound interfaces are shown by using the corresponding commands in the context menu of instances.

Using the Simplified View

Overview

Zooming out to the lowest level in the **Asset Workspace Editor** switches the display to the simplified view.

This view mode lets you see at a glance the instances and the relation between them without showing the details of links, interfaces, and elements of instances.

Some engineering functionality is not available.

To exit the simplified view, zoom in.

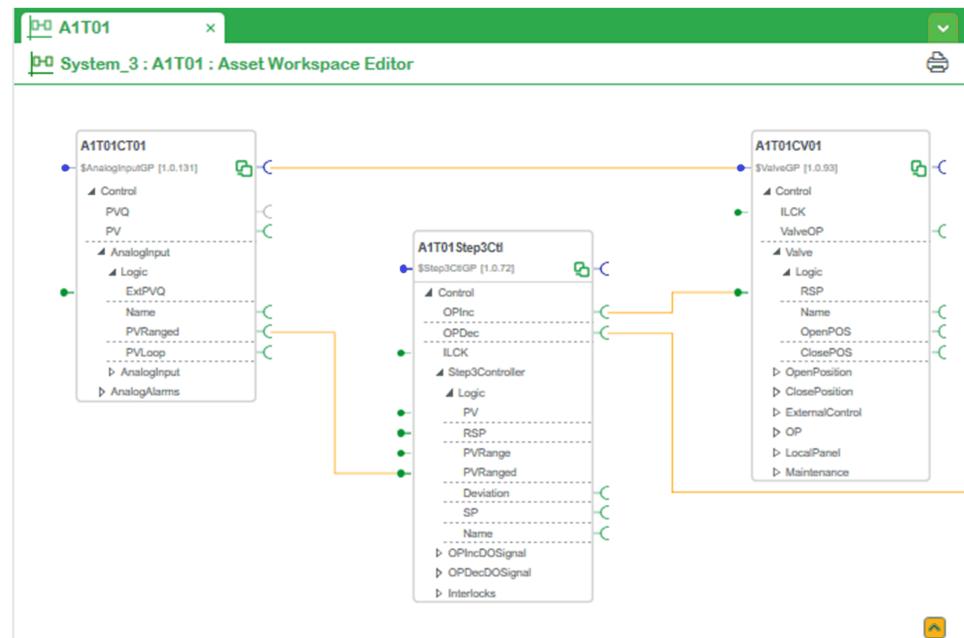
Restrictions of the Simplified View

The table describes the restrictions that are in effect while the simplified view is active.

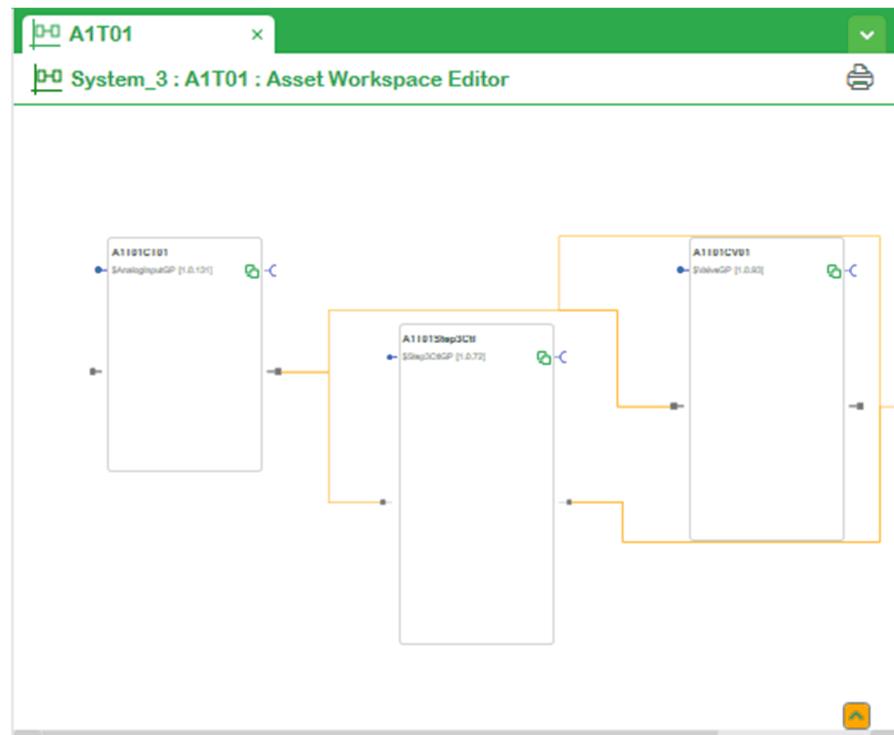
Actions or object	Restrictions
Actions related to links such as creating and moving	Actions are not allowed
Elements and interfaces	Are not shown
Multiple links to and from an instance	Are shown as a single line

Description

The following figure shows an example of instances as they normally appear in the **Asset Workspace Editor**.



The following figure shows the same view when simplified view is active (the view is enlarged for better visibility).



- Instances maintain their relative size and position to preserve the aspect of the layout.
- Links between instances are grouped into a single line.

Instance Link Status

Overview

The **Link** status of an instance can be one of the following:

- **Valid:** The interface rules and interface element rules that are defined at the template level are satisfied.
- **Invalid:** At least one of the defined interface rules or interface element rules are not satisfied.

You cannot assign to projects instances that have a **Link** status that is **Invalid**.

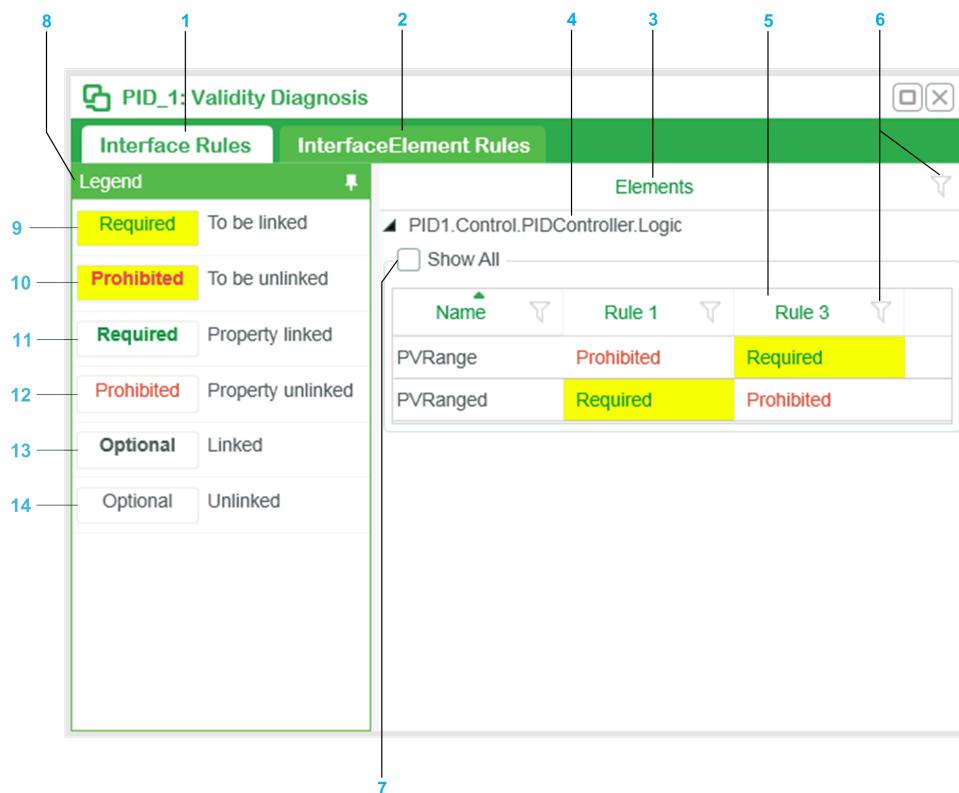
When the status is displayed as **Invalid**, click the validity status information button



or **Invalid** in the **Application Browser** to open the **Validity Diagnosis** window, which contains information about the cause of the invalid status.

Validity Diagnosis Window - Interface Rules

The following figure shows an example of the **Interface Rules** tab of the **Validity Diagnosis** window.



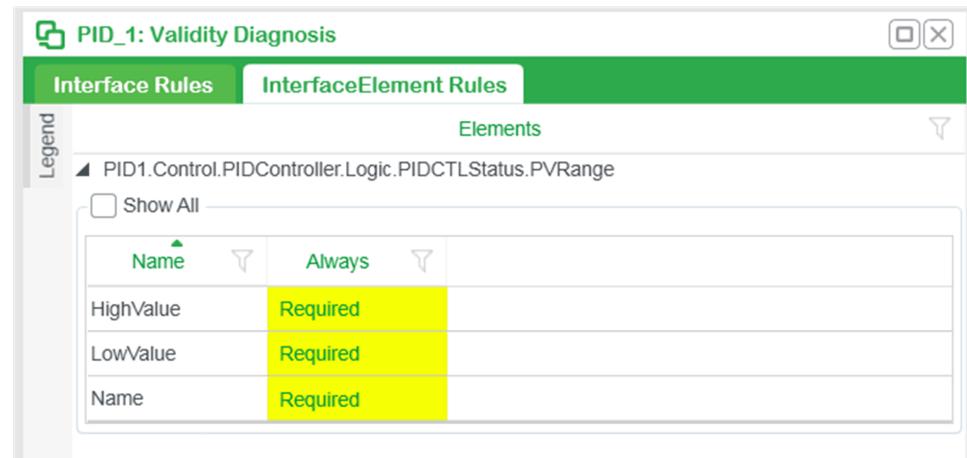
Item	Description
1	Tab displaying information on interface rules of the instance with invalid Link status.
2	Tab displaying information on rules of interface elements, page 199 of the instance with invalid Link status.
3	By default, this section displays only the elements of the instance for which at least one interface rule is not satisfied.
4	Path to the element whose interface rules are shown. You can view the element by editing the instance, page 176.
5	Shows for each element, the rules that are defined and their status.
6	Filter, page 118.
7	Select the check box to view elements of the instance for which an interface rule is defined independently if the rule is satisfied or not.
8	Hover with the pointer over Legend to view an explanation of the statuses displayed in the rule columns.

The table describes the content of the **Legend** tab of the **Validity Diagnosis** window.

Item	Description
9	The link is required to satisfy the rule.
10	Remove the link; otherwise the Status remains Invalid .
11	The link is made, which satisfies the rule.
12	No link is made in accordance with the rule.
13	The link is optional and a link is made.
14	The link is optional and no link is made.

Validity Diagnosis Window - Interface Element Rules

The figure shows the **Interface Element Rules** tab, which is available if rules exist for elements of an interface. In this example, interface **PVRange**, which is one of the interfaces shown in the **Interface Rules** tab has three elements for which a rule is defined but not satisfied.



Linking Instances

Linking Instances That Are Open for Editing

While an instance is open in the **Instance Editor**, you can add it to and remove it from, page 193 the **Asset Workspace Editor** but you cannot create, modify, nor delete links for it.

Naming Rules for Variable Names

The name of the variable that is created by linking two instances needs to satisfy the naming rules, page 89 for the subsequent generation process to complete successfully.

Creating Interface Links

To create an interface link between two or more instances by using the **Asset Workspace Editor**, proceed as follows.

Step	Action
1	Right-click an asset workspace and select Edit Workspace . Result: The asset workspace opens in the editor.
2	From the Application Browser pane, drag the instances that you want to link to the editor and position them as needed. Result: The instances are shown in the editor with their interfaces exposed. If links already exist between these instances, they are shown as well. NOTE: You can drag a maximum of 20 instances at once either by selecting them individually and/or selecting folders.
3	From either instance, position the pointer over an interface and when the pointer turns into a hand, click and move the pointer away from it while holding down the left mouse button. Result: A line appears and follows the pointer. You can release the mouse button. NOTE: Press ESC to return to the normal pointer function.
4	Move the pointer over the other instance. Result: Compatible and enabled interfaces of the instance are automatically highlighted and the line snaps to the nearest matching connector.
5	Click to confirm the connection or move the pointer to another highlighted interface until the line snaps to it. Result: A click performs the following actions: <ul style="list-style-type: none"> Links the interfaces of both instances with a line. Changes the status of the active instance to Valid if the interface rules are satisfied (if applicable). NOTE: To delete the link, right-click the line and select Delete .
6	Verify that the link status of the instances is Valid ; otherwise click Invalid in the Link column of the Application Browser or in the Asset Workspace Editor to open the Validity Diagnosis window and check which links are required. Create these links for the instance status to become Valid .

Changing the Destination of a Link

A destination interface is represented graphically by a plug .

It is typically the interface with the *REF* role.

To change the destination of an interface link, proceed as follows.

Step	Action
1	Select the link whose destination you want to change. Result: A black dot appears on the destination connector.
2	Drag the black dot to a new compatible destination connector of the same or another instance and release the mouse button. Result: The link is moved to the new destination.

Enabled Interfaces That Cannot Be Connected

The interface of an instance may appear as enabled, unbound, and be compatible but the link does not snap to it. This can be the case if this interface is linked to another instance that is not in the same asset workspace and the interface accepts only one connection at a time.

To view the links that exist for an instance, open it in the **Links Editor** so that it is the active instance or open the **Inspect Instance** window, page 204.

Using the RTNS Interface

Purpose of the RTNS Interface

When no application link is required between instances, you can use the RTNS interface to link them so that they appear in the **Linked Production Assets** section of the operation client (see EcoStruxure™ Process Expert, Runtime Navigation Services, User Guide).

The cardinality of the interface is n to n and it can be connected only to another RTNS interface.

Using the RTNS Interface

You can create a link between any instances that exist in the application by using this interface. The procedure to create an RTNS link is the same as the one to create a regular interface link.

NOTE:

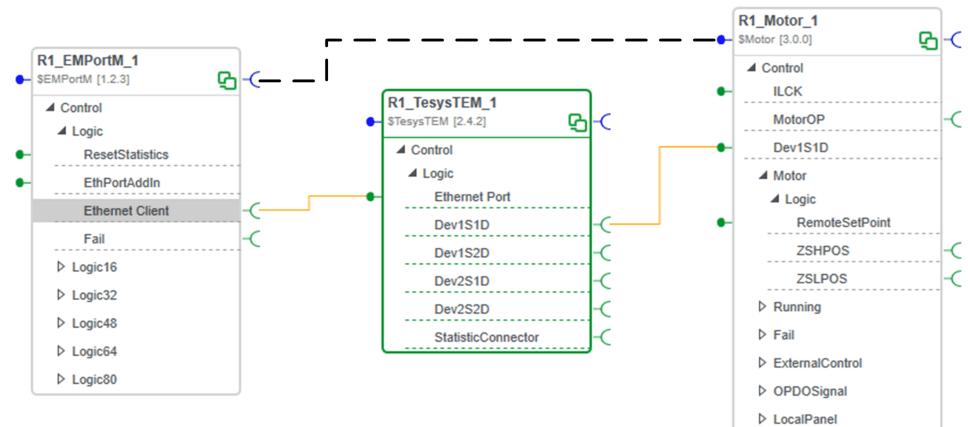
- Creating an RTNS link between two instances that are already linked by an application link has no effect.
- Adding or removing an RTNS link between two instances does not change the assignment status, page 411 of facets of these instances.

RTNS Interface Identifiers and Roles

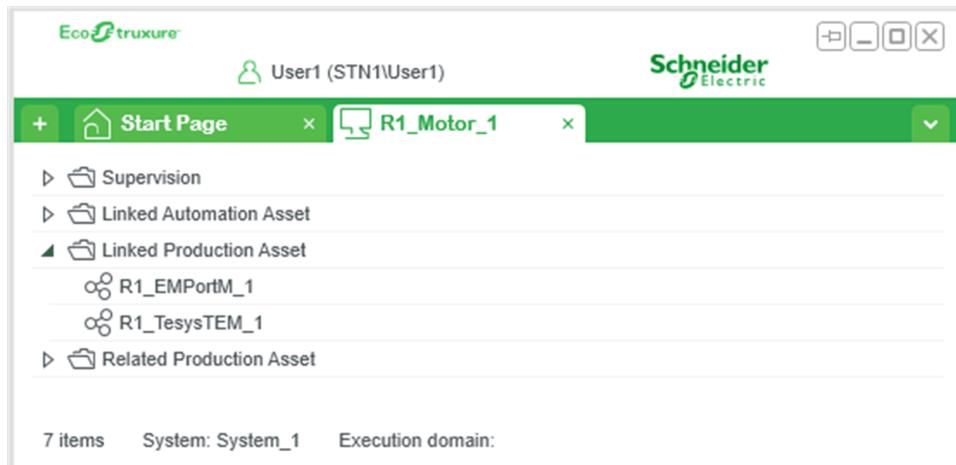
Representation	Identifier	Role
	<i>NullInterfaceRoleA</i>	A
	<i>NullInterfaceRoleB</i>	B

Example

In the following example, three instances are linked but no link exists between *R1_EMPortM_1* and *R1_Motor_1*. To be able to view also information of *R1_EMPortM_1* in the operation client when accessing runtime navigation services from *Motor_1*, you can create an RTNS link with *R1_EMPortM_1* (shown here as a dotted line for illustration purposes).



The following figure shows the result in the EcoStruxure Process Expert operation client after creating the RTNS link when you access runtime navigation services from *R1_Motor_1*. Instance *R1_EMPortM_1* also appears in the **Linked Production Assets** section because of its RTNS link with *R1_Motor_1*.



Managing Application Folders and Instances

Overview

This section describes the **Instance Inspector** and the functionalities that are available in the software to manage many application folders and instances.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Viewing Instance Configuration

Overview

The **Inspect Instance** command lets you view the entire composition of any instance of the application along with current values of its parameters, elements, functions, and interfaces.

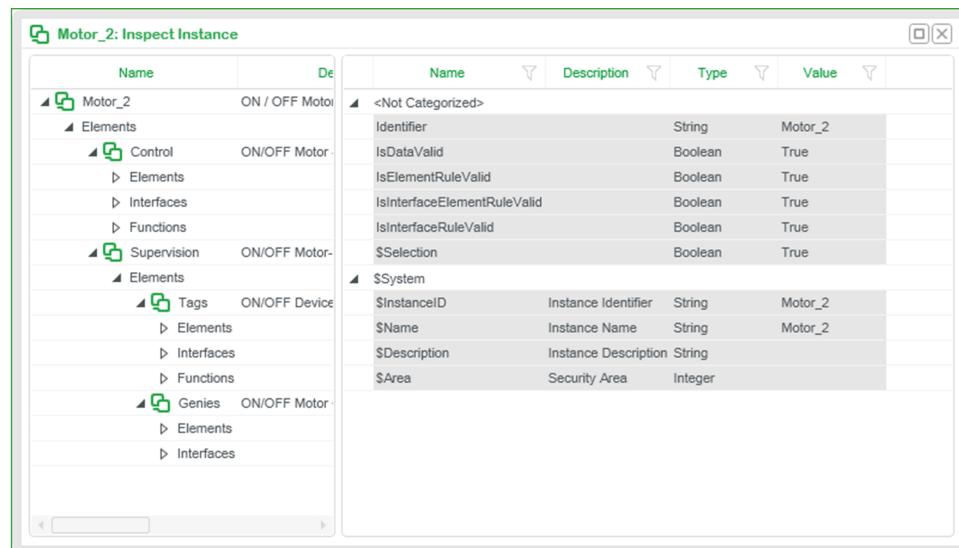
The functionality does not allow you to change values.

Opening the Inspect Instance Window

In the **Application Browser**, right-click an instance and select **Inspect Instance**.

Inspect Instance Window

The following figure shows an example of the **Inspect Instance** window for an instance of the application.



The table describes the various components of an instance that you can view in the left-hand pane of the **Inspect Instance** window.

Component	Description
Instance identifier	Identifier of the instance from which you have opened the Inspect Instance window.
Elements	Elements of the instance grouped by Participant (for example, Control) Inside each category, you can view the hierarchy of elements as they are visible in the Instance Editor down to the constituent level.
Functions	Binding functions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) that are used by the parent element. For example, it lets you view information related to the location of an instance in the application, which can be provided by the LocationGet function.
Interfaces	Interfaces that are referenced by the parent element. For each interface, you can view its identifier, role used by the element, its template identifier, parameters, and their current value.

The table describes the parameter information that you can view in the right-hand pane of the **Inspect Instance** window. The data pertains to the component that is selected in the left-hand pane of the window.

Header	Description
Name	Name of the parameters of the element grouped by category.
Description	Description of the parameter.
Type	Data type of the parameter.
Value	Current resolved value of the parameter, page 176 or property.

Changing the Application Hierarchy

Changes Impacting Assigned and Generated Supervision Facets

When you change the hierarchy of the application, facets of instances that are assigned to a Supervision project and generated are affected when they generate equipment based on this hierarchy, page 425. The table details these changes and their impact.

Change	Impact
Moving an application folder to a different parent folder.	The assignment status of any facets of any instances that the folder and its subfolders contain is set to Out Of Date .
Modifying the identifier of a folder.	
Moving an instance to a different folder.	The assignment status of any facets of the instance is set to Out Of Date .

For information on steps that you need to take to integrate the changes into the system, refer to Generating Control or Supervision Projects After Changes, page 417.

Copying and Pasting Application Folders and Instances

Description of the Functionality

You can copy and paste instances and application folders:

- Within the application of a system.
- Between applications of different systems.

The **Notification Panel** provides detailed information about the result of the paste process.

NOTE: Pasting instances affects the instance count, page 174.

The table describes the effects of the copy and paste functionality when applied to application folders and instances.

Item	Effect
Folders and instances	<p>The copied folder and any instances that it contains, including any subfolders and their contents are pasted.</p> <p>NOTE:</p> <ul style="list-style-type: none"> • When you select the Copy command for a folder, this folder, any subfolders, and any instances they contain are copied. This is the case even if you initially selected, in addition to the folder, only one of the instances or subfolders it contains with the intention to copy only the selected items. • If you delete an instance or a folder after copying it, you cannot paste it. Also, if you copy a folder containing several instances and you delete an instance in this folder, when you paste the folder, the instance that you deleted is not pasted.
Folder identifier and instance name	<p>If the folder identifier already exists at the same hierarchical level, or the instance identifier already exists in the target location, the software changes the folder identifier or the instance name by adding an incremental numerical value as suffix; otherwise the identifier or instance name is not changed.</p> <p>For example, <i>Motor_1</i> becomes <i>Motor_1_1</i>, and <i>Folder_1</i> becomes <i>Folder_1_1</i>.</p> <p>NOTE: If the hierarchical naming function, page 183 is activated for an instance, the software changes the instance identifier according to the new location of the instance and the alias of the folders in its path.</p>
Interface links	<p>Interface links are maintained between pasted instances like they were defined between the original instances if you copy and paste the instances that are linked at the same time.</p> <p>Interface links can be copied across systems under the same conditions.</p>
Other user-defined configuration data	<p>The software does not modify the data during the paste process. Therefore, relative expressions that the software has created at the instantiation stage could produce different results when applied to the new location.</p> <p>Also, when you paste a folder and/or instance in a folder to which you have assigned an area value, page 166, this value is propagated to the pasted items if they do not have an area value yet and if they are not locked. For details refer to the topic describing the propagation of the area parameter value, page 167.</p>

Aborting Paste Tasks

After selecting the **Paste** command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. No instances and folders are pasted.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are pasting only a few items, the task may complete before you are able to click the icon.

Copying and Pasting Instances

To copy and paste instances, proceed as follows.

Step	Action
1	Expand the folder that contains the instances that you want to copy.
2	Select the instances. You can make multiple selections within one folder by holding: <ul style="list-style-type: none">• The Shift key for contiguous instances.• The Ctrl key if instances are non-contiguous.
3	Right-click a selected instance and select Copy .
4	Right-click the destination folder and select Paste . You can select multiple folders as targets. NOTE: If the target is not valid for pasting, the command is not available. Also, you cannot paste in the Application Browser in grid view mode.

Copying and Pasting Application Folders

To copy and paste application folders, proceed as follows.

Step	Action
1	Right-click the folder that you want to copy and select Copy . You can make multiple selections within the Application Browser by holding the Ctrl key.
2	Right-click the destination folder and select Paste . You can select multiple folders as targets. NOTE: If the target is not valid for pasting, the command is not available. Also, you cannot paste in the Application Browser in grid view mode.

Moving Application Folders and Instances

Overview

You can move instances and application folders:

- Within the application of a system.
- Between the applications of different systems.

Both actions change the structure of the application, page 154. The software displays a notification informing you of the possible impact on the assignment status of already assigned facets before executing the move operation.

Once the move operation is completed, if the software had to make changes to folder and/or instance identifiers to satisfy naming rules, it displays the **Move Summary** window, page 211 to inform you of the changes.

NOTE:

- You cannot move instances within a folder.
- You can change the position of folders without modifying the hierarchy of the application, page 211.

Moving Folders and Instances

By using the **Application Browser** pane, you can select a combination of folders and instances to move them at once. This is possible in tree view mode only.

When you move a folder, this folder, any subfolders, and any instances they contain are moved.

This is the case even if you initially selected, in addition to the folder, only one of the instances or subfolders it contains with the intention to move only the selected items.

NOTE:

- You can make multiple selections by holding the **Shift** key for contiguous and the **Ctrl** key for non-contiguous items.
- If sorting and/or filters are applied, they are also applied to instances and folders that you have moved.

Effects of the Move Operation

The table describes the effects of the move operation when applied to application folders and instances.

Item	Effect
Folders and instances	Moves the folder and any instances that it contains, including any subfolders and their contents.
Folder identifier and instance name	If the folder identifier already exists at the same hierarchical level, or the instance identifier already exists in the target application, the software changes the folder identifier or the instance name by adding an incremental numerical value as suffix; otherwise the identifier or instance name is not changed. For example, <i>Motor_1</i> becomes <i>Motor_1_1</i> , and <i>Folder_1</i> becomes <i>Folder_1_1</i> . NOTE: If the hierarchical naming function, page 183 is activated for an instance, the software changes the instance identifier according to the new location of the instance and the alias of the folders in its path.

Item	Effect
Interface links	Interface links are maintained between instances if you move the instances that are linked at the same time. NOTE: Moving Interface links across systems is supported.
Other user-configured properties	The software does not modify the data during the move process. Therefore, relative expressions that the software has created at the instantiation stage could produce different results when applied to the new location. Also, when you move a folder and/or instance to a folder to which you have assigned an area value, page 166, this value is propagated to the moved items if they do not have an area value yet and if they are not locked. For details refer to the topic describing the propagation of the area parameter value, page 167.

Moving Instances

To move instances, proceed as follows.

Step	Action
1	Expand the folder that contains the instances that you want to move.
2	Select the instances. You can make multiple selections by holding the Shift key for contiguous and the Ctrl key for non-contiguous instances. Result: The selected instances are highlighted. NOTE: To move instances across systems, use two workspaces, page 105; one for each application.
3	Drag one of the highlighted instances to the target application folder. Result: The identifiers of the instances you are moving appear in a tooltip.
4	Release the mouse button. Result: The software: <ul style="list-style-type: none">• Moves the selected instances to the new location. When no sorting is applied, instances are inserted in ascending alphabetical order among existing instances.• Displays the Move Summary window if the software has performed a change to comply with applicable naming rules.
5	Close the Move Summary window.

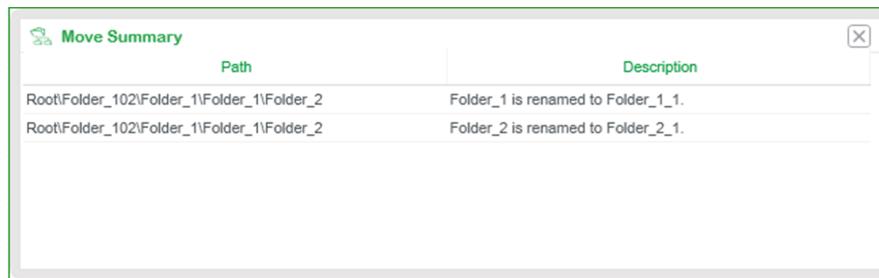
Moving Application Folders

To move application folders, proceed as follows.

Step	Action
1	Select the folders that you want to move. You can make multiple selections by holding the Shift key for contiguous and the Ctrl key for non-contiguous folders. Result: The selected folders are highlighted. NOTE: To move folders across systems, use two workspaces, page 105; one for each application.
2	Drag one of the highlighted folders that you want to move to the target. Result: The identifiers of the folders you are moving appear in a tooltip.
3	Release the mouse button. Result: The software: <ul style="list-style-type: none">• Moves the folders and their contents to the new location. When no sorting is applied, folders are added at the end of existing folders.• Displays the Move Summary window if the software has performed a change to comply with applicable naming rules.
4	Close the Move Summary window.

Move Summary Window

The following figure shows an example of the **Move Summary** window.



Path Indicates the path to the folder or instance location after the move.

Description Indicates the name change that the software has performed on the folder and/or instance during the move.

Changing the Position of Folders Without Changing the Hierarchy

You can change the position of folders inside the structure without modifying the hierarchy of the application. You can do this one folder at a time.

Step	Action
1	Verify that no sorting is applied in the Application Browser pane of the Application Explorer and that tree view mode is selected.
2	In the Identifier column, start dragging a folder along the folder structure until one of the following messages appears in a tooltip: <ul style="list-style-type: none"> • Move X before Y. • Move X after Y Where: <ul style="list-style-type: none"> • X is the identifier of the folder that you are moving. • Y is the identifier of the target folder.
3	Release the mouse button. Result: The software relocates the folder to the selected position. NOTE: If you release the mouse button while only the identifier of the folder that you are moving appears in the tooltip, you are changing the hierarchy of the folder structure, page 209.

Updating and Replacing Templates of Instances

Overview

In the **Application Browser**, you can perform the following actions on the template that is used by an instance of the application. The working principle of both commands is the same. The difference resides in the choice of templates provided by each command.

Update Template	Lets you update the template that is used by the instance with the latest available version. When you select the command from a folder, the command applies to any instance that is contained in the folder and any subfolders, and for which a template of a newer version is available.
Replace Template	Lets you replace the template that is used by one or more instances with another version of this template or a different template.

NOTE: The **Update Template** command is also available in the **Topological Explorer** for certain topological entities, page 550.

Updating Versus Replacing Templates

The following examples are given to illustrate the difference between the two commands.

Update Template command:

- The following templates are available in the Global Templates library: *\$Motor (2.0.0)*, *\$Motor (2.0.1)*, and *\$Motor (2.0.2)*
- Instance *Motor_1* uses template *\$Motor (2.0.0)*
- The command lets you update the template that is used by *Motor_1* with *\$Motor (2.0.2)* only (latest version).

Replace Template command:

- The following templates are available in the Global Templates library: *\$Motor (2.0.0)*, *\$Motor (2.0.1)*, and *\$Valve (3.0.0)*
- Instance *Motor_1* uses template *\$Motor (2.0.1)*
- The command lets you replace the template that is used by *Motor_1* with either:
 - *\$Motor (2.0.0)*: Same template, any available version.
 - *\$Valve (3.0.0)*: Different template.

Using the Commands on a Folder

When you select the **Update Template** or **Replace Template** command at the folder level, the command is performed on this folder, any subfolders, and any instances they contain.

This is the case even if you initially selected, in addition to the folder, only one of the instances or subfolders it contains with the intention to perform the command only the selected items.

Accessing the Commands

Update Template	Right-click one instance, a selection of several instances in the same or in various folders, the root folder, or any application folder, and select the command. The command executes only if a newer version of the template that is used by the instance is available in the Global Templates library.
Replace Template	Right-click one instance or a selection of several instances in the same folder, and select the command.

Description of the Functionality

The software proceeds with the update or replacement of a template in the following order:

- The template that is associated to the instance is updated or replaced.
- Interface links are processed.

The following conditions need to be satisfied:

- The template or template version that you want to use must be available in the Global Templates library.
- It needs to have the usability state **Approved** (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) or **Deprecated**.

NOTE: Refer to Importing Templates, page 889.

The table describes the effects on the various template components when you update or replace the template that is used by an instance.

Template component	Identifier in new template ⁽¹⁾	Effect
Element	The identifier of the element and its path in the composition of the template is the same as in current template.	<ul style="list-style-type: none"> • The element is maintained or updated. If updated, the status of the corresponding facet is set to Out Of Date if the facet was generated. Refinements that you have made are preserved if the element that creates the facet encapsulates the same constituents. • The element selection that you have made in the current template is applied to the new template. For example, in the current and the new template, the Interlocks element is at the path Control\Interlocks. If it was selected in the current template, it is selected in the new template. • If the identifier of the template used by the corresponding facet (Facet Template column) is different, the status of the facet is set to Out Of Date if the facet was generated. <p>NOTE: If you replace a control module template (highest level template) by one that has a different identifier but an identical composition, no changes are made to the instance, except for the template identifier.</p>
	The identifier of the element does not exist in the current template or the path to the element is different.	<p>The element is added and the corresponding facet is created.</p> <p>An optional element that is not part of the current template keeps its default value (selected or cleared).</p> <p>NOTE: Any element of the current template whose identifier does not exist in the new template or whose path is different is removed.</p>
Parameter value	Is same as in current template	Explicitly defined parameters of the current template keep their respective value in the new template.
	Is not in current template	New parameters that are not part of the current template keep their default value.
Interface link	–	<p>The software re-establishes interface links that are valid after the template update or replacement in their original configuration.</p> <p>The software discards interface links that are not valid after the template update or replacement because, for example:</p> <ul style="list-style-type: none"> • The cardinality of the interface role is not respected. • A linked element does not exist anymore. • An incompatibility of the type and/or role of the interface.

(1) The term *new template* refers to either a different version of the same template or a different template

This example illustrates the impact of a template update operation on generated Control facets of an instance. In this scenario, instance *Valve_1* uses template *\$Valve* version 1.3.9. The optional **Maintenance** service is selected at the instance level.

The following figure shows the composition of the template as it appears in the **Instance Editor**. Elements appear in the **Name** column and the identifier of the template that is used by the corresponding facet in the **Template** column.

Name	Description	Template	Version
Valve_1	(0/262) ON/OFF Valve	\$Valve	1.5.0
Control	(0/23) ON/OFF Valve	SVALVE_UC	1.2.12
Valve	(0/3) ON/OFF Device Control - Unity Control	SDEVCTL_UC	1.0.8
Logic	(0) ON/OFF Device Control Unity Logic	SDEVCTL_UL	1.0.8
LowLimitSwitch	(0) Digital Input Signal	\$DISignal_UL	5.6.9
HighLimitSwitch	(0) Digital Input Signal	\$DISignal_UL	5.6.9
EXTCTLCond	(0) Digital Input Signal	\$DISignal_UL	5.6.9
OPControl	(0) Digital Output Signal	\$DOSignal_UL	5.4.5
Interlocks	(0) Interlock Condition summary with Unity HMI	SCONDSUM1_UC	1.0.8
Maintenance	(0) Device Maintenance Unity Logic with Unity HMI	SDEVMNT_UC	1.0.3
LocalPanel	(0/12) Device Local Panel	SDEVLP_UC	2.1.9

The following figure shows the Control facets (**Facet** column) of instance **Valve_1**, which are assigned to a section of a Control project. By relating this information to the one shown in the **Instance Editor**, you can see, for example, that element **LowLimitSwitch** creates facet **Valve_1_ZSL** by using template **\$DISignal_UL**. This information is also visible at a glance in the **View Assignments** window, page 379.

Instance	Template	State	Facet	Facet Template	Path	Order	Assignment	Generation
\$Valve		Valid	Valve_1_DEVCTL	\$DEVCTL_UL	Control\Valve	0	Assigned	Generated
\$Valve		Valid	Valve_1_ZSL	\$DISignal_UL	Control	1	Assigned	Generated
\$Valve		Valid	Valve_1_ZSH	\$DISignal_UL	Control	2	Assigned	Generated
\$Valve		Valid	Valve_1_VALVE_OP	\$DOSignal_UL	Control	3	Assigned	Generated
\$Valve		Valid	Valve_1_CONDSUM1	SCONDSUM1_UL	Control\Interlocks	4	Assigned	Generated
\$Valve		Valid	Valve_1_DEVMNT	SDEVMNT_UL	Control\Maintenance	5	Assigned	Generated

\$Valve 1.3.9 is updated to version 1.3.10 by using the **Update Template** command. \$Valve 1.3.10 has a modified **Interlocks** service, which is provided by the modified **COND_SUM1_UL** template. All other elements are identical.

The screenshot shows the Application Browser interface with the following details:

- Application Browser** window title.
- Search Application Instances and Folders** search bar.
- Identifier**, **Template**, and **Version** filter buttons.
- System_3** folder expanded.
- Folder_1** folder expanded.
- Valve_1** instance selected, showing its identifier as **SValve** and version as **1.3.9**.
- A modal dialog titled **Valve_1: Update Instance Template** is displayed:
 - Path**: `\Folder_1\Valve_1`
 - Selected Item**: `$Valve 1.3.9 -> 1.3.10`
 - Buttons**: `OK` and `Cancel`

The result of the template update operation shows that one facet has a modified assignment status. All facets remain generated.

FBDSection_1 - Assignments								
Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation	
\$Valve	Valid	Valve_1_DEVCTL	SDEVCTL_UL	Control\Valve	0	Assigned	Generated	
\$Valve	Valid	Valve_1_ZSL	SDISignal_UL	Control	1	Assigned	Generated	
\$Valve	Valid	Valve_1_ZSH	SDISignal_UL	Control	2	Assigned	Generated	
\$Valve	Valid	Valve_1_VALVE_OP	SDOSignal_UL	Control	3	Assigned	Generated	
\$Valve	Valid	Valve_1_CONDSTUM1	SCONDSTUM1_UL	Control\Interlocks	4	Out Of Date	Generated	
\$Valve	Valid	Valve_1_DEVMNT	SDEVMNT_UL	Control\Maintenance	5	Assigned	Generated	

1 2

1 The facet created by the CONDSUM1_UL template (**Interlocks** service) has a status of **Out Of Date** because the already generated constituents (from \$Valve 1.3.9) are not consistent with the constituents of the modified CONDSUM1_UL template anymore. Generating the section injects the constituents coming from the modified CONDSUM1_UL template into the logical Control Participant project.

2 The optional **Maintenance** service remains selected after the template update because it is also available in the latest version of \$Valve.

This example illustrates the impact of a template replacement operation on generated Control facets of an instance. In this scenario, instance *Motor_1* uses template \$Motor. \$Motor has an **Interlocks** service provided by the CONDSUM1_UL template. The optional **Maintenance** service is selected at the instance level.

FBDSection_1 - Assignments								
Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation	
\$Motor	Valid	Motor_1_DEVCTL	SDEVCTL_UL	Control\Motor	0	Assigned	Generated	
\$Motor	Valid	Motor_1_CONDSTUM1	SCONDSTUM1_UL	Control\Failures	1	Assigned	Generated	
\$Motor	Valid	Motor_1_CONDSTUM1	SCONDSTUM1_UL	Control\Interlocks	2	Assigned	Generated	
\$Motor	Valid	Motor_1_DEVMNT	SDEVMNT_UL	Control\Maintenance	3	Assigned	Generated	

\$Motor is replaced with \$Valve by using the **Replace Template** command. Compared to \$Motor, \$Valve has a modified **Interlocks** service, which is provided by the modified CONDSUM1_UL template (located at the same path). The **Maintenance** service is identical in both and provided by the same template at the same path. Their other elements are either different (for example, \$CONDSTUM_UL) or located at a different path (for example, \$DEVCTL_UL) in the composition of the control module template.

The screenshot shows the Application Browser interface with the following details:

- Application Browser:** Shows the hierarchy: System_3 > Folder_1 > Motor_1. The Motor_1 instance is selected, and its properties are shown: Identifier: Motor_1, Template: \$Motor, Version: 3.1.14.
- Modal Dialog: Motor_1: Replace Instance Template**
 - Select Template:** A dropdown menu is open, showing the option '\$Valve [1.3.10]'.
 - Instance:** A dropdown menu is open, showing the option 'Motor_1'.
 - Current Template:** A dropdown menu is open, showing the option '\$Motor'.
 - Buttons:** OK and Cancel.

The result of the template replacement operation shows that several facets have a modified assignment status.

Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation
\$Motor	Deleted	Motor_1_DEVCTL	SDEVCTL_UL	Control\Motor	0	Deleted	Generated
\$Motor	Deleted	Motor_1_CONDSTUM	SCONDSTUM_UL	Control\Failures	1	Deleted	Generated
\$Valve	Valid	Motor_1_CONDSTUM1	SCONDSTUM1_UL	Control\Interlocks	2	Out Of Date	Generated
\$Valve	Valid	Motor_1_DEVMNT	SDEVMNT_UL	Control\Maintenance	3	Assigned	Generated

1 2 3

1 These two facets are deleted because in \$Valve, \$DEVCTL_UL is at a different path and \$CONDSTUM_UL does not exist. Generating the section removes the corresponding constituents from the logical Control Participant project.

2 The facet providing the **Interlocks** service is set to **Out Of Date** because Valve contains this service (provided also by \$CONDSTUM1_UL) but a modified version of it. Already generated constituents (from \$Motor) are not consistent with the constituents of the modified CONDSTUM1_UL template anymore. Generating the section injects the constituents coming from the modified CONDSTUM1_UL template into the logical Control Participant project.

3 The optional **Maintenance** service is not affected because it is identical for both templates. The **Instance Template** column indicates that the facet is now created by the Valve template.

In the **Assignment Editor**, you can see the facets that are specific to Valve and that have been created for the *Motor_1* instance. You need to assign these to the section as needed. These facets have the **Candidate** status. By default, the software assigns them automatically, page 368.

Instance	Instance Template	Facet	Facet Template	Path	System	Project
Motor_1	\$Valve	Motor_1_CONDSTUM1	SCONDSTUM1_UL	Control\Interlocks	Assigned	Done
Motor_1	\$Valve	Motor_1_DEVCTL	SDEVCTL_UL	Control\Valve	Not Assigned	Candidate
Motor_1	\$Valve	Motor_1_DEVMNT	SDEVMNT_UL	Control\Maintenance	Assigned	Done
Motor_1	\$Valve	Motor_1_VALVE_OP	SDOSignal_UL	Control	Not Assigned	Candidate
Motor_1	\$Valve	Motor_1_ZSH	SDISignal_UL	Control	Not Assigned	Candidate
Motor_1	\$Valve	Motor_1_ZSL	SDISignal_UL	Control	Not Assigned	Candidate

NOTE: The same that is described in the above examples for Control facets also applies to Supervision facets of the instance.

Updating or Replacing Templates of Linked Instances

When instances are linked by interface links (**Asset Workspace Editor**), update the templates of these instances at once, page 217 to maintain the links.

Updating or Replacing Templates Containing a New Version of a Type

When you update or replace the template used by an instance and the Control facets that are referenced by the instance are already generated, if the new template that you have selected contains a version of the type that is different from the one that has been generated, the software requires that you select the version to be used when you generate the Control Participant project. In certain cases, generation may not be able to complete successfully. For details, refer to the topic describing the selection of type versions during generation, page 396.

NOTE: To view the version of a type that is used by a facet, edit the facet encapsulating the type by using the **Global Templates** explorer, select the element that represents the type, and click **Templatizer**. The version of the type is indicated in the **Global Types** section of the **Select Variables** window (for example, DEVCTL 0.01).

Validating Instance Identifiers

When you perform a template replacement operation, the software verifies if the result of the operation satisfies the uniqueness requirement for instance identifiers. If it does not, it displays the **Replace Template Summary** dialog box, which indicates which instance creates a duplicate identifier, and cancels the replace operation.

This can be the case when either of the following applies:

- Two instances use a different template but have the same identifier.
- You replace one of the two templates with the same template that the other instance uses.

For example:

- The application contains two instances:
 - *Valve_1*, which uses template *\$HandValve*
 - *Valve_1*, which uses template *\$Valve*
- If you replace template *\$HandValve* by template *\$Valve* (any version), the two instances use the same template and have the same identifier. This is not allowed.

NOTE: When replacing a template creates an element with the same identifier as the element of an existing instance, you can still proceed with the template replacement but the **Data** status of the instance is set to **Invalid**.

For example:

- The application contains two instances:
 - *Pump_1*, which uses template *\$Motor*
 - *Pump_1*, which uses template *\$HandValve*
- If you replace template *\$HandValve* by template *\$Valve* (any version), although the two instances do not use the same template, they have the element *Logic* in common, which has the same identifier (*Pump_1_DEVCTL*) in both instances because both instances have the same identifier. This is not allowed.

Updating the Template of One Instance

To update the template that is used by one instance, proceed as follows.

Step	Action
1	<p>Right-click the instance that you want to update and select Update Template.</p> <p>Result: The Update Template dialog box opens. It shows the latest version of the template that is available for the update.</p>
2	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> • Updates the template that is used by the selected instance with the latest version of the same template. • Displays the version of the template that is used by the instance after the update in the Version column. • Displays the result of the update process in the Notification Panel. <p>NOTE: Click Cancel to close the dialog box without updating the template.</p>

NOTE: To revert to an earlier version of a template, use the **Replace Template** command.

Updating the Templates of Several Instances

To update the templates of several instances of the application at once, proceed as follows.

Step	Action
1	<p>Right-click the root folder of the application and select Update Template.</p> <p>Result: The Update Template dialog box opens, which:</p> <ul style="list-style-type: none"> Lists the templates, which are used by instances of the application for which a template of a later version is available. It indicates the current version of the template and the latest version that is available for the update. Indicates, for each template, the instances that are using the template with an indication of the location of the instances (Path).
2	<p>Select the instances that you want to update:</p> <ul style="list-style-type: none"> Instance by instance. By template. Through a combination of both.
3	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> Proceeds with the update of the template that is used by the selected instances. Displays the version of the template that is used by each instance after the update in the Version column. Displays the result of the update process in the Notification Panel. <p>NOTE: Click Cancel to close the dialog without updating the templates.</p>

NOTE: If you use a filter in the **Name** column of the **Update Template** dialog box and as a result, not all the instances that you had selected are displayed, the update operation is still performed on the entire selection of instances.

Replacing Templates of Instances

To replace the template that is used by one or more instances, proceed as follows.

Step	Action
1	<p>Select one or more instances and/or one or more folders.</p> <p>Result: The software highlights the selected items.</p>
2	<p>Right-click a highlighted item and select Replace Template.</p> <p>Result: The Replace Template dialog box opens.</p>
3	<p>Click the Select Template list.</p> <p>Result: The software displays the list of templates that are available for the replace operation.</p>
4	<p>Select the template that you want to use by scrolling or entering a keyword in the field of the menu.</p> <p>To search by entering the first letter of the template identifier, enter also the \$ prefix for Schneider Electric templates (for example, \$m).</p>
5	<p>Click OK.</p> <p>Result: If the result of the replacement operation does not satisfy the identifier uniqueness requirement, the software displays the Replace Template Summary dialog box and cancels the replace operation.</p> <p>If no conflict is detected, the software:</p> <ul style="list-style-type: none"> Closes the dialog box and replaces the template that is used by the instances with the template that you have selected. Displays the identifier and the version of the template used by the instances respectively in the Template and Version columns. Displays the result of the replace process in the Notification Panel. <p>NOTE: Click Cancel to close the dialog without replacing the template.</p>

NOTE: If you use a filter in the **Instance** column of the **Replace Template** dialog box and as a result, not all the instances that you had selected are displayed, the replace operation is still performed on the entire selection of instances.

Exporting and Importing Application Objects to/from CSV and XML Formats

Overview

The export and import functionality lets you create comma-separated value (CSV) or XML files and import them into the application of the same or another system.

This allows you to do the following:

- Move or copy the entire application or a selection of folders, asset workspaces (CSV only), page 222, and instances to another system.
- Perform bulk processes such as creating, modifying and/or deleting instances, folders, asset workspaces (CSV only), and interface links by editing the exported file.

This topic describes how to export objects, edit the export file to perform the required changes, and import it.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Exporting Folders and Instances to CSV and XML Formats

Overview

You can export application folders, instances, and interface links by using the **Export** command of the **Application Explorer**.

The application objects and their respective configuration parameters are saved to an export file in comma-separated, page 228 value (CSV) or XML format.

For example, you can use the export file to import, page 246 the application into another system while modifying some parameters, which helps avoid having to create it from scratch. Also, you can use it to restore instances after deleting them or to restore them to an earlier state.

NOTE: You also need to export templates, page 886 that are used by the instances that you are exporting if the same version of template is not present in the Global Templates library of the target system at the time of import. Instances whose template is not present cannot be imported.

Good Practices

- If you intend to edit the export file, try to limit your export selection to the objects that you want to modify or delete, for example, by using filters, page 118 in the **Application Browser**. It reduces the steps to manage unwanted objects during import and speeds up the process.
- If you want to create instances, export, at least, one instance that uses the same template. If you want to create interface links, export, at least, two instances that are linked. This lets you use the data as model. For example, you can copy and paste it and modify only necessary location, element, and/or parameter values.

Export Format Selection

The CSV, page 223 and XML, page 236 export files contain similar information and let you perform the same actions.

Data in CSV format can appear more comprehensible and easier to manage by using a spreadsheet editor because of its user-friendly, tabular presentation. Also, it facilitates the integration of existing data, such as an asset list, depending on their format.

Exporting to XML may be preferred by users who are familiar with the use of structured data and more powerful XML editors.

NOTE: You cannot update existing interface links by using the XML format.

Invariant Culture

The software uses the invariant culture, page 228 to format parameter values independently of the language setting that is used on the computer on which you create the export file.

For example, a decimal value that appears as 5,5 (comma separator) in the **Instance Editor** because of your regional settings appears as 5.5 (dot separator) in the export file.

Software Version Compatibility

You can import into a system instances that were exported from an earlier version of the software.

Some restrictions apply, page 246.

Object Selection

You can select the following objects to export them:

- One or several instances: Exports the data pertaining to the selected instances and their location in the folder structure of the application. You can select individual instances located in various folders and export them at once.
- One or several application folders: Exports the data pertaining to the selected folders, the instances contained in the folders, and in any subfolders.
- The root folder of the application: Exports any application folders and instances of the system.

An interface link is exported only if both linked instances are exported.

NOTE: When you select the **Export** command for a folder, this folder, any subfolders, and any instances they contain are exported. This is the case even if you initially selected, in addition to the folder, only one of the instances or subfolders it contains with the intention to export only the selected objects.

Aborting Export Tasks

After confirming the export by clicking **Save** in the **Save As** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The export file is not created.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are exporting only a few objects, the task may complete before you are able to click the icon.

Creating the Export File in CSV or XML Format

To export application objects, proceed as follows.

Step	Action
1	In the Application Browser , select the objects that you want to export. Use filters and sorting, page 118 to facilitate the selection.
2	Right-click your selection and select Export . Result: The Export dialog box opens.
3	Select the location, file name, and format for the export file.
4	Click Save . Result: <ul style="list-style-type: none"> • The export file containing the selected objects is created. • A summary of the exported objects is displayed in the notification pane.

NOTE: If any object that you have selected for export is open in an editor (for example, the **Instance Editor**), accessed for an assignment, update, or replace operation, or if its **Properties** tab is open, you cannot create the export file.

Exporting Asset Workspaces to CSV Format

Overview

You can export asset workspaces and the instances that belong to them by using the **Export** command.

The application objects and their respective configuration parameters are saved to an export file in comma-separated value (CSV), page 228 format.

NOTE: You also need to export templates, page 886 that are used by the instances that you are exporting if the same version of template is not present in the Global Templates library of the target system at the time of import. Instances whose template is not present cannot be imported.

Object Selection

You can select the following objects to export asset workspaces:

- The root node of the **Asset Workspace** pane: Exports the existing asset workspaces and the instances that belong to them.
- One or several asset workspaces: Exports the workspaces and the instances that belong to them.
- The root folder of the application: Exports any application folders, asset workspaces, and instances of the system.

Creating the Export File Containing Asset Workspace Data

For information on how to create the export file, refer to the topic describing the export of folders and instances, page 220.

CSV Application Export File

Export File Description

When you export application objects to comma-separated, page 228 format (CSV), two files are created.

File	Default file name	Content and purpose
Application export file	<System identifier>	<p>Data of the folders, instances, interface links, and asset workspaces that you have exported.</p> <p>The data is organized in five types of groups.</p> <p>You need to edit the content, page 612 of this file to create, modify, or delete application objects.</p>
Application interface data supporting file	<System identifier>_Pending Interfaces where _Pending Interfaces is appended to the file name once the file is created.	<p>Data on application interfaces that the exported instances expose and that are not connected. It includes interfaces that are disabled.</p> <p>The data is organized by instance name.</p> <p>It helps you edit data of the InterfaceLinks group in the application export file by providing the necessary data of each interface that you can connect.</p>

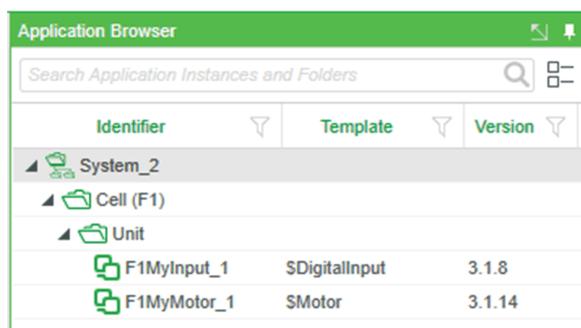
The following table describes the grouping of data in the application export file. Each group has a group header and a header row, which describes the data of objects.

Group header	Description
<i>Folders</i>	<p>Groups that contain folder data, page 228.</p> <p>More than one group exists when you export folders that have a different number of parameters as part of the optional folder properties.</p> <p>The number of columns depends on the folder configuration. A column with its header exists for each system parameter and each configured optional parameter.</p> <p>NOTE: A group header and its header row appear only if the export file contains folder data.</p>
<i>TemplateIdentifier=<Template ID> Version=<Value> where:</i> <ul style="list-style-type: none"> • <Template ID> is the identifier of the template that is used by the instance. • <Value> is the template version. Example: <i>TemplateIdentifier=\$Motor Version=3.1.0</i>	<p>Groups that contain instance data, page 228.</p> <p>There are as many groups as templates and versions used by exported instances.</p> <p>The number of columns depends on the instance configuration. A column with its header exists for each element and parameter of the instance.</p> <p>NOTE: A group header and its header row appear only if the export file contains instance data.</p>
<i>InterfaceLinks</i>	<p>Group that contains the data of interface links, page 228, which link exported instances.</p> <p>The number of columns and their headers are fixed.</p> <p>NOTE: The group and its header row are always present.</p>

Group header	Description
<i>Workspaces</i>	Group that contains the data of the exported asset workspaces, page 228. NOTE: The group and its header row are always present.
<i>WorkspacelIdentifier=<Asset workspace identifier></i> Example: <i>WorkspacelIdentifier=AssetWorkspace_1</i>	Group that contains the data of instances that belong to an asset workspace. There are as many groups as exported asset workspaces. NOTE: The group and its header row are present even if the exported asset workspace contains no instance.

Example of CSV Export Files

The following example shows two instances, which have been linked by an interface link.

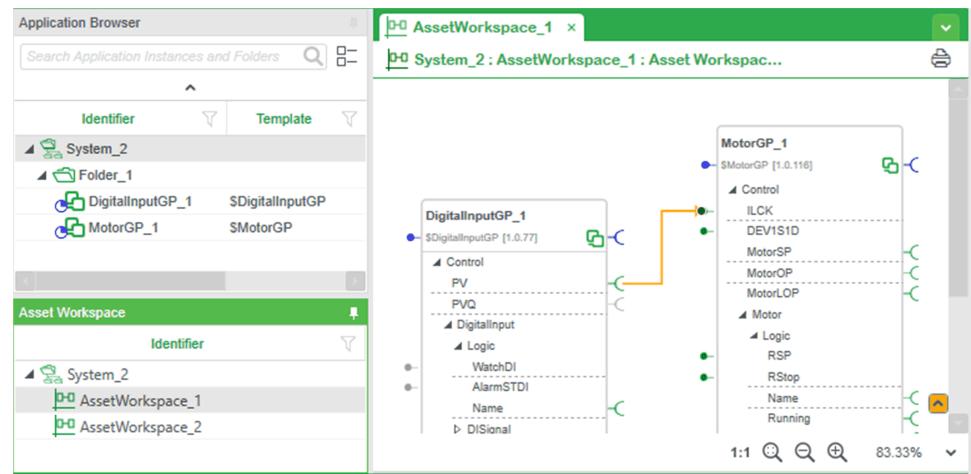


The following figure shows a partial view of the resulting export file when you export folder *Cell* to CSV format.

A	B	C	D	E	F	G	H
1 Export Version=2019	Created on=05/25/2019 09:04	System=System_2	D Encoding=UTF-8				
2 :Folders	SLocation	SIdentifier	S Alias	S Description	S Area		
3 \$Action							
4 Create	Root	Cell	F1				
5 Create	Root	Unit					
6							
7 :TemplateIdentifier=SDigitalInput Version=3.1.8							
8 \$Action	\$InstanceName	Identifier(Non_Editable)	S IsHierarchical	S Location	S Description	Control.Enabled(Non_Editable)	Parameter.D.SignalName
9 Create	MyInput_1	F1MyInput_1	TRUE	Root\Cell\Unit		TRUE	
10							
11 :TemplateIdentifier=SMotor Version=3.1.14							
12 \$Action	\$InstanceName	Identifier(Non_Editable)	S IsHierarchical	S Location	S Description	Control.Enabled(Non_Editable)	Parameter.Control.Running.Negate
13 Create	MyMotor_1	F1MyMotor_1	TRUE	Root\Cell\Unit		TRUE	FALSE
14							
15 :InterfaceLinks							
16 \$Action	\$SourceLocation	\$SourceInstanceName	\$SourceInterface	\$SourceInterfa	\$DestinationInstanceName	\$DestinationInstanceTemplate(De	
17 Create	Root\Folder_1	MyInput_1	SDigitalInput	Control	PV	MyMotor_1	SMotor

Item	Description
1	Export file information that is added by the software.
2	Group headers.
3	Header rows.
4	Element of the instance and its selection state.
5	Parameter of the instance and its configured value. Parameters appear right after their parent element.
6	Parameter of the instance for which no value is configured.
7	Data rows of exported objects (one row per object, such as a folder, an instance, or an interface link).

The following example shows two asset workspaces that exist in **System_2**. **AssetWorkspace_1** contains two linked instances and **AssetWorkspace_2** contains none.



The following figure shows the workspace related portion of the resulting export file when you export **System_2** to CSV format either from the root node of the **Application Browser** or the **Asset Workspace**.

A	B	C	D	E	F
22					
23 :WorkSpaces					
24 :\$Action	\$Location	\$Identifier			
25 Create	Root	AssetWorkSpace_1			
26 Create	Root	AssetWorkSpace_2			
27					
28 :WorkspaceIdentifier=AssetWorkSpace_1					
29 :\$Action(Non_Editable)	\$InstanceLocation	\$InstanceName	\$InstanceIdentifier	\$TemplateIdentifier	\$Position(X,Y)
30 Create	Root\Folder_1	MotorGP_1	MotorGP_1	\$MotorGP	455,68
31 Create	Root\Folder_1	DigitalInputGP_1	DigitalInputGP_1	\$DigitalInputGP	125,153
32					
33 :WorkspaceIdentifier=AssetWorkSpace_2					
34 :\$Action(Non_Editable)	\$InstanceLocation	\$InstanceName	\$InstanceIdentifier	\$TemplateIdentifier	\$Position(X,Y)
35					
36					

1 2 3 4

Item	Description
1	Group headers.
2	Header rows.
3	Rows containing data of the exported asset workspaces.
4	Rows containing data of the instances that belong to each exported asset workspace. NOTE: The data of their interface links and configuration parameters appears in their respective groups in the same export file.

The following figure shows a partial view of a supporting file that is created in addition to the actual export file.

	A	B	C	D	E
1	:Interfaces				
2	:\$Location	\$InstanceName	\$InstanceTemplateIdentifier	\$InterfacePath	\$InterfaceIdentifier
3	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	NullInterfaceRoleA	NullInterfaceRoleA
4	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	NullInterfaceRoleB	NullInterfaceRoleB
5	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	Control	PVQ
6	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	Control\DIInput\Logic	WatchDI
7	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	Control\DIInput\Logic	DInputName
8	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	Control\DIInput\Logic	AlarmSTDI
9	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	Control\DIInput\DISignal	BoolVar
10	Root\Cell\Unit	F1MyInput_1	\$DigitalInput	Control\DIInput\DISignal	DINPUTSignalExt
11	Root\Cell\Unit	F1MyMotor_1	\$Motor	NullInterfaceRoleA	NullInterfaceRoleA
12	Root\Cell\Unit	F1MyMotor_1	\$Motor	NullInterfaceRoleB	NullInterfaceRoleB
13	Root\Cell\Unit	F1MyMotor_1	\$Motor	Control	ILCK
14	Root\Cell\Unit	F1MyMotor_1	\$Motor	Control	MotorOP
15	Root\Cell\Unit	F1MyMotor_1	\$Motor	Control	Dev1S1D
16	Root\Cell\Unit	F1MyMotor_1	\$Motor	Control\Motor\Logic	ZSHPOS
17	Root\Cell\Unit	F1MyMotor_1	\$Motor	Control\Motor\Logic	ZSLPOS
18	Root\Cell\Unit	F1MyMotor_1	\$Motor	Control\Motor\Logic	DEVCTLName
19	Root\Cell\Unit	F1MyMotor_1	\$Motor	Control\Running	BoolVar

Editing the CSV Application Export File

Overview

By editing the application export file (.csv), you can create, update, and/or delete application objects.

Various rules and requirements apply when you open and edit CSV export files. They are described in this topic.

You can use the supporting file (**_PendingInterfaces.csv*) to help with the editing of interface link data.

The software applies the changes when you import the edited CSV file, page 246 into the application of the system. At that time, you have the possibility to review and modify the list of imported objects (except interface links).

Good Practices

- Before editing an application export file, create a copy of it. If needed, it lets you restore the original application.
- Edit only an up-to-date export file to help avoid data conflicts when importing.
- Carefully remove from the file objects that are not impacted by the import, for example, instances that you are not updating or deleting.
- When you use data from the supporting file to create or update interface links, cut the data rather than copying it to help avoid connecting the same interface several times.

The other way around, add to the supporting file data of interfaces that you are disconnecting.

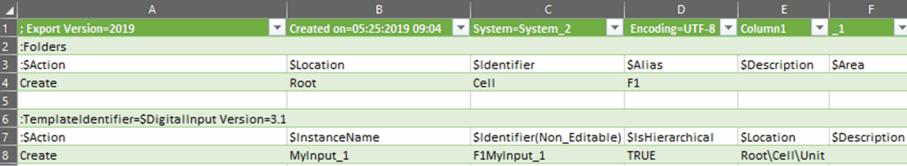
Opening the CSV File by Using a Spreadsheet Editor

When you open a CSV export file by using a spreadsheet editor, some instance parameter values (for example, durations with millisecond values, certain special characters, or strings starting with - (hyphen)) may get modified by the editor.

So that these values remain unchanged, you first need to import the data into a blank workbook in *Unicode (UTF-8)* format.

Follow this procedure. Microsoft® Excel® is used as an example. Menu items and dialog box names may vary depending on the version.

Step	Action
1	Open a blank workbook.
2	In the Data menu, select to import from text or CSV.
3	In the Import Text File dialog box, select the exported CSV file.
4	Select the following attribute values for the import of data: <ul style="list-style-type: none"> • File origin: <i>65001 : Unicode (UTF-8)</i> • List separator (delimiter): <i>Comma</i> • Format for all columns (data type): <i>Text⁽¹⁾</i> • <i>Use First Row as Header⁽¹⁾</i> (depending on version): Open the menu and select this setting even if it is already selected by default.
5	Load the values to the blank workbook. <p>NOTE: After loading is completed, the first row of the CSV file must start with <i>;Export Version=<Software Version></i>. You may see additional text in columns E and higher of the first row (for example, <i>Column1, _1</i>). This text is treated as comment, page 234 and ignored during import.</p>

Step	Action
	The following example shows an extract of a CSV export file after selecting <i>Use First Row as Header</i> and loading values to the blank workbook. 
6	Save the workbook in CSV <i>UTF-8 (comma delimited)</i> format (keep saving in this format when you edit the file). (1) You may need to edit the file during the import process to access the attribute.

Setting the List Separator Format

CSV export files use the comma as list separator (delimiter) independently of the setting that is in effect on the computer on which the file is created.

If the format of the list separator on the computer is different from comma, data will not be displayed correctly when you open the export file by using a spreadsheet editor.

To verify the setting and change it to comma if necessary, open the Windows® **Control Panel** and click **Clock and Region > Region**. In the **Formats** tab, click **Additional settings....**

NOTE: For information on how to use commas in values of string data types in the CSV file, refer to the topic describing editing rules, page 234.

Using the Invariant Culture

Use the invariant culture to modify or add parameter values in the application export file.

If required, during import the software converts the format according to the language setting that is used on the target computer.

The table describes the formats of the invariant culture that are used in the export file.

Type	Format	Example
Decimal separator	x.y	2.5
Time	hh:mm:ss	14:30:00
Date	mm/dd/yyyy	11/20/2014
Date Time	mm/dd/yyyy hh:mm:ss	11/20/2014 14:30:00
Duration	DD.hh:mm:ss.ms	10.12:30:10.15 translating into 10 days 12 hours 30 min 10 sec 15 msec

Application Export File Content Requirements

This topic describes the columns that appear in the each type of group and the requirements for each one. Use this information when you edit existing data, add columns, and/or add entire groups. For details on the format of the group headers, refer to the topic describing the export file, page 223.

Folders group, page 223. The information applies to each group that appears in the file or that you add.

Header	Description	Requirement
\$Action	Action that is performed on the folder during import.	Possible values (not case-sensitive): <ul style="list-style-type: none"> • Create (default) • Update • Delete • Blank: <ul style="list-style-type: none"> ◦ Creates the folder if it does not exist yet. ◦ Updates the folder if it exists.
\$Location	Path to the folder in the application starting from the system root folder. Example: <i>Root\Site\Area\Unit_A</i> NOTE: The name of the system root folder, page 155 is replaced by <i>Root</i> .	A column with this header and a valid value in each row are required. The path must exist in the application or be created by the import. For folders whose parent is the system root folder, enter <i>Root</i> .
\$Identifier	Folder identifier. NOTE: You cannot update the identifier of a folder through import.	A column with this header and a valid value in each row are required.
\$Alias	Folder alias, page 166.	Column and value are optional.
\$Description	Folder description.	If a row contains data, it must be valid.
\$Area	Folder area, page 166. NOTE: If the value is inherited from a parent folder, page 167, it does not appear in the export file.	
Parameter x.\$Identifier	Identifier of the folder parameter, page 166.	Column and value are optional. If a row contains data, it must be valid.
Parameter x.\$Description	Description of the folder parameter.	
Parameter x.\$Value	Value of the folder parameter.	

TemplateIdentifier group, page 223 for instances. The information applies to each group that appears in the file or that you add.

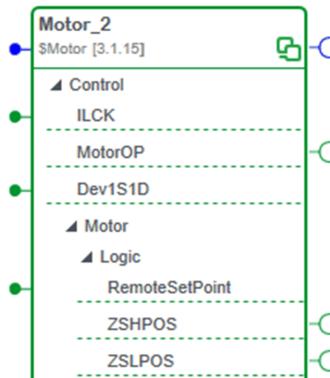
Header	Description	Requirement
\$Action	Action that is performed on the instance during import.	Possible values (not case-sensitive): <ul style="list-style-type: none"> • Create⁽¹⁾ (default) • Update • Delete • Blank: <ul style="list-style-type: none"> ◦ Creates the instance if it does not exist yet. ◦ Updates the instance if it exists.
\$InstanceName	Instance name, page 184.	A column with this header and a valid value in each row are required.
\$Identifier(Non_Editable)	Instance identifier, page 171. The value is used to identify the instance or calculated by the software in case of instance creation.	A column with this header is required and: <ul style="list-style-type: none"> • To create an instance, the value can be left blank. • To update or delete an instance, the value must be valid.

Header	Description	Requirement
\$IsHierarchical	Hierarchical naming function, page 183.	Column and value are optional. Possible values: <ul style="list-style-type: none"> • True • False • Blank (or not present): <ul style="list-style-type: none"> ◦ Equals true for an instance that is created. ◦ Keeps the configured value for an existing instance.
\$Location	Path to the instance in the application. Example: <i>Root\Site\Area\Unit_A</i> NOTE: The name of the system root folder, page 155 is replaced by <i>Root</i> .	A column with this header and a valid value in each row are required. The path must exist in the application or be created by the import.
\$Description	Instance description.	Column and value are optional.
\$Area	Instance area, page 184. NOTE: If the value is inherited from a parent folder, page 167, it does not appear in the export file.	If a row contains data, it must be valid.
<Element path>.Enabled	A column exists for each service (element, page 224) of the instance as they appear in the Instance Editor . The instance identifier is excluded from the element path. Example: For the <i>Running</i> service that belongs to the <i>Control</i> element, the column header is: <i>Control.Running.Enabled</i>	Column and value are optional. Possible values (not case-sensitive): <ul style="list-style-type: none"> • True: The service is enabled. • False: The service is disabled. • Blank: <ul style="list-style-type: none"> ◦ Equals false for an instance that is created. ◦ Keeps the configured value for an existing instance. Mandatory elements (whose check box cannot be cleared in the Instance Editor) are identified with (<i>Non-Editable</i>) in the header. If you set the value to false, it is not considered during import and the element remains enabled.
Parameter.<parameter full path>	A column exists for each parameter, page 224 that is featured by the instance, including parameters of disabled services (elements), page 234 and not configured parameters. Example: For the <i>DISignalName</i> parameter of the <i>Running</i> service, the column header is: <i>Parameter.Control.Running.DISignalName</i>	Column and value are optional. If a row contains data, it must be valid. When you edit a parameter, verify that the element, page 224 to which it belongs is set to true; otherwise, the parameter value is not applied during import. This rule does not apply to parameter values of mandatory elements, which are applied even if the element is set to false in the CSV file. NOTE: To view the hierarchy of elements and parameters of an instance, use the Instance Editor or the View Instance window, page 204. NOTE: For parameters of the enumerated data type, page 240, the name is exported as value. For example, if the parameter value, page 176 that is configured in the Instance Editor is <i>Operator</i> (2), the exported value is <i>Operator</i> . To edit the value, you can enter either the name of the enumerator or the associated integer (for example, 3 for an enumerator that is <i>Supervisor</i> (3)).

(1) Instances with action set to create and that you do not want to import can be unselected during import, page 250.

InterfaceLinks group, page 223.

Parameter	Description	Requirement
\$Action	Action that is performed on the interface link during import.	Possible values (not case-sensitive): <ul style="list-style-type: none"> • Create (default) • Update⁽¹⁾ • Delete • Blank: <ul style="list-style-type: none"> ◦ Creates the link if it does not exist yet. ◦ Updates⁽¹⁾ the link destination if the link exists.
\$SourceLocation	Path to the instance that is the source of the link in the Application Explorer . The name of the system root folder, page 155 is replaced by <i>Root</i> .	A column with this header and a valid value in each row are required.
\$SourceInstanceName	Value of the \$Name parameter, page 184 of the instance that is the source of the link.	
\$SourceInstanceTemplateIdentifier	Identifier of the template that is used by the instance, which is the source of the link.	
\$SourceInterfacePath	Path to the source interface in the composition of the instance. Example: In the following sample instance, the value for interface ZSHPOS is <i>Control\Motor\Logic</i> .	A column with this header and a valid value in each row are required. NOTE: For RTNS interfaces, the column is optional and does not contain values.
\$SourceInterfaceIdentifier	Identifier of the interface that is the source of the link.	A column with this header and a valid value in each row are required. NOTE: Refer to the description of RTNS interfaces for information on their specific identifiers, page 201.



Parameter	Description	Requirement
\$DestinationLocation	<p>Path to the instance that is the destination of the link in the Application Explorer.</p> <p>The name of the system root folder, page 155 is replaced by <i>Root</i>.</p>	A column with this header and a valid value in each row are required.
\$DestinationInstanceName	Value of the \$Name parameter, page 184 of the instance that is the destination of the link.	
\$DestinationInstanceTemplateIdentifier	Identifier of the template that is used by the instance, which is the destination of the link.	
\$DestinationInterfacePath	<p>Path to the destination interface in the composition of the instance.</p> <p>Example:</p> <p>In the following sample instance, the value for interface <i>BoolVar</i> is empty (blank) because it has no parent element.</p>	A column with this header and a valid value in each row are required. NOTE: For RTNS interfaces, the column is optional and does not contain values.
\$DestinationInterfaceIdentifier	Identifier of the interface that is the destination of the link.	A column with this header and a valid value in each row are required. NOTE: Refer to the description of RTNS interfaces for information on their specific identifiers, page 201.
\$NewDestinationLocation	The description is the same as for the \$Destination* parameters but the data applies to the new destination interface of the same or another instance.	Columns with these headers are required. Rows can be left blank.
\$NewDestinationInstanceName	The parameters are empty by default.	The columns must contain a valid value only when you change the destination, page 235 of an existing link.
\$NewDestinationInstanceTemplateIdentifier		
\$NewDestinationInterfacePath		
\$NewDestinationInterfaceIdentifier		

(1) You can only change the destination, page 235 of an existing link by using the *Update* command.

Workspaces group, page 223.

Header	Description	Requirement
\$Action	Action that is performed on the asset workspace during import.	Possible values (not case-sensitive): <ul style="list-style-type: none"> Create (default): Creates the asset workspace or updates an existing one based on the information in the <i>WorkspaceIdentifier=<Asset workspace name></i> group. Delete: Deletes the workspace only. Does not delete from the application the instances that belong to it nor their interface links. Blank: Same as <i>Create</i>.
\$Location	Path to the asset workspace in the Asset Workspace pane. <i>Root</i> by default.	A column with this header and <i>Root</i> as value in each row are required. Asset workspaces can only be located at the system root node level.
\$Identifier	Asset workspace identifier. NOTE: You cannot update the identifier of an asset workspace through import.	A column with this header and a valid value in each row are required.

WorkspaceIdentifier group, page 223. The information applies to each group that appears in the file or that you add.

Header	Description	Requirement
\$Action	Action that is performed on the instance in the context of the asset workspace during import.	Not editable. To remove an instance from an asset workspace, delete the row. To add an instance, add a row and enter the required information in each column.
\$InstanceLocation	Path to the instance in the application.	A column with this header and the same value as in the <i>\$Location</i> column of the <i>TemplateIdentifier</i> group for this instance are required.
\$InstanceName	Name of the instance.	A column with this header and the same value as in the <i>\$InstanceName</i> column of the <i>TemplateIdentifier</i> group for this instance are required.
\$InstanceIdentifier	Identifier of the instance.	A column with this header and the same value as in the <i>\$Identifier</i> column of the <i>TemplateIdentifier</i> group for this instance are required.
\$TemplateIdentifier	Identifier of the template that is used by the instance.	A column with this header and the same template name as in the <i>TemplateIdentifier</i> group header for this instance are required. For example, <i>\$AnalogInputGP</i> if the header for this instance is : <i>TemplateIdentifier=\$AnalogInputGP Version=1.0.131</i> .
\$Position	Position in pixels of the top left corner of the instance representation in the asset workspace .	Two comma-separated values indicating the position on the X and Y axis respectively. For example, 0,0 positions the instance along the left and top borders of the asset workspace.

Editing Rules

The table indicates the rules that apply when you edit a CSV application export file.

Item	Rule						
Header of a group	Must start with a colon.						
First header of a header row	You can remove a group header and its header row if the group contains no data. You can change the order of groups.						
Headers of non-editable data	Headers of data that does not need to be edited or that is ignored during import are identified by the software with the <i>Non_Editable</i> suffix.						
Rows of object data	Refer to the requirements, page 228. You can delete the row of an object if you do not want to import it or modify it through import.						
Empty rows	Can be added anywhere.						
Comment rows	Can be added anywhere and must start with a semi-colon. Comments are ignored during import.						
Order of columns	Can be changed within a group given the first header of the row starts with a colon.						
Empty columns	Can be added after the first column.						
Interface links	For any actions on links, page 235, the data of the linked instances must be included in the application export file even if the instances already exist in the target application.						
List separator	Only comma, page 228 is accepted.						
Values	Use the invariant culture, page 228. To use commas or quotation marks in values of string data types when you edit the file by using a text editor, for example, Notepad, follow these rules: <table border="1"> <thead> <tr> <th>Expected value after import</th> <th>Value to be entered</th> </tr> </thead> <tbody> <tr> <td><i>Motor, Pump</i></td> <td><i>"Motor, Pump"</i></td> </tr> <tr> <td><i>Motor"Pump</i></td> <td><i>"Motor""Pump"</i></td> </tr> </tbody> </table>	Expected value after import	Value to be entered	<i>Motor, Pump</i>	<i>"Motor, Pump"</i>	<i>Motor"Pump</i>	<i>"Motor""Pump"</i>
Expected value after import	Value to be entered						
<i>Motor, Pump</i>	<i>"Motor, Pump"</i>						
<i>Motor"Pump</i>	<i>"Motor""Pump"</i>						

Creating, Updating, and Deleting Folders

To create an instance in a new folder, you must also enter in the CSV file the necessary data to create the folder.

Since each row corresponds to only one folder, to create, for example, the structure *Root\Folder_1\Folder_10*, you must configure two rows.

NOTE: It is not possible to rename a folder by using the **Update** action.

Perform the following steps in the **Folders** groups of the CSV application export file.

Step	Action
1	Edit the rows that contain data of folders and/or create rows. Verify that at least the required columns and data, page 228 are present.
2	Specify a value for \$Action for each row as needed.

Creating, Updating, and Deleting Instances

Perform the following steps in the **TemplateIdentifier** groups of the CSV application export file.

Step	Action
1	Edit the rows that contain data of instances and/or create new groups and/or rows. Verify that at least the required columns and data, page 228 are present.
2	Verify that the data for <i>\$Location</i> is consistent with the target application and/or <i>\$Location</i> in the Folders group.
3	Specify a value for <i>\$Action</i> for each row as needed.

NOTE: If you update data of instances that are already linked and want the link to be maintained, it is not required to update the row of the **InterfaceLinks** group that pertains to the link as long as the linked interfaces remain enabled. This applies even if you change the value of *\$InstanceName* or the instance identifier (for example, by configuring an alias for a folder) for either or both instances.

Creating and Deleting Interface Links

Perform the following steps in the CSV application export file.

Step	Action
1	In the InterfaceLinks group, edit the rows that contain data of the links and/or create new rows. Verify that at least the required columns and data, page 228 are present.
2	Verify that the data for <i>\$SourceLocation</i> and <i>\$DestinationLocation</i> is consistent with the target application and/or <i>\$Location</i> in the other groups (if applicable).
3	Specify a value for <i>\$Action</i> for each row as needed.
4	In the TemplateIdentifier group, verify that the data of the linked instances is entered. If the linked instances already exist in the target application with the same configuration and you do not want to modify them, change the value of <i>\$Action</i> to <i>Update</i> or leave it blank for each instance. NOTE: If the instances already exist in the target application and you leave the value as <i>Create</i> , the Resolve Uniqueness Conflicts dialog box, page 252 opens during import and you need to select the <i>Update</i> action for the instances there.

Changing the Destination of An Existing Interface Link

To move the destination of an existing link to a different interface, edit the CSV application export file as follows. The new destination interface must be available for connection during import.

Step	Action
1	In the InterfaceLinks group, for the row that contains the data of the existing link, change the value of <i>\$Action</i> to <i>Update</i> or leave it blank.
2	Enter the appropriate data in each of the five columns <i>\$NewDestinationLocation</i> to <i>\$NewDestinationInterfaceIdentifier</i> . NOTE: You do not need to edit the data of the existing interface link. If the interface link does not exist in the target application, a link from the specified source to the new destination is created anyway.
3	In the TemplateIdentifier group, verify that the data of the linked instances is included. If the instances already exist in the target application with the same configuration and you do not want to modify them, change the value of <i>\$Action</i> to <i>Update</i> or leave it blank for each instance. NOTE: If the instances already exist in the target application and you leave the value as <i>Create</i> , the Resolve Uniqueness Conflicts dialog box, page 252 opens during import and you need to set the action to <i>Update</i> there.

XML Application Export File

Export File Description

The following example shows the structure of an XML application export file (partial view) that was created by exporting Folder_1, which contains instance DISignal_UL_1.

```

      5   6   7   9   8
<?xml version="1.0" encoding="utf-8"?>
<Export Version="4.1.something">
  <ApplicationFolder Type=
    "SchneiderElectric.ProcessExpert.Application.Transient.ApplicationFolderRecord"
    Identifier="Folder_1" Action="Create">
    <Alias>F1</Alias>
    <Description>My Description</Description>
    <Area />
    <Location>Root</Location>
    <GraphicalLocation>0</GraphicalLocation>
    <Parameters />
    <ApplicationInstances>
      <ApplicationInstance Type=
        "SchneiderElectric.ProcessExpert.Application.Transient.ApplicationInstanceRecord"
        Identifier="F1DISignal_UL_1" TemplateIdentifier="$DISignal_UL" Version="6.3.6"
        IsValid="True" FacetType=
          "SchneiderElectric.ProcessExpert.Unity.Templatization.UnityFacet.Logic" IsAssigned=
          "False" Action="Create" IsHierarchical="True">
        <!--Please do not modify 'Identifier' as these are system generated.
        If your intention is to change name of the instance Do it by changing Parameter $Name
        value below-->
        <Location>Root\Folder_1</Location>
        <Parameters>
          <Parameter Identifier="$Name" Value="DISignal_UL_1" />
          <Parameter Identifier="$Description" />
          <Parameter Identifier="$Area" />
          <Category Name="Configuration">
            <Parameter Identifier="Negate" Value="False" />
            <Parameter Identifier="Type" Value="False" />
            <Parameter Identifier="DISignalName" />
            <Parameter Identifier="TimeStamping" Value="None" />
            <Parameter Identifier="PLCHMIVar" Value="False" />
          </Category>
        </Parameters>
        <ElementSelections />
        <Elements />
        <Interfaces>
          <Interface Identifier="BoolVar" />
          <Interface Identifier="DINPUTSignalExt" />
          <Interface Identifier="DIChannel" />
          <Interface Identifier="DISignal" />
          <Interface Identifier="DINPUTSignal" />
          <Interface Identifier="TSDISignalVar" />
          <Interface Identifier="NullInterfaceRoleA" />
          <Interface Identifier="NullInterfaceRoleB" />
        </Interfaces>
      </ApplicationInstance>
    </ApplicationInstances>
  </ApplicationFolder>
</Export>

```

Item	Description
1	Data pertaining to the folder.
2	Data pertaining to the instance.
3	Data of the parameters and elements of the instance.
4	Data of the application interfaces, page 238 that the instance exposes.
5	Folder identifier.
6	Folder <i>Action</i> parameter.
7	\$InstanceID parameter of the instance, which is generated by the software. This is the identifier that the software uses to identify the instance for delete and update actions. Do not edit this parameter.
8	\$Name parameter of the instance.
9	Instance <i>Action</i> parameter.

When instances that are linked are exported, the information of the interface links is grouped in the *InterfaceLinks* element, which appears at the end of the file. The following example shows an extract of an export file, which contains two instances linked by two interface links.

```

<InterfaceLinks Type="SchneiderElectric.ProcessExpert.Application.Source.Model.InterfaceLink">
  <InterfaceLink Action="Create" DestinationPath="Root\Folder_1" DestinationInstanceName="DISignal_UL_1"
    DestinationInstanceTemplateIdentifier="$DISignal_UL" DestinationInstanceIdentifier="DISignal_UL_1"
    DestinationInterfacePath="" DestinationInterfaceIdentifier="DISignal"
    IsDestinationInstanceHierarchical="True" SourcePath="Root\Folder_1" SourceInstanceName="Motor_1"
    SourceInstanceTemplateIdentifier="$Motor" SourceInstanceIdentifier="Motor_1" SourceInterfacePath=
    "Control\Motor\Logic" SourceInterfaceIdentifier="RSP" IsSourceInstanceHierarchical="True" />
  <InterfaceLink Action="Create" DestinationPath="Root\Folder_1" DestinationInstanceName="DISignal_UL_1"
    DestinationInstanceTemplateIdentifier="$DISignal_UL" DestinationInstanceIdentifier="DISignal_UL_1"
    DestinationInterfacePath="" DestinationInterfaceIdentifier="NullInterfaceRoleA"
    IsDestinationInstanceHierarchical="True" SourcePath="Root\Folder_1" SourceInstanceName="Motor_1"
    SourceInstanceTemplateIdentifier="$Motor" SourceInstanceIdentifier="Motor_1" SourceInterfacePath=""
    SourceInterfaceIdentifier="NullInterfaceRoleB" IsSourceInstanceHierarchical="True" />
</InterfaceLinks>

```

Item	Description
1	Action parameter plus source and destination information pertaining to the first interface link.
2	Same information pertaining to the second interface link.

Editing the XML Application Export File to Manage Folders and Instances

Overview

You can edit the XML export file by using any XML editing software to perform one or more of the following actions on application folders and/or instances:

- Create an object by using the parameter values of the export file.
- Update an object that has the same identifier by using the parameter values in the export file.
- Delete an object that has the same identifier as in the export file.

You can also create and delete interface links, page 243.

The software applies the changes when you import the edited XML file, page 246 into the application of the system. At that time, you have the possibility to review and modify the list of imported objects (except interface links).

NOTE: You cannot update the identifier of a folder through import.

Good Practices

- Before editing an application export file, create a copy of it. If needed, it lets you restore the original application.
- Edit only an up-to-date export file to help avoid data conflicts when importing.

Application Export File Content

The table shows the objects and their respective parameters that are exported to the XML file and that are configurable.

Object	Parameters
Application folder	For the folder and any subfolders
	<i>Action</i>
	<i>Identifier</i>
	<i>Alias</i>
	<i>Description</i>
	<i>Area⁽¹⁾</i>
	<i>Location</i>
	<i>Parameters</i>
Instance	For the instances in the folder and its subfolders
	<i>Action</i>
	<i>TemplateIdentifier</i>
	<i>Version</i>
	<i>IsHierarchical</i>
	<i>Location</i>
	<i>\$Name</i>
	<i>\$Description</i>
	<i>\$Area⁽¹⁾</i>
	Elements that are enabled (selected in the Instance Editor)
Interface link	Parameters of each element For parameters of the enumerated data type, page 240, the name is exported as value. For example, if the parameter value, page 176 that is configured in the Instance Editor is <i>Operator</i> (2), the exported value is <i>Operator</i> . To edit the value, you can enter either the name of the enumerator or the associated integer (for example, 3 for an enumerator that is <i>Supervisor</i> (3)).
	Identifier of exposed interfaces and their path in the composition of the instance. Includes data of disabled and connected interfaces.
	When an alias is configured for an interface, both the alias and the identifier are indicated. The alias is the name that appears in the Asset Workspace Editor or Links Editor .
Interface link	Parameters of interface links that exist between exported source and destination instances. Refer to the topic describing how to edit the XML file to manage interface links, page 243.
(1) If the value is inherited from a parent folder, page 167, it does not appear in the export file.	

Action Parameter

Configuring the *Action* parameter is optional.

Possible values (not case-sensitive):

- Create (default)
- Update (except for interface links, page 243)
- Delete
- Blank:
 - Creates the object if it does not exist yet.
 - Updates the object if it exists (except interface links).

NOTE: Instances with action set to create and that you do not want to import can be removed from the list of imported objects during import, page 250.

Using the Invariant Culture

Use the invariant culture, page 228 to modify or add parameter values in the application export file.

Editing Rules for XML Application Export Files

To maintain the consistency of the folder structure and instances contained in the export file, follow these rules.

Item	Rule
Instance	Do not modify Identifier of the instance (<code>\$InstanceID</code> parameter) because it is generated by the software.
Parameters	Modify only the parameters described in the procedures that appear in this topic.
Object hierarchy	<p>Do not modify the structure of child elements.</p> <p>For example, a folder must be the parent of an instance and not the opposite. However, you can modify the order of <i>Elements</i> or <i>Parameters</i> inside an instance (<i>Application/Instance</i> section).</p>
	<p>When the export file contains data of a folder structure, do not change the <code><Location></code> parameter of the highest level folder (the folder from which the export file was created). For subfolders, the hierarchy from the export file prevails.</p>
	When the export file contains data of a single instance or single folder, do not modify the <code><Location></code> parameter.

The following figure shows an extract of an export file of instance *Motor_3*, which was exported from *Folder_5*. The instance is located in *Folder_1*, a subfolder of *Folder_5*. The *Location* parameters of these folders are highlighted.

```

<!--
<Alias />
<Description />
<Area />
-> <Location>Root\Folder_5</Location>
<GraphicalLocation>0</GraphicalLocation>
<Parameters />
<ApplicationInstances>
  - <ApplicationInstance Type="SchneiderElectric.ProcessExpert.Application
    Version="2.4.4" IsValid="True" Action="Create" IsHierarchical="True">
      <!-- Please do not modify 'Identifier' as these are system gener
      If your intention is to change name of the instance Do it by ch
-> <Location>Root\Folder_5\Folder_1</Location>
  - <Parameters>
    <Parameter Identifier="$Name" Value="Motor_3" />
    <Parameter Identifier="$Description" />
    <Parameter Identifier="$Area" />
  </Parameters>
</ApplicationInstances>

```

Creating an Instance

To edit the XML export file to create a new instance in an existing folder, proceed as follows.

Step	Action
1	<p>Change the value of the <code>\$Name</code> parameter to the name of the new instance that you want to create.</p> <p>NOTE: To create multiple instances during import:</p> <ul style="list-style-type: none"> Copy the entire <code>Application/Instance</code> section of the export file that contains the data of the instance. Paste it at the end of the section that you have copied, as many times as you want to create instances. Change the value of <code>\$Name</code> to the new name of each instance.
2	Verify that <code>Action</code> is set to <code>Create</code> ; otherwise replace the value by <code>Create</code> .
3	Save your changes.

Updating an Instance

To edit the XML export file format to update an existing instance, proceed as follows.

Step	Action
1	For the instance that you want to update, change the value of <code>Action</code> from <code>Create</code> to <code>Update</code> .
2	Change the value of the parameters, page 238 that you want to update.
3	Save your changes.

NOTE: If you update data of instances that are already linked and want the link to be maintained, it is not required to update the data of the **InterfaceLinks** element that pertains to the link as long as the linked interfaces remain enabled. This applies even if you change the value of `$Name` or the instance identifier (for example, by configuring an alias for a folder) for either or both instances.

Deleting an Instance And/Or a Folder

To edit the XML export file to delete an existing instance and/or folder, proceed as follows.

Step	Action
1	If you want to delete: <ul style="list-style-type: none"> The instance and the folder containing the instance, verify that the export file also includes the information pertaining to the folder. The instance only, proceed to step 3.
2	For the folder that you want to delete, change the value of <code>Action</code> from <code>Create</code> to <code>Delete</code> .
3	For the instance that you want to delete, change the value of <code>Action</code> from <code>Create</code> to <code>Delete</code> .
4	Save your changes.

Creating Folders

To edit the XML export file to create one or more folders or a folder hierarchy, proceed as follows.

Step	Action
1	Verify that <code>Action</code> is set to <code>Create</code> for the folders that you want to import; otherwise replace the value by <code>Create</code> .
2	Save your changes.

Updating a Folder

To edit the XML export file to update an existing folder, proceed as follows.

Step	Action
1	For the folder that you want to update, change the value of <i>Action</i> from <i>Create</i> to <i>Update</i> .
2	Change the value of any of the following folder parameters that you want to update: <ul style="list-style-type: none">• <i>Alias</i>• <i>Description</i>• <i>Area</i>• <i>Parameters</i> If the value of the parameter that you want to update is blank, enter the new value in the following format <code><Parameter>Value</Parameter></code> , where <i>Parameter</i> represents one of the four parameters that you can change and <i>Value</i> the new parameter value. For example, to modify the blank Area parameter of a folder to 100, enter <code><Area>100</Area></code> . NOTE: You must not change the Folder Identifier value.
3	Save your changes.

Editing the XML File to Manage Interface Links

Overview

You can use any XML editing software to edit the export file. This lets you create and delete interface links.

The software applies the configuration when you import the edited XML file, page 246 into the application of the system.

If you modify the data of a link that is contained in the export file, the existing link is not deleted after import. The new link is created in addition.

Good Practices

- Before editing an application export file, create a copy of it. If needed, it lets you restore the original application.
- Edit only an up-to-date export file to help avoid data conflicts when importing.
- Use the data of exposed interfaces, page 236 to create interface links.

Linked Instances Data

For any actions on links, the data of the linked instances must appear in the application export file, page 238 even if the instances already exist in the target application.

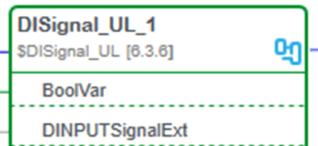
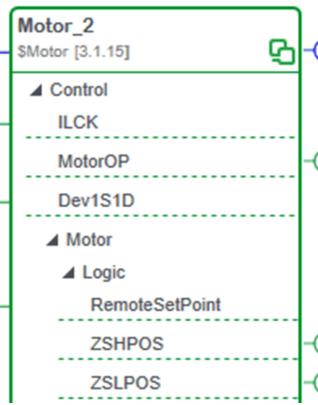
If the linked instances already exist in the target application with the same configuration and you do not want to modify them, change the value of `$Action` to `Update` for each instance.

NOTE: If the instances already exist in the target application and you leave the value as `Create`, the **Resolve Uniqueness Conflicts** dialog box, page 252 opens during import and you need to select the `Update` action for the instances there.

Contents of the *InterfaceLinks* Element

The table describes the parameters of the *InterfaceLinks* element, page 236 that appears at the end of the XML application export file.

Parameter	Description
<code>Action</code>	Action that the software performs by using the interface link data. Possible values: <ul style="list-style-type: none"> • Create (default): Creates new links in addition to existing links whose data has been exported. • Delete: Deletes links whose configuration matches the data in the imported file. NOTE: To create a new link in place of an existing link, you need to explicitly delete the existing link. NOTE: Entering <code>Update</code> produces the same result as <code>Create</code> .
<code>DestinationPath</code>	Path to the instance that is the destination of the link in the Application Explorer . The name of the system root folder, page 155 is replaced by <code>Root</code> .
<code>DestinationInstanceName</code>	Value of the <code>\$Name</code> parameter, page 184 of the instance that is the destination of the link.
<code>DestinationInstanceTemplateIdentifier</code>	Identifier of the template that is used by the instance, which is the destination of the link.
<code>DestinationInstanceIdentifier</code>	Identifier, page 171 of the instance that is the destination of the link.
<code>DestinationInterfacePath</code>	Path to the destination interface in the composition of the instance.

Parameter	Description
	<p>Example:</p> <p>In the following sample instance, value for interface <code>BoolVar</code> is "" (blank) because it has no parent element.</p> 
<code>DestinationInterfaceIdentifier</code>	Identifier of the interface that is the destination of the link. 
<code>IsDestinationInstanceHierarchical</code>	Indicates if the hierarchical naming function, page 183 is enabled for the instance.
<code>SourcePath</code>	<p>Path to the instance that is the source of the link in the Application Explorer.</p> <p>The name of the system root folder, page 155 is replaced by Root.</p> <p>Example:</p> <p>For instance <code>Motor_1</code> of <code>System_1</code> that is located in <code>Folder_2</code>, which is a subfolder of <code>Folder_1</code>, the value is <code>Root\Folder_1\Folder_2</code>.</p>
<code>SourceInstanceName</code>	Value of the <code>\$Name</code> parameter, page 184 of the instance that is the source of the link.
<code>SourceInstanceTemplateIdentifier</code>	Identifier of the template that is used by the instance, which is the source of the link.
<code>SourceInstanceId</code>	Identifier, page 171 of the instance that is the source of the link.
<code>SourceInterfacePath</code>	<p>Path to the source interface in the composition of the instance.</p> <p>Example:</p> <p>In the following sample instance, the value for interface <code>ZSHPOS</code> is <code>Control\Motor\Logic</code>.</p> 
<code>SourceInterfaceIdentifier</code>	Identifier of the interface that is the source of the link. 
<code>IsSourceInstanceHierarchical</code>	Indicates if the hierarchical naming function, page 183 is enabled for the instance.

Creating a Link

To edit the XML export file to create a link, proceed as follows.

Step	Action
1	In the <i>InterfaceLinks</i> element, copy the source and destination data that pertains to an existing interface link and paste it inside the element.
2	Edit the necessary source and destination data.
3	Verify that the value of <i>Action</i> is <i>Create</i> .
4	Save your changes.

Deleting a Link

To edit the XML export file to delete a link, proceed as follows.

Step	Action
1	In the <i>InterfaceLinks</i> element, locate the source and destination data that pertains to the interface link and change the value of <i>Action</i> to <i>Delete</i> .
2	Save your changes.

Importing a CSV or XML File Into the Application

Overview

By using the **Import** command, you can modify the application with the configurations that you have made in the CSV or XML file.

The import process is performed in steps during which the software:

1. Validates the structure of the CSV or XML file.
2. If valid, displays the folders, asset workspaces, and instances contained in the imported file along with the action that you have configured for each one, informs you of detected issues, and lets you select which objects to import.
3. Imports selected objects that satisfy applicable rules (for example, uniqueness of instance identifiers and presence of template).
4. For the other objects, informs you of the detected uniqueness conflicts, page 252 and lets you resolve them so that you can import these objects.
5. Displays a summary of the import operation, which includes information about incorrect configurations and objects that were not imported.

Creating and deleting instances through import affects the instance count, page 174.

NOTE: Verify that the version of the templates that are used by the instances that you are importing exists in the Global Templates library before you proceed with the import; otherwise you will not be able to import these instances. Refer to the topics describing how to:

- *Browse application templates*, page 158.
- *Export/import templates*, page 881.

Software Version Compatibility

You can import into a system instances that were exported to XML format from Process Expert 4.0 and later.

If the XML export file was created from a version earlier than Process Expert 4.0, you need to import the file into the version from which it was created, migrate the database in steps to version 4.0, and export the instances again from this version.

Instance Import Behavior

The table describes the default import behavior when you configure the export file to *create* instances in the following scenarios.

Scenario	Result
An instance parameter value is invalid or no value is configured.	The instance is created and configured with parameters whose value is valid. For the invalid or not configured parameter, the default value of the template is used.
An instance element selection value is invalid or no value is configured.	The instance is created and configured with element selections whose value is valid. For the invalid or not configured element selection, the value is set to false. The values that are configured for its parameters are not applied.

Folders	Instance	Result
No <i>Folders</i> group, page 223 or no data, page 228 in the group.	Data is valid.	<ul style="list-style-type: none"> If the folders exist in the target application: The instance is created. If at least one folder in the instance path does not exist: The instance is not created.
Invalid data for at least one folder in the instance path.	Data is valid.	<ul style="list-style-type: none"> If the folders exist in the target application: The instance is created. If the folders do not exist, only those that have valid data and a parent folder with valid data are created. <p>The instance is created only if the folder that contains it is created.</p>

The table describes the import behavior when you configure the export file to *update* instances in the following scenarios.

Scenario	Result
An instance parameter value is invalid or no value is configured.	<p>Only parameters whose value is valid are updated.</p> <p>For the invalid or not configured parameter, the current value of the instance is retained.</p>
An instance element selection value is invalid or no value is configured.	<p>Only element selections whose value is valid are updated.</p> <p>For the invalid or not configured element selection, the current value of the instance is retained.</p>

Folders	Instance	Result
No <i>Folders</i> group, page 223 or no data, page 228 in the group.	Data is valid.	The instance is updated.
Invalid or missing data for at least one folder in instance path.	Data is valid.	<p>NOTE: When you import an XML file, folders whose data is invalid or missing are tagged with a red dot in the Import window.</p>

Folder Import Behavior

The table describes the default import behavior when you configure the export file to *create* folders in the following scenarios.

Scenario	Result
The parameter <i>Action</i> for all the folders in the export file is set to <i>Create</i> and the parent folder already exists in the target application.	<p>Only its subfolders are created if these do not exist yet.</p> <p>If a subfolder already exists, it is not created but any lower-level folder in the hierarchy is created if it does not exist yet. And so on.</p>
The data of at least one folder in the folder structure is invalid.	Only folders that have valid data and a parent folder with valid data are created.

The table describes the import behavior when you configure the export file to *update* folders in the following scenario.

Scenario	Result
The data of at least one folder in the folder structure is invalid.	Only folders that have valid data are updated.

NOTE: If it is not able to propagate the value of the **Area** parameter to any object because the object is locked, the software:

- Displays a notification to inform you which object is locked.
- Rolls back any propagated area value.
- Reverts the **Area** parameter value that you have entered in the folder properties to its previous value.

NOTICE

NO CHANGE PROPAGATION

When the **Area** parameter value of a folder is not propagated:

- Reenter the **Area** parameter value in the properties of the folder.
- Verify in the **Notification Panel** that parameter value propagation completed successfully.

Failure to follow these instructions can result in alarms being signaled in the wrong area during operation.

The table describes the import behavior when you configure the export file to delete folders in the following scenario.

Scenario	Result
The folder that you are deleting contains subfolders and instances.	The software deletes the subfolders and instances.

Interface Link Import Behavior

When you import two instances that are linked and select the **Relative Path** import setting, page 250, which imports the instances to a location that is different from the one that is specified for the source and destination instance in the interface links data group, the link is maintained.

Aborting Import Tasks

After confirming the import by clicking **OK** in the **Import** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task.

Only the user who has selected the command is allowed to abort it.

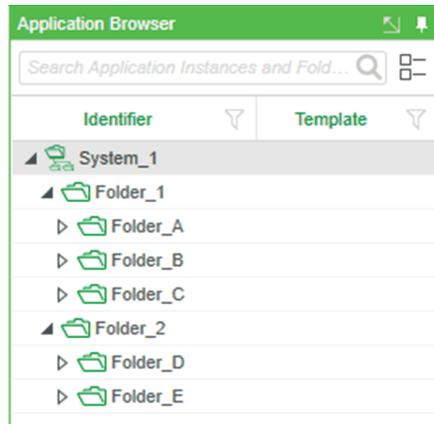
Folders and instances that already appear in the **Application Browser** at the moment you abort the task are retained given they belong to a tree structure that starts from the system root folder and that is completely imported at the moment the import task is aborted.

If you click the abort icon after clicking **OK** in the **Resolve Uniqueness Conflicts** dialog box, instances and folders that you had selected to be created and/or updated are not imported. Already imported objects (non-conflicting ones) are retained.

In any case, links that exist between instances are not recreated when you abort an import task even if these instances are retained. This is because recreating links is the last step performed before completing an import task.

NOTE: If you are importing only a few objects, the task may complete before you are able to click the abort icon.

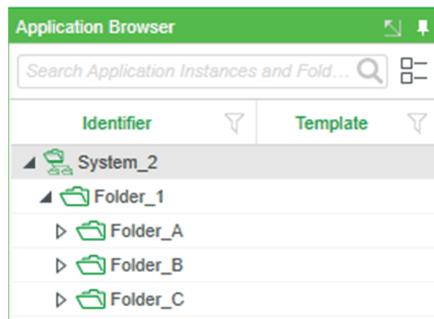
The following example illustrates two scenarios. Parent folders *Folder_1* and *Folder_2* are exported with their respective subfolders from the system root folder (*System_1*).



When you import the two structures into a system, folders are imported in sequence, structure after structure. If the import task is aborted:

- Before *Folder_C* and its contents is completely imported (displayed), nothing is retained.
- After *Folder_C* and its contents is completely imported (displayed), only the complete *Folder_1* tree structure and the instances therein are retained.

The following figure shows the objects that are retained when the import task is aborted after *Folder_C* and its contents has finished importing.



If a link exists, for example, between an instance in *Folder_A* and an instance in *Folder_C*, it is not recreated.

Opening the Application Import Window

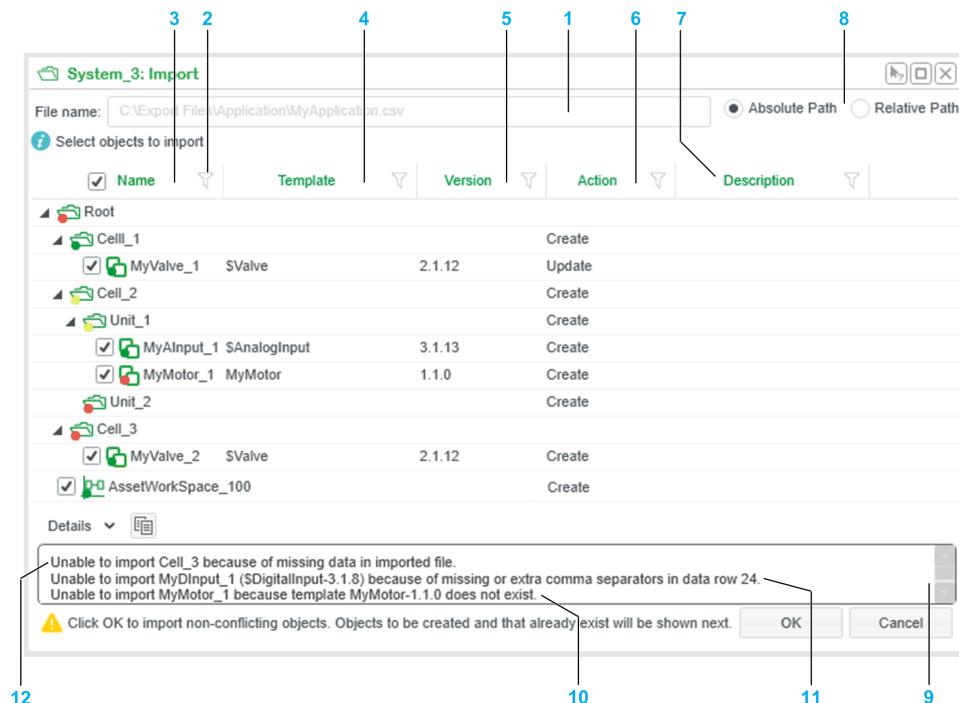
To open the application **Import** window, proceed as follows.

Step	Action
1	In the Application Browser , right-click the folder in which you want to import instances or that is to become the parent folder and select Import . Result: The Open dialog box opens.
2	Select the application export file and click Open . Result: The application Import window opens and displays the contents of the file.

NOTE: If you are importing only asset workspaces, you can right-click independently the root node in the **Application Browser** or **Asset Workspace** pane.

Import Window Description

The following figure shows an example of the application **Import** window. This example illustrates a scenario where three instances and one folder cannot be imported. Details are provided in the legend.



Item	Description
1	Full path of the file that you are importing.
2	You can sort and filter, page 118 data.

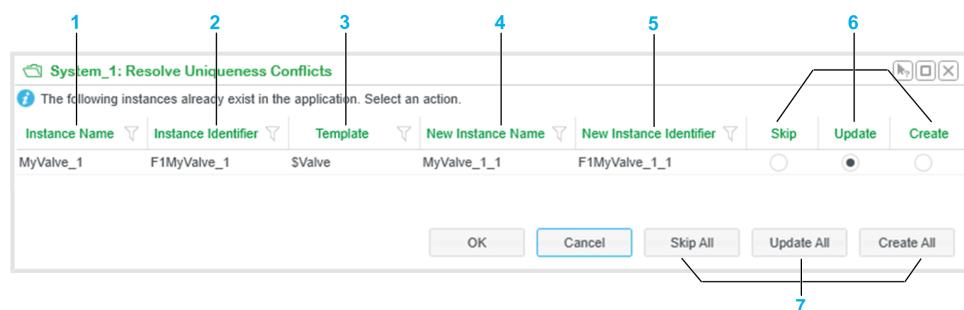
Item	Description										
3	<p>Identifier of the folders and asset workspaces and name (<code>\$Name</code>) of the instances contained in the imported file.</p> <p>A selected check box indicates that the object is imported if possible.</p> <p>A colored dot next to each object indicates the import status. Details are shown in a tooltip.</p> <p>NOTE:</p> <ul style="list-style-type: none"> Instances whose data is invalid are not shown. Instances whose data is valid are shown but may not be created, page 246. <p>Refer to the Details section.</p>										
	<table border="1"> <thead> <tr> <th>Color</th><th>Colored dot description</th></tr> </thead> <tbody> <tr> <td>Green</td><td> <p>For instances: The template that is used by the instance (same identifier and version) is available in the Global Templates library.</p> <p>For folders: Subfolders and instances inside the folder and its subfolders are marked with a green dot.</p> </td></tr> <tr> <td>Red</td><td> <p>For instances: The template that is used by the instance (same identifier and version) is not available in the Global Templates library. If you proceed, the instance is not imported.</p> <p>For folders, may appear in these situations:</p> <ul style="list-style-type: none"> Subfolders and instances inside the folder and its subfolders are marked with a red dot. The folder data in the CSV file is invalid. The folder contains an instance whose data is invalid. <p>If the location of the instance is not specified, the red dot appears on the system root folder.</p> </td></tr> <tr> <td></td><td>For asset workspaces, appears if the asset workspace identifier in the CSV file is invalid.</td></tr> <tr> <td>Yellow</td><td>For folders only: The folder or its subfolders contains at least one subfolder or instance that are marked with a red dot while at least one instance or subfolder is marked with a green dot.</td></tr> </tbody> </table>	Color	Colored dot description	Green	<p>For instances: The template that is used by the instance (same identifier and version) is available in the Global Templates library.</p> <p>For folders: Subfolders and instances inside the folder and its subfolders are marked with a green dot.</p>	Red	<p>For instances: The template that is used by the instance (same identifier and version) is not available in the Global Templates library. If you proceed, the instance is not imported.</p> <p>For folders, may appear in these situations:</p> <ul style="list-style-type: none"> Subfolders and instances inside the folder and its subfolders are marked with a red dot. The folder data in the CSV file is invalid. The folder contains an instance whose data is invalid. <p>If the location of the instance is not specified, the red dot appears on the system root folder.</p>		For asset workspaces, appears if the asset workspace identifier in the CSV file is invalid.	Yellow	For folders only: The folder or its subfolders contains at least one subfolder or instance that are marked with a red dot while at least one instance or subfolder is marked with a green dot.
Color	Colored dot description										
Green	<p>For instances: The template that is used by the instance (same identifier and version) is available in the Global Templates library.</p> <p>For folders: Subfolders and instances inside the folder and its subfolders are marked with a green dot.</p>										
Red	<p>For instances: The template that is used by the instance (same identifier and version) is not available in the Global Templates library. If you proceed, the instance is not imported.</p> <p>For folders, may appear in these situations:</p> <ul style="list-style-type: none"> Subfolders and instances inside the folder and its subfolders are marked with a red dot. The folder data in the CSV file is invalid. The folder contains an instance whose data is invalid. <p>If the location of the instance is not specified, the red dot appears on the system root folder.</p>										
	For asset workspaces, appears if the asset workspace identifier in the CSV file is invalid.										
Yellow	For folders only: The folder or its subfolders contains at least one subfolder or instance that are marked with a red dot while at least one instance or subfolder is marked with a green dot.										
4	Identifier of the template that instance is using.										
5	Version of the template that the instance is using.										
6	<p>Indicates the action that the software performs on the selected object according to the value of the <i>Action</i> parameter in the imported file.</p> <p>NOTE: When you create an instance, in the import summary that is shown in the notification panel, the action appears as <i>Create</i> for the instance but as <i>Update</i> for its elements and parameters.</p>										
7	Contents of the <code>\$Description</code> parameter of the object.										
8	<p>Import setting. The view is updated to show the impact of your selection.</p> <table border="1"> <tr> <td>Absolute Path</td><td>During import, the software maintains the folder structure of the imported file independently from where you select the import command given the required folder data is present in the imported file or the folders exist in the target application. The system root folder appears as Root.</td></tr> <tr> <td>Relative Path</td><td>During import, the folder from which you select the import command becomes the parent folder of the imported folder structure. However, folders that do not exist in the target application and whose data is missing in the import file are not imported. This may change the location of instances. NOTE: To be able to import an asset workspace when you are importing at the folder level, you need to select Absolute Path.</td></tr> </table>	Absolute Path	During import, the software maintains the folder structure of the imported file independently from where you select the import command given the required folder data is present in the imported file or the folders exist in the target application. The system root folder appears as Root .	Relative Path	During import, the folder from which you select the import command becomes the parent folder of the imported folder structure. However, folders that do not exist in the target application and whose data is missing in the import file are not imported. This may change the location of instances. NOTE: To be able to import an asset workspace when you are importing at the folder level, you need to select Absolute Path .						
Absolute Path	During import, the software maintains the folder structure of the imported file independently from where you select the import command given the required folder data is present in the imported file or the folders exist in the target application. The system root folder appears as Root .										
Relative Path	During import, the folder from which you select the import command becomes the parent folder of the imported folder structure. However, folders that do not exist in the target application and whose data is missing in the import file are not imported. This may change the location of instances. NOTE: To be able to import an asset workspace when you are importing at the folder level, you need to select Absolute Path .										
9	Notification area, which contains information about detected issues.										
10	The message indicates that the instance cannot be imported because its template is not available in the Global Templates library. As a result, the instance is tagged with a red dot.										

Item	Description
11	The message indicates that the instance that is located in folder <i>Unit_2</i> cannot be imported. As a result, the folder is tagged with a red dot and the instance is not shown in the window.
12	Folder <i>Cell_3</i> cannot be imported because of missing data in the imported file. As a result, the folder is tagged with a red dot and no action appears for it. Also, instance <i>MyValve_2</i> will not be created. It is tagged with a green dot because its template is available in the Global Templates library.

Resolving Conflicts During Import

You can resolve the object-uniqueness conflicts that the software detects during import, for example, if you are importing an instance or asset workspace and one with the same name or identifier, page 184 already exists in the application.

The **Resolve Uniqueness Conflicts** dialog box opens once non-conflicting objects have been imported. The following figure shows an example for an instance.



Item	Description
1	\$Name parameter of the instance as it appears in the imported file. The column can also show the identifier of an asset workspace.
2	Identifier of the instance based on the data of the imported file.
3	Identifier of the template that the instance uses.
4	New value for the \$Name parameter of the instance that the software will use to satisfy applicable naming rules if you select to create the instance. The column can also show the new identifier of an asset workspace.
5	Identifier of the object after import if you select to create it.
6	Buttons that let you select an action for individual objects: <ul style="list-style-type: none"> Skip: The existing instance and its links are not modified or the asset workspace is not imported. Update: Default action. Modifies the existing object with the imported parameter and link configuration. The name of the object is not modified. Create: Creates a new object with New Instance Name and the imported folder, parameter, and link configuration. NOTE: Update is disabled if you have selected the Relative Path import setting and, as a result, the instance is imported in a different folder.
7	Buttons that let you select an action for the conflicting objects as a whole. The actions are the same as when you select the action for objects individually. NOTE: Clicking these buttons overrides the selection you have made for individual objects.

Importing Objects into the Application

To import the content of an application export file into the application of a system, proceed as follows.

Step	Action
1	In the application Import window: <ul style="list-style-type: none"> Review information that is shown in the Details pane (if applicable). Select the objects that you want to import. Select the import settings that you want to apply.
2	Click OK . <p>Result: If you select to import with relative path, a confirmation dialog box opens.</p> <p>NOTE: Click Cancel to close the Import window without importing any object.</p>
3	Click OK . <p>Result: If it does not detect conflicting objects, the software: <ul style="list-style-type: none"> Imports the contents of the file and applies the import settings that you have selected. Displays the imported objects in the Application Browser. Displays a summary of the import process in the notification pane. If it detects conflicts, the software: <ul style="list-style-type: none"> Imports the objects that satisfy import rules and applies the import settings that you have selected. For conflicting objects, opens the Resolve Uniqueness Conflicts dialog box; proceed to step 4. </p>
4	In the Resolve Uniqueness Conflicts dialog box, select the actions that you want to perform for conflicting objects. <p>NOTE: Click Cancel to close the Resolve Uniqueness Conflicts dialog box without importing objects for which the software has detected a conflict. Objects that satisfied import rules remain imported.</p>
5	Click OK . <p>Result: The software: <ul style="list-style-type: none"> Imports the remainder of the objects with the import settings that you have selected. Displays the entirety of the imported objects in the Application Browser. Displays a summary of the entire import process in the notification pane. </p>

Project Definition Stage

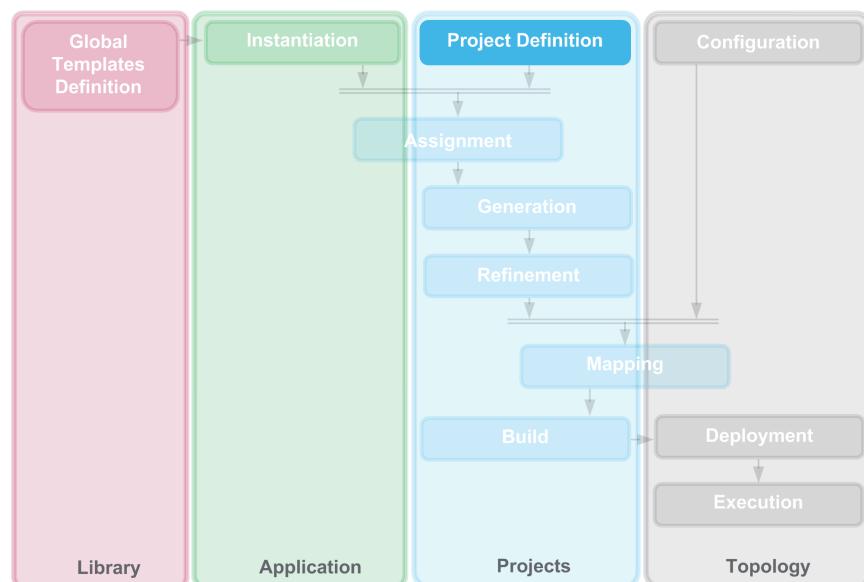
What's in This Chapter

Project Explorer and Project Browser.....	255
Control Project Definition Stage.....	259
Managing Control Projects	265
Supervision Project Definition Stage	288
Creating Supervision Project Components	294
Managing Supervision Projects	306
Participant Project Indicators.....	321

Overview

This chapter contains information to help you define the Control and Supervision projects of the system.

The following figure shows the position of the **Project Definition** stage within the system engineering life cycle.



Refer to the Project Definition stage, page 41 for a description of the purpose of this stage.

Project Explorer and Project Browser

Overview

This section describes the **Project Explorer**, which features the various editors that let you manage Control and Supervision Participant project components. It also describes the **Control Project Browser** and **Supervision Project Browser**, which let you create these projects and open their respective editors.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Project Explorer

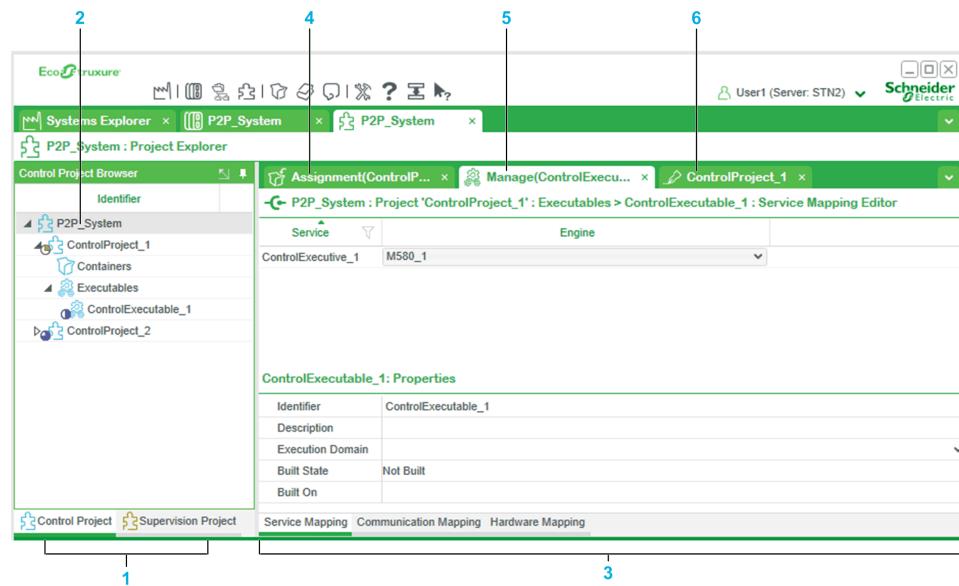
Opening the Project Explorer

To open the **Project Explorer** for a system, verify that the system is selected in the **Systems Explorer** or that at least one explorer/editor is already open for this system. Then, click the **Open Project Explorer** button in the engineering client toolbar.

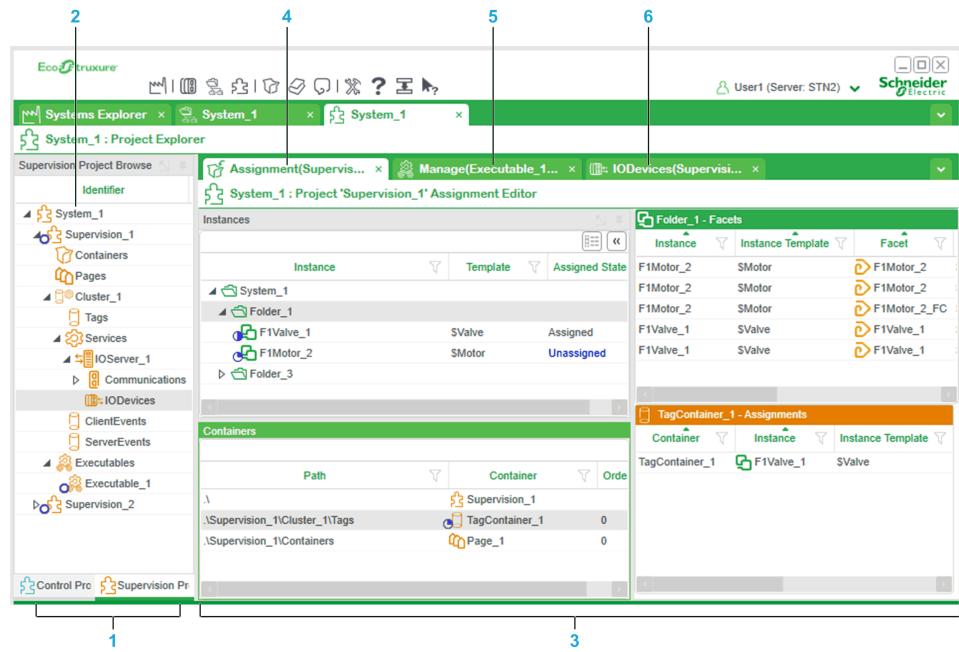
Otherwise, right-click the system in the **Systems Explorer** and select **Open Project**.

Project Explorer Window

The following figure shows an example of the **Project Explorer** for a system in which two Participant projects exist. The **Control Project Browser** is selected.



The following figure shows an example of the **Project Explorer** for a system in which two Participant projects exist. The **Supervision Project Browser** is selected.



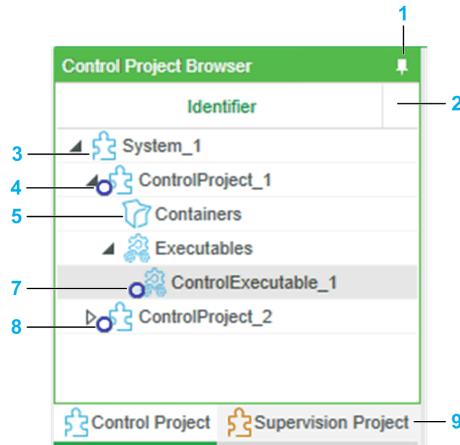
Item	Description
1	Tabs to show the Control Project Browser or Supervision Project Browser pane, page 258.
2	System root folder.
3	Remaining work area, page 105 of the Project Explorer when the project browser pane is pinned. The data that is shown depends on the project browser tab that is selected. The work area is empty by default.
4	Assignment Editor tab, page 329. It shows the containers of the Participant projects and lets you assign facets to them.
5	Executable management, page 577 tab, which lets you view and edit executable properties and perform the various mappings by selecting the corresponding subtabs: <ul style="list-style-type: none">• Service Mapping• Communication Mapping• Hardware Mapping (Control only)
6	Properties tab, page 261 for the Participant projects and their components other than executables.

Project Browser

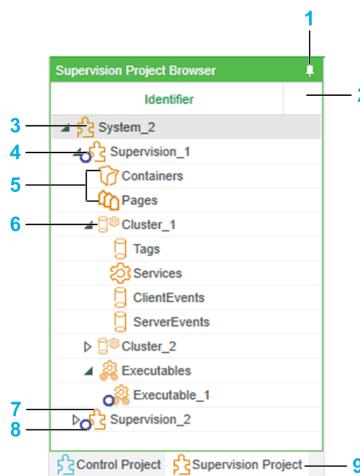
Description

The project browsers let you see, create, and interact with the Control and Supervision Participant projects that exist in a system.

The following figure shows an example of the **Control Project Browser** in which two projects have been created.



The following figure shows an example of the **Supervision Project Browser** in which one project has been created.



Item	Description
1	Pin button, page 111. Lets you hide the pane and restore it when it is docked.
2	You can customize, page 259 which columns appear.
3	System root folder.
4	Node giving access to project-level context menu commands.
5	Node giving access to container-level context menu commands.
6	Node giving access to cluster-specific context menu commands.
7	User-created executable, page 577.
8	Indicator icons, page 321.
9	Project type selection tabs.

Navigating Inside the Project Browser

The keys to navigate are the same as in the **Application Browser** pane, page 163.

Customizing Column Display

You can choose which columns you want to see by right-clicking a column header and selecting **Customize**. Certain columns are always shown.

For a description of the columns, refer to the topic describing properties of Control executables, page 579 and Supervision executables, page 620.

System Root Folder Actions

Right-click the system root folder to open a context menu with the following commands. The availability of commands depends on the tab of the Project Browser that is selected.

Command	Description
Create Control Project, page 260	Creates a new Control project that is associated to one of the supported controller families and displays it in the tree view of the Project Explorer .
Import Control Project, page 284	Opens an Open dialog box, which allows you to create a Control project by importing data of a project that was created with Control Expert. Export files in .stu or .xef format are supported.
Create Supervision Project, page 290	Creates a new Supervision project, and displays it in the tree view of the Project Explorer .
Open Application	Opens the Application Explorer , page 155, which allows you to start the instantiation stage or access the application of the system.
Open Topology	Opens the Topological Explorer , page 261, which allows you to start the configuration stage or access the topology of the system.
Export	Opens a Save As dialog box, which allows you to create an export file containing the information of the Control projects, page 267 and Supervision projects, page 307 that exist in the system.
Import	Opens an Import dialog box, which allows you to display the compatible contents of an export file (.sbk) in the Import window. You can import entire Control projects, page 271 and/or Supervision projects, page 311, or parts of them into the system.

Control Project Definition Stage

Overview

This section describes how to create Control projects of a system and their sections.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Creating Control Projects

Before Creating a Control Project

At this stage of the system engineering life cycle, you need to know:

- Which family of controllers, page 507 you will be using to run the Control project.
- How you will organize the function block diagram (FBD) sections of the project.

NOTE: For information on working with Control projects of M580 safety controllers, refer to the topic describing the engineering of M580 safety systems, page 759.

Creating Control Projects

To create a Control project, proceed as follows.

Step	Action
1	<p>In the Control Project Browser, right-click the system root folder, select Create Control Project and the controller platform, page 261 that is associated to the project.</p> <p>Result: The Control project is shown in the tree view.</p>

NOTE: In each system, you can create as many Control projects as needed. For example, you can create a project, which is engineered for a Modicon M580 controller in the plant and a project, which is engineered for an M340 controller for test purposes in the laboratory.

Control Project Actions

The table describes the Control project context menu commands.

Command	Description	Applies to
Assign Facets	Opens the Assignment Editor , which lets you create and manage sections and assign facets to them.	Assignment Stage, page 328
Generate , page 413	Starts a process to create or update the logic at the project level.	Generation stage
Refine	Opens the Control Participant, which allows you to make changes to the logic of the project.	Refinement stage, page 435
Manage Network Variables , page 459	Opens the Manage Network Variables dialog box, which allows creating, modifying, and deleting network variables. These are variables that are used to share data by using peer to peer communication.	
Manage Data Backup Files , page 286	Opens the Manage Data Backup Files dialog box, which lets you view and delete data backup files that exist for Control Participant project.	Project Definition stage
Manage Peer to Peer , page 462	<p>Opens the Peer to Peer Communication Configuration window, which lets you create and manage existing peer to peer communication mappings between controllers of a system. This functionality uses the specialized function blocks of the Control Participant.</p> <p>The command is available only if, for the executable of the other Control project, the service mapping has been performed.</p>	Refinement stage, page 435
ReGenerate , page 417	<p>Starts a process to recreate the logic from the beginning at the project level.</p> <p>NOTE: Certain changes that you have made during refinement may be discarded.</p>	Generation stage
Last Action Summary , page 327	Opens a dialog box that indicates the last action that was performed and a summary of the result.	Project Definition stage

Command	Description	Applies to
Export, page 267	Opens a Save As dialog box, which allows you to create an export file containing the Control project information.	
Settings	Opens the Settings window, page 263, which lets you configure various project-related settings.	Assignment stage
Delete, page 262	Deletes the Control project including: <ul style="list-style-type: none"> • Sections • Assignments • Executables, page 578 NOTE: For M580 safety projects, deletes also the associated M580 safety controller, page 769.	–
Rename	Allows you to enter a new identifier, page 261 for the project.	–
Properties	Opens the Properties window, which allows you to view and/or edit the properties of the Control project.	–

Containers Node Actions

The table describes the **Containers** node context menu commands.

Command	Description	Applies to
Assign Facets	Opens the Assignment Editor , which lets you create and manage sections and assign facets to them.	Assignment Stage, page 328
Create FBD Section	Lets you create an FBD section, page 335 in the task folder that you select. The Assignment Editor opens in the background.	
Generate	Starts a process to create or update the logic at the section level.	Generation Stage, page 383
ReGenerate	Starts a process to recreate the logic from the beginning at the section level. NOTE: Certain changes that you have made during refinement may be discarded.	

Executables Node Actions

The table describes the **Executables** node context menu commands.

Command	Description	Applies to
Create Executable	Creates one or more Control executables, page 578, which let you associate one or more controllers to the Control project, map project facets to hardware, and create communication channels by using the I/O scanner service of the controller.	Mapping Stage, page 577

Control Project Properties

Right-click the Control project and select **Properties**.

You can view and/or edit the following items.

Item		Description
General	Identifier	The identifier must be unique within the system.
	Description	Optional. You can enter a description of the project with free text.
Controller Family	Value	<p>Indicates the controller platform that you have associated to the project at the time of creation.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • M580 • M580 Safety • M340 • Quantum <p>Read-only.</p> <p>NOTE: When you associate a Control project with a controller family, the software limits the assignment of facets to the project to those that have the corresponding attribute (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) defined.</p>

Deleting a Control Project

When you select the command to delete a Control project, the software displays a confirmation dialog box. Once you have confirmed the deletion, the software executes the following actions in relation to the project that you are deleting:

- Deletes any components of the project and their contents, including any assignments and mappings.
- Deletes any associated files (for example, data backup files and expressions, page 350 for container naming rules).

NOTE: For M580 safety projects, deletes also the associated M580 safety controller, page 769.

Project Settings Window

The **Settings** window opens when you select the **Settings** command in the context menu of a Control or Supervision project.

Section	Description
Container Naming Rules	Lets you configure expressions, page 350, which define how sections are created and how Control facets are assigned to the project. The section appears only for Control projects.
Assignment	Lets you set the functionality, which can automatically assign remaining candidate facets, page 368 of instances. Possible values: <ul style="list-style-type: none"> • Enabled • Disabled
Project Analysis	Lets you enable or disable the automatic analysis of the Control project, which is performed when you generate the project/sections or save refinement changes. When the analysis is disabled: <ul style="list-style-type: none"> • The Control project validity indicators, page 324 are not functional. • The time required to generate and save refinement changes is reduced. You can perform an analysis of the Control project manually at any time by refining it offline and clicking Build > Analyze on the menu bar. After you click the save button, the project validity indicators indicate the status of the project. Possible values: <ul style="list-style-type: none"> • Enabled • Disabled: Default value. The section appears only for Control projects. NOTE: For M580 safety Control projects, the setting also impacts the configuration, page 769 of the associated M580 safety controller.
Generation	Automatically generates sections of the Control project that are impacted by changes to facets, page 394, which are being assigned or already assigned to them. Other sections whose facet assignment is not impacted need to be generated manually. The setting remains effective until you change it. Possible values: <ul style="list-style-type: none"> • Enabled • Disabled: Default value. You need to generate all sections of the Control project manually. NOTE: You can still use the Generate and Generate and Build commands while automatic generation is enabled.

NOTE: The **OK** and **Cancel** buttons apply to all the sections of the **Settings** window.

Creating Control Project Sections

Overview

In the **Project Explorer**, you can create only FBD sections in a Control Participant project by using context menu commands.

Creating FBD Sections

To create FBD sections in task folders, open the **Assignment Editor** by selecting **Assign Facets** from the context menu of the **Containers** node in the **Control Project Browser**.

For a description of sections and actions that you can perform, refer to the topic describing the **Assignment Editor**, page 330.

NOTE: You can create FBD sections also by refining the Control Participant project or importing them.

Creating Non-FBD Sections

You can create non-FBD sections by:

- Refining, page 436 the Control Participant project and creating the sections.
- Refining the Control Participant project online, page 727, creating the sections, and updating, page 748 the Control project.
- Importing, page 271 the sections.

Non-FBD sections are sections in any of the following languages: IL, LD, LL984, SFC, and ST. They appear also in the **Containers** pane of the **Assignment Editor**.

NOTE: In the following cases, existing non-FBD sections may be displayed only after you make a change by using the **Refine** command, page 436 at the project level and save it:

- The Control project is contained in a database of an earlier version of the software that you have migrated.
- The Control project is contained in a system that you have restored by using a backup file created with an earlier version, page 142 of the software.

Managing Control Projects

Overview

This section describes how to use the **Project Explorer** to manage entire Control projects or sections of such projects.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Control Project Section Order

Overview

The software assigns an order to sections, starting at 0, and displays it in the **Containers** pane, page 330. The values appear in the **Order** column.

It indicates the order in which the software generates the sections of the Control project and thereafter the execution order of the sections by the Control Participant.

You can change the order of sections inside the **Containers** pane.

The software applies changes to the order of sections in the logical Control Participant project when you perform either action:

- Select the **Refine** command at the project level (it is not required to make a change or save).
- Generate, page 417 the entire project or a section of the project.
- Regenerate, page 421 the entire project or a section of the project.

You then need to propagate the change downward in the system engineering life cycle.

NOTE: You can also change the order of sections by refining, page 436 the Control Participant project or by refining the deployed Control project online, page 727 and updating the Control Participant project.

Moving Sections

In the **Assignment Editor**, you can move sections only within the **MAST** or **FAST** task folder.

To move sections from the **MAST** to the **FAST** task folder or the other way around, refine the Control Participant project.

Changing the Order of Sections

To change the order of sections of a Control project, proceed as follows.

Step	Action
1	In the Containers pane of the Assignment Editor , clear filters, page 120.
2	Drag one section up or down to a new position. Result: A tooltip indicates the position that the section will occupy. NOTE: In the Assignment Editor , you can select multiple, contiguous or non-contiguous sections that belong to the same task folder.
3	Release the mouse button. Result: The section is inserted at the selected position and the values of the Order column are updated.
4	Apply the new order to the logical Control project by performing the necessary action.

NOTE: To change the order of the facets inside an FBD section, refer to *Changing the Facet Order*, page 381.

Exporting Control Projects

Overview

The export functionality lets you create a non-editable export file (.sbk), which contains selected Control project information.

You can use this data by using the Control project import, page 271 functionality to transfer project information to a different system or to merge information from different Control projects.

However, you cannot merge data related to communication mapping during import. You need to import either the complete communication channel, page 582 and/or peer to peer communication data, page 594 that is contained in the export file for selected Control projects or nothing.

NOTE: To export the Control and Supervision projects of a system at once in one export file, right-click the system root folder in either **Project Browser**, select **Export**, and follow the project export procedure described in this section. Refer also to the topic describing the export and import of Supervision projects, page 306.

Exporting M580 Safety Control Projects

For specific information about exporting M580 safety Control projects, refer to the topic describing how to engineer systems with M580 safety controllers, page 763.

Software Version Compatibility

The version of EcoStruxure Process Expert that you use to import the information contained in the export file must be the same as the version that you used to create the export file (a service pack is considered the same version); otherwise you cannot proceed with the import.

Project Related Data

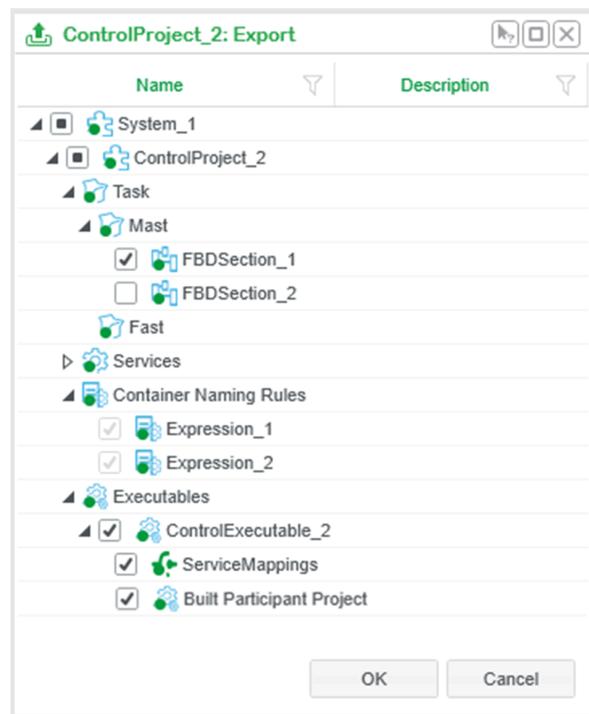
To be able to use the export file to import a project into a different system, verify that the instances, their templates, and topological entities related to the project exist in the target system before importing the project, page 272.

NOTE: The parameter values of instances that reference the exported facets are not contained in the export file. However, the export file contains information that allows the software to verify during import if the configuration of an instance is different from when its facets were exported.

Project Export Window

The Control window lets you select, which project information the export file will contain.

The following figure shows an example of the **Export** window where the check box for *FBDSection_2* has been cleared. To indicate that at least one item of the Control project is not selected, the check box of parent nodes are filled with a square.



The table indicates the meaning of the colored dots that appear in the **Export** window next to items of a Control project.

Color code	Description
Green	You can export the item.
Red	You cannot export the item because it contains at least one facet with the following state: <ul style="list-style-type: none"> Deleted Unassigned Out of Date Moved Generate sections that contain facets with such states in order to export the Control project. <p>NOTE: You cannot export items that are tagged with a red dot.</p>
Yellow	Indicates that the item contains at least one subitem that is tagged with a red dot. <p>The item can be exported partially only.</p>

NOTE: To expand/collapse a node in the **Export** window, select the node and press **Enter**.

Export Settings

The table describes which Control project information you can export.

Item	Selected for export	Description
System_n	By default	Unselecting the item excludes any Control project and disables the creation of the export file.
ControlProject_n	By default	Unselect the item to exclude the entire project from the export file.
Tasks folder	By default	You cannot unselect the item.

Item	Selected for export	Description
MAST and FAST task folders	By default	You cannot unselect the item.
Sections	By default	You can unselect sections individually. Selecting an FBD section automatically selects any facets that are assigned to it.
Safe Program	By default	The item is available only for M580 safety Control projects, page 759. You cannot unselect the item.
Container Naming Rules	By default	Represents the expressions, page 350 that are configured. You cannot unselect expressions.
Executables	By default	You cannot unselect the item.
ControlExecutable_n	By default	You can unselect each executable and its contents individually. Selecting an executable automatically selects any associated mapping.
Service Mappings	By default	Selecting the item exports any data related to the service mapping including any required contents in the contents repository. Unselecting the item also unselects both the associated hardware and communication mapping.
Hardware Mappings	By default	You cannot select hardware mappings individually. Selecting the item also selects the associated control executable and service mapping.
Communication Mappings	By default	You cannot select communication mappings individually. Selecting the item also selects the associated executable and exports the complete communication mapping data. NOTE: If the project contains peer to peer communication data, refer to the topic describing peer to peer communication export settings, page 269.
Built Participant Project	By default	Represents the built Control Participant file (.stu).
Backups	By default	You can select data backup files individually. The time stamp and description of each backup file, page 733 are shown. The format of the time stamp is YYYYMMDDHHMMSS.

Peer to Peer Communication Export Settings

To be able to recreate peer to peer communication channels and existing variable mappings at the time of import if the necessary conditions are fulfilled, you need to export all of the following:

- The owner project and its Control executable.
- The consumer project and its communication mappings.

NOTE: Including communication mappings in the export file may also include communication channels other than those for peer to peer communication (for example, communication channels for I/O devices).

Aborting Export Tasks

After confirming the export by clicking **Save** in the **Export** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The export file is not created.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are exporting a small project, the task may complete before you are able to click the icon.

Exporting a Control Project

To export a Control project to file, proceed as follows.

Step	Action
1	In the Control Project Browser , right-click the Control project that you want to export and select Export . Result: The Export window opens and displays the Control project that you have selected. NOTE: To be able to export two or more Control projects, for example, to export peer to peer communication data, right-click the system root folder.
2	Select the project components that you want to export. NOTE: If you had right-clicked the system root folder, you can select which projects you want to export.
3	Click OK . Result: The Export window closes and the Export dialog box opens.
4	Select a location where you want to save the export file, enter a file name, and click Save . Result: The software creates the export file containing the selected Control project information. NOTE: If you click Cancel , the Export dialog box closes without creating the export file and the selection you have made in the Export window is discarded.

Importing Control Projects

Overview

The **Import** command lets you import data of a Control project from an export file (.sbk), which was created by using the [export](#), page 267 functionality of the **Project Explorer**.

If the system into which you are importing a Control project already contains a Control project with the same identifier, the software compares the data of both projects. It then proceeds with the partial import, according to the import settings that you select and the import rules that are described in this topic.

However, you cannot do a partial import of data related to communication mapping, page 582. You need to import either the complete communication channel and/or peer to peer communication data, page 594 that is contained in the export file for selected Control projects or nothing. Importing it overrides any existing communication mapping and variable mappings. Specific conditions apply and are described in this topic.

If the software detects conflicts related to communication mapping before import, you have the choice to import the Control without communication mapping or to cancel the operation. If it detects conflicts during the import process, the entire import operation is canceled and rolled back.

NOTE: If the export file also contains information of a Supervision project, you can see the data that is related to the Supervision project, and have the choice to import it, page 311 as well.

Importing M580 Safety Control Projects

For specific information about importing M580 safety Control projects, refer to the topic describing how to engineer systems with M580 safety controllers, page 763.

Software Version Compatibility

The version of EcoStruxure Process Expert that you use to import the information contained in the export file must be the same as the version that you used to create the export file (a service pack is considered the same version); otherwise you cannot proceed with the import.

Naming Rules for Variable Names and Section Identifiers

Names of variables and identifiers of sections that you import into a system need to satisfy the naming rules, page 89 for the generation process to complete successfully.

Checking Consistency After Import

After importing a Control project, perform a consistency check. This allows the software to identify if the constituents of the project into which you have imported are not consistent with the facets that are assigned to the project.

For more information on the process, refer to the topic [Consistency Management](#), page 449.

Import Settings

The table describes which Control project information you can import into a system.

Item	Selected for import ⁽¹⁾	Description
System_n	By default	Unselecting the item excludes any Control project and disables the import of the export file.
ControlProject_n	By default	Unselect the item to exclude the entire project from the import process.
Tasks folder	By default	You cannot unselect the item.
MAST and FAST task folders	MAST and FAST task folders	By default You cannot unselect the item.
	Sections	By default You can select sections individually. Displays the identifier and path of the facets that are assigned to the section. A Constituent File is associated to each FBD section. Selecting an FBD section imports any facets that are assigned to this section. NOTE: A section can be selected only if it is tagged with a green dot, page 280.
	Constituent File (FBD sections only)	By default If the identifier of the imported FBD section already exists in the target project, the selection of the constituent file has an impact on the import behavior, page 274. NOTE: The Constituent File check box is selected and disabled if the identifier of the imported FBD section does not exist in the target project.
Safe Program	By default	The item is available only for M580 safety Control projects, page 759. Unselect the item to exclude the entire safety program from the import process.
Container Naming Rules	Project dependent	Expressions, page 350 are imported only if a Control project with the same identifier does not exist in the target system.
Executables	By default	You cannot unselect the item.
ControlExecutable_n	ControlExecutable_n	By default You can select each Control executable and its contents individually. Selecting the Control executable automatically selects any existing mapping.
	Service Mappings	By default Any data related to the service mapping including any required contents in the Contents Repository. Selecting the item automatically selects the associated executable.
	Hardware Mappings	By default You cannot select hardware mappings individually. Selecting the item automatically selects the associated executable and service mapping.
	Communication Mappings	By default The complete data related to the communication mapping. Selecting the item automatically selects the associated executable and service mapping.
	Built Participant Project	By default Represents the Control Participant file (.stu). It appears and is selected only if the built Participant project was selected at the time of export.
Backups	By default	You can select data backup files individually The time stamp and description of the backup file, page 733 are shown. The format of the time stamp is YYYYMMDDHHMMSS. Importing a data backup file makes it available in the Manage Data Backup File dialog box, page 286 and Deploy Data dialog box, page 677.
(1) The information needs to be contained in the export file to be selected.		

Import Rules for Project-Related Objects

Facets assigned to Control projects are related to instances of the application and mappings of project executables are related to entities of the topology. For the

Control project import to complete entirely, these project-related objects need to be present in the system in which you want to import the project.

The table describes the impact on the import operation when project-related objects are not present or have a different configuration. It also describes the corrective action.

Project-related object that is not present or different	Impact if not present in the target system	Required action
An instance whose facets are assigned to the Control project that you are importing (same identifier, template identifier, template version, and status Valid) is not present.	<p>You can import the project but not the sections to which such facets are assigned. This requires that the project contains at least one section that can be imported, page 280, even empty.</p> <p>A dialog box opens, page 281 asking you to confirm the partial import.</p> <p>NOTE: Performing a partial import may cancel the entire import operation if import rules for peer to peer communication variables, page 277 are not satisfied.</p>	<p>Do either of the following first to create missing instances:</p> <ul style="list-style-type: none"> Use the export/import functionality, page 219 of the Application Explorer to import missing instances. Create instances. This may require importing templates first. <p>You can obtain information about the missing instances and their templates in the tooltip of the container that is tagged with a red dot.</p>
An element of the instance creating a facet that is being imported is not selected.	<p>The facet that is normally created by this element is not imported.</p> <p>The notification panel provides information about the facet that is not imported.</p> <p>For example, you import a Control project to which the facet created by the Maintenance element of an instance is assigned but in the application of the target system, the Maintenance element of the same instance is not selected.</p>	Edit the instance in the Application Explorer and select the corresponding element.
The properties of an instance are different from the instance configuration at the time of export. The difference lies in the element selection (services) and/or the parameter values, page 176.	The assignment status of facets that are imported is set to Out Of Date .	Generate the project or section to apply the changes.
A topological entity to which you have mapped the executable of the project (service and/or communication) that you are importing (same identifier, same devices, and same services).	You can import the project but not the Control project executable and any of its associated mappings.	Use the export-import functionality, page 552 of the Topology Explorer first to create the missing topological entities.

Import Rules for Sections

The table describes the import rules that apply to FBD and non-FBD sections with respect to the section order when a Control project with the same identifier already exists in the target system.

Contents of export file selected for import	Section in Control project of target system	Result in Control project of target system after import
Section1, order 0 Section2, order 1 Section4, order 2	Section3, order 0 Section1, order 1	<p>Section3, order 0 Section1, order 1 Section2, order 2 Section4, order 3</p> <p>The order of existing sections is maintained and imported sections are added at the end.</p> <p>NOTE: If a section already exists in the target system, its order is maintained even if the order of the section in the export file is different.</p>

The table describes the import rules that apply to non-FBD sections when a Control project with the same identifier already exists in the target system.

Contents of export file selected for import	Section in Control project of target system	Result in Control project of target system after import
Section1	Section1	Section1 from export file replaces Section1 in the Control project of target of the target system.
Section1	Section2	Section1 and Section2 exist in the Control project of target of the target system.

Import Rules for Facets

The software applies import rules to facets if the system into which you are importing a Control project already contains a Control project with the same identifier.

The following are the general import rules that the software applies if facets with the same identifier contained in FBD sections with the same identifier exist in both the target project and the export file that you are importing:

- If you **select the constituent file** of the section:
 - Facets from the export file overwrite facets of the target system and maintain the state they have in the export file.
 - Facets that exist only in the export file are imported into the target system and maintain the state they have in the export file.
 - Facets that exist only in the target system become **Assigned** and **Non Generated**. If the status of the facet is deleted, unassigned, or unlinked, the software removes the facet from the section after import.
- If you **unselect the constituent file** of the section:
 - Facets with the same identifier maintain the state they have in the target system.
 - Facets that exist only in the export file are imported into the target system and become **Assigned** and **Non Generated** unless the generation status of the facet is **Unlinked**.
 - Facets that exist only in the target system maintain their state.

NOTE: The software changes the assignment status of any facet of an instance to **Out Of Date** if a property of an instance, page 176 is different from what it was at the time of export. The notification panel provides information about the change of the assignment status after import is complete.

The following tables illustrate these import rules based on the different use cases. The section and facet names are examples, and represent sections and facets in the **MAST** or **FAST** task folders of the projects.

If you select the FBD section **Constituent File** check box.

Contents of export file selected for import		Item in Control project of target system		Result in Control project of target system after import
Section identifier is identical				
Section1	Facet1	Section1	-	Facet1 from export file with status from export file.
Section1	Facet1	Section1	Facet1	Facet1 from export file with status from export file.
Section1	Facet1	Section1	Facet2	<p>Facet1 from export file with status from export file.</p> <p>Facet2 with status non-generated.</p> <p>The software adds imported facets at the end of existing facets in the target section.</p> <p>If the status of Facet2 is deleted, unassigned or unlinked, the facet is removed from the section after import.</p>
Section identifier does not exist in target project				

Contents of export file selected for import		Item in Control project of target system		Result in Control project of target system after import
Section2	Facet3	–	–	<p>Section2 with Facet3 from export file with status of export file.</p> <p>The software adds imported sections at the end of existing sections in the target task folder.</p>
– The item does not exist				

If you unselect the FBD section **Constituent File** check box.

Contents of export file selected for import		Item in Control project of target system		Result in Control project of target system after import
Section identifier is identical				
Section1	Facet1	Section1	–	Facet1 from export file with status non-generated unless generation status of Facet1 is Unlinked .
Section1	Facet1	Section1	Facet1	Facet1 from target system with same status.
Section1	Facet1	Section1	Facet2	<p>Facet1 from target system with status non-generated.</p> <p>Facet2 from target system with same status.</p> <p>The software adds imported facets at the end of existing facets in the target section.</p>
– The item does not exist				

Import Rules for Executables and Mappings

Communication mapping is created with the mapping information from the export file if the following conditions are fulfilled:

- For peer to peer communication: The export file contains necessary data, page 269 of both the consumer and the owner projects and the corresponding elements are selected for import.
- The I/O scanner service of the client controller that is mapped to the imported executable is enabled.
- Sufficient free memory is available in the client and/or server memory length property of the controllers to fit the communication channels that exist in the export file.
- The client and server controllers that are mapped to the imported executables are connected to the same Ethernet network.
- For I/O device communication channels, in addition:
 - The I/O scanner service that is imported needs to match with the service of the topological I/O device entity with which communication is to be implemented. Refer to the description of the **Service** column in the **Device IO** section of the **Communication Mapping Editor** window, page 583.
 - The topological I/O device entity with which communication is to be implemented needs to be connected to the same Ethernet network as the controller.

If the export file contains mapped peer to peer variables, refer to the topic describing import rules for peer to peer communication variables, page 277.

If the software is not able to import communication mappings, a dialog box opens, page 282 indicating the reason.

The software updates the **Build State**, page 636 of a **Built** executable after import depending on the scenario.

NOTE: if the system into which you are importing a Control project already contains a Control project and an executable with the same identifier, the software applies import rules to executables and service, hardware, and communication mappings. These rules are described in the following tables.

The table describes the import rules that apply to **Built** executables based on different scenarios.

Contents of export file selected for import	Item in Control project of target system	Resulting built state in Control project of target system after import
Executable1	Does not exist	Out Of Date
Executable1	Executable1	Out Of Date
Executable1	Executable2	Executable1: Out Of Date. Executable2: Out Of Date.

The table describes the import rules that apply to mappings based on different scenarios.

Contents of export file selected for import	Item in Control project of target system	Result in Control project of target system after import
Service mapping	Exists	Service mapping of target system is not changed.
	Does not exist	Service mapping is created with the mapping information from the export file.
Hardware mapping	Exists	<p>Hardware mapping of the target project is updated with the mapping information from the export file in the following way:</p> <ul style="list-style-type: none"> • Existing mappings of the target project are overwritten. • New mappings are created. • Mappings of the target project that are not affected by the import process are maintained. <p>NOTE: Any new mapping for which the required mapping interface from the topological instance of the target project is not present is rejected.</p>
	Does not exist	<p>Hardware mapping is created with the mapping information from the export file.</p> <p>NOTE: Any mapping for which the required mapping interface from the topological instance of the target project is not present is rejected.</p>

Contents of export file selected for import	Item in Control project of target system	Result in Control project of target system after import
Communication mapping	Exists	<p>Communication mapping is overwritten as a whole if the following conditions are fulfilled:</p> <ul style="list-style-type: none"> For peer to peer communication: The export file contains necessary data, page 269 of both the consumer and the owner projects and the corresponding elements are selected for import. Sufficient free memory is available in the client and/or server memory length property of the controllers to fit the communication channels that exist in the export file. For I/O device communication channels, in addition: <ul style="list-style-type: none"> The I/O scanner service that is imported needs to match with the service of the topological I/O device entity with which communication is to be implemented. Refer to the description of the Service column in the Device IO section of the Communication Mapping Editor window, page 583. The topological I/O device entity with which communication is to be implemented needs to be connected to the same Ethernet network as the controller. If a service mapping exists, it needs to be identical to the service mapping contained in the export file for the Control project that you want to import. <p>NOTE: If the export file contains mapped peer to peer variables, refer to the topic describing import rules for peer to peer communication variables, page 277.</p>
Communication mapping	Does not exist	<p>Communication mapping is created with the mapping information from the export file if the following conditions are fulfilled:</p> <ul style="list-style-type: none"> For peer to peer communication: The export file contains necessary data, page 269 of both the consumer and the owner projects and the corresponding elements are selected for import. The I/O scanner service of the client controller that is mapped to the imported executable is enabled. Sufficient free memory is available in the client and/or server memory length property of the controllers to fit the communication channels that exist in the export file. The client and server controllers that are mapped to the imported executables are connected to the same Ethernet network. For I/O device communication channels, in addition: <ul style="list-style-type: none"> The I/O scanner service that is imported needs to match with the service of the topological I/O device entity with which communication is to be implemented. Refer to the description of the Service column in the Device IO section of the Communication Mapping Editor window, page 583. The topological I/O device entity with which communication is to be implemented needs to be connected to the same Ethernet network as the controller. If a service mapping exists, it needs to be identical to the service mapping contained in the export file for the Control project that you want to import. <p>NOTE: If the export file contains mapped peer to peer variables, refer to the topic describing import rules for peer to peer communication variables, page 277.</p>

Import Rules for Peer to Peer Communication Variables

To be able to import communication mappings as a whole, the software also needs to be able to create variable mappings that exist in the export file.

If none of the imported Control projects exist in the target system, the necessary variables are created, given that import rules for project-related items are satisfied.

If either Control project that you are importing already exists in the target system but a variable that is mapped to a peer to peer communication channel present in the export file has been modified or deleted, two scenarios are possible:

- The variable was created by refining the Control project. In such case, when you import communication mappings, the software is not able to recreate the variable. Therefore, it cannot create variables mappings and the entire import process is canceled and rolled back. The software displays a dialog box to inform you.

You need to recreate the variable manually to be able to import communication mapping.

- The variable was created by a facet assigned to the Control project. In such case, if you select the constituent file of the section to which the facet is assigned, the import process is able to recreate the required variable and the import succeeds, given that other necessary conditions are fulfilled.

The following example illustrates the second scenario: You have created two Control projects, *Owner* and *Consumer* between which a peer to peer communication channel exists. *Var_1* is mapped to this communication channel. The variable exists as network variable in *Consumer* (with *P2P* as custom attribute) and as variable in *Owner*.

You have exported both projects with their complete contents and have kept on working on both projects, which altered the communication mapping . After a while, you realize that you are not satisfied with some changes you made and therefore, you want to recover the original projects and communication mapping while keeping some of your changes (for example, you want to keep an FDB section that you have added to *Consumer* after exporting).

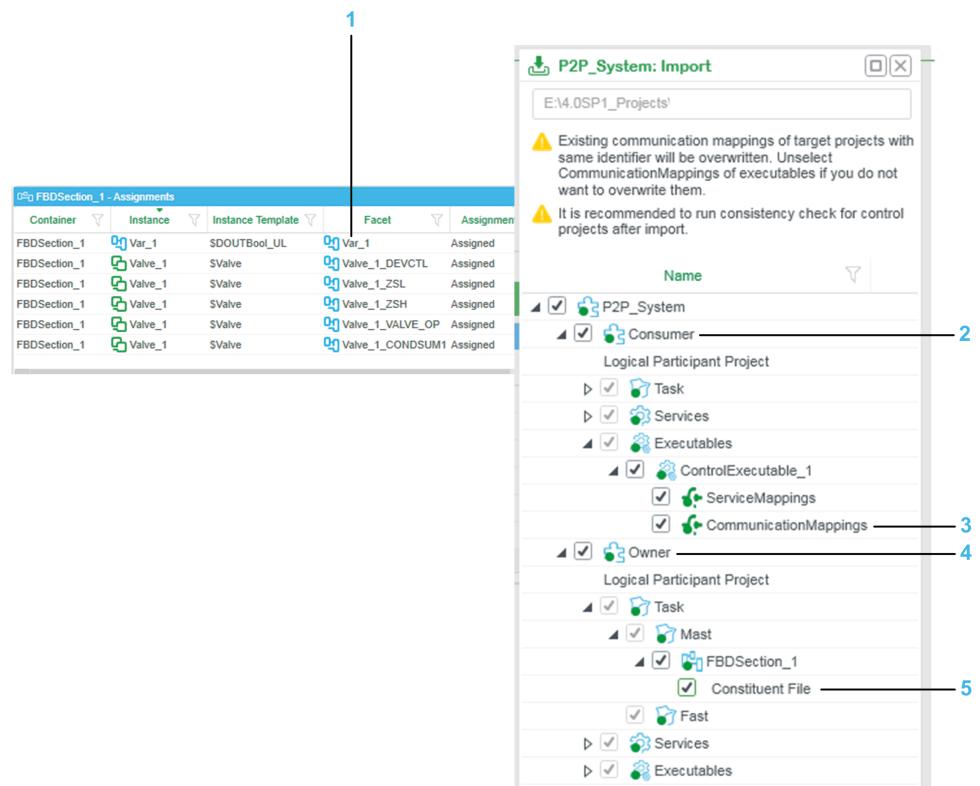
You import the export file in the **Projects Explorer** where *Owner* and *Consumer* Control exist. However, you have renamed *Var_1* in the *Owner* project.

Because you want to import communication mappings, you have selected both Control projects in the **Import** window.

Because *Var_1* had been created by a facet that is assigned, for example, to *FBDSection_1* in the *Owner* Control project, by selecting the constituent file, page 274 of *FBDSection_1* in the **Import** window, you are able to recreate the variable and import communication mappings.

NOTE: A network variable can be used only once per project, independently of the number of Control executables that exist in the project.

The following figure illustrates the import example by using a target system in which the *Owner* and *Consumer* projects exist and showing the objects that are selected for import.



Item	Description
1	<i>FBDSection_1</i> of Owner Control project showing the facet that creates <i>Var_1</i> .
2	<i>Consumer</i> Control project.
3	Communication mappings of <i>Consumer</i> .
4	<i>Owner</i> Control project.
5	Constituent file of <i>FBDSection_1</i> to which facet that creates <i>Var_1</i> is assigned.

Import Rules for Data Backup Files

When you import a Control project and a Control project with the same identifier already exists in the target system, if you have selected to import the data backup file and a file with the following identical identification information, page 733 already exists, the file is not imported:

- Time stamp.
- Identifier of the executable.
- Identifier of the controller and role.
- Description.
- Type of data that is backed up.

NOTE: The notification panel displays information about the not imported data backup file.

Opening the Project Import Window

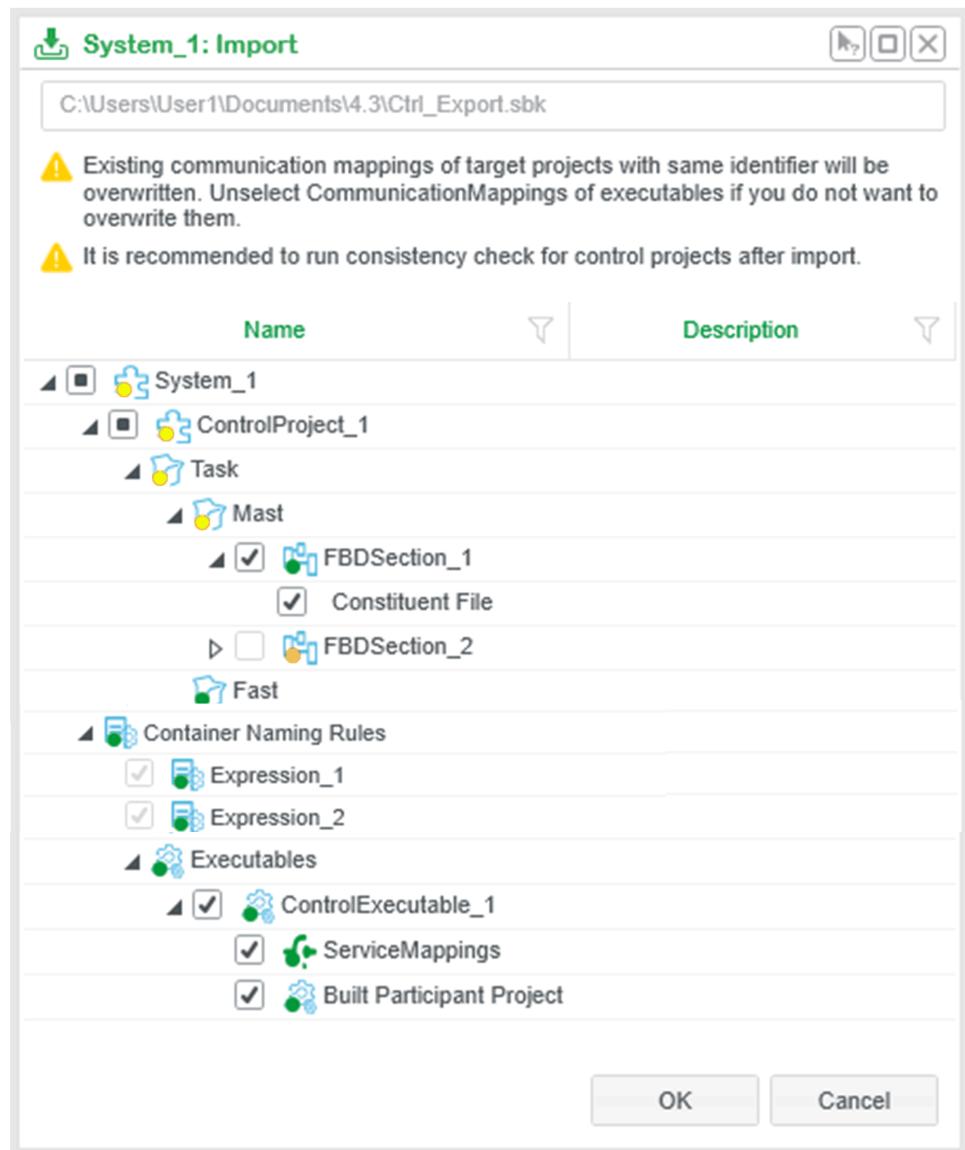
To open the **Import** window, proceed as follows.

Step	Action
1	Right-click the system root folder in a Project Browser of the Project Explorer and select Import . Result: The Import dialog box opens.
2	Browse to the export file (.sbk) that contains the Control project information that you want to import and click Open . Result: The Import window opens and displays the contents of the file that you have selected once it has read the data contained in the file. This can take time depending on the number and size of projects contained in the file. NOTE: You can abort the command, page 282.

Project Import Window

The **Import** window lets you select Control project information that is contained in an export file and that you can import into an existing system. You can select one or more entire Control Participant project and/or parts of them.

The following figure shows an example of the **Import** window displaying the contents of a Control project export file. In this example, the check box for *FBDSection_2* is cleared because the section cannot be imported. To indicate that at least one item of the Control project is not selected, the check box of parent nodes are filled with a square.



The table indicates the meaning of the colored dots that appear in the **Import** window next to items of a Control project.

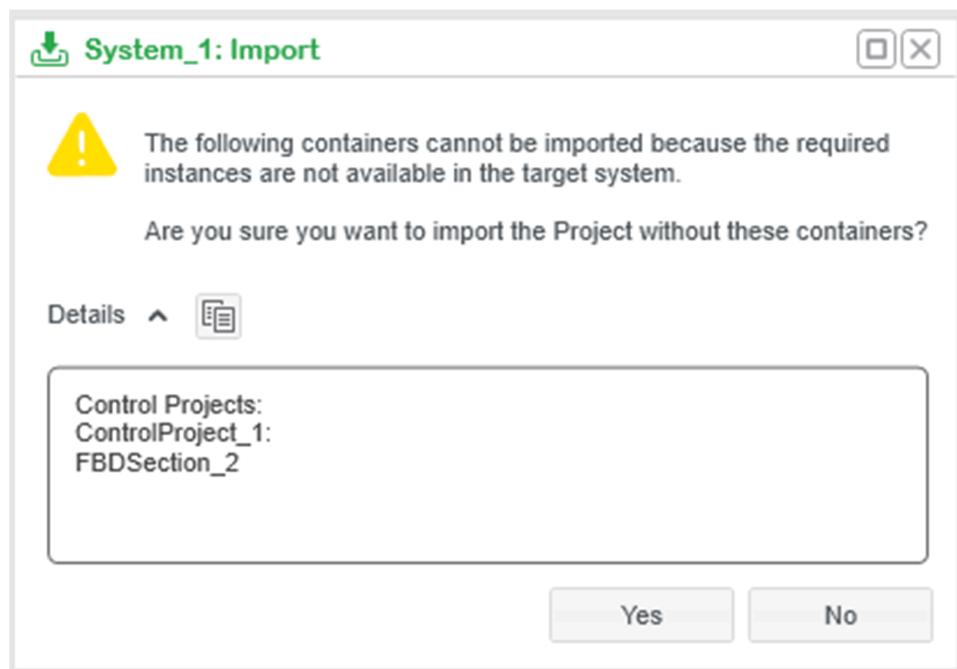
Color code	Description
Green	You can import the item.
Red	<ul style="list-style-type: none"> At the section level: The section contains at least one facet that is referenced by an instance, which is not present in the application, page 272. At the executable level: A project-related topological object, page 272 is not present in the target system. <p>A tooltip provides additional information about the issue. NOTE: You cannot import items that are tagged with a red dot.</p>
Yellow	<p>The item contains at least one subitem that is tagged with a red dot.</p> <p>The item can be imported partially only.</p>

NOTE: To expand/collapse a node in the **Import** window, select the node and press **Enter**.

Confirmation Dialog Box For Sections That Cannot Be Imported

The following figure shows an example of the dialog box that opens when a section of a Control project selected for import is tagged with a red dot and you click **OK** in the **Import** window.

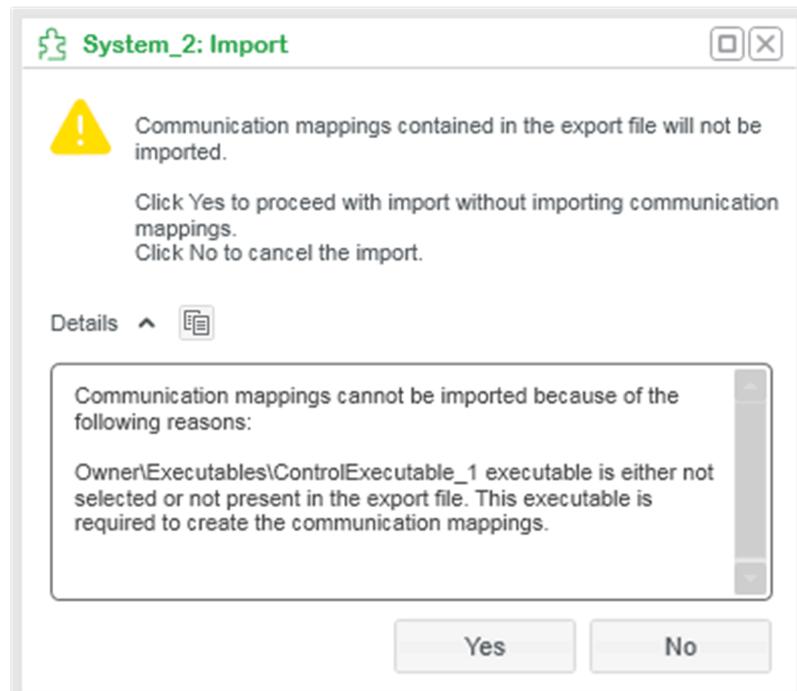
If you are importing also Supervision projects, containers are grouped by Participant then by project.



Button	Description
Yes	Closes the dialog box, the Import window, and imports the selected projects without sections tagged with a red dot.
No	Closes the dialog box and reverts to the Import window without importing projects. You can modify the selection of projects or cancel the import operation.

Confirmation Dialog Box For Communication Mappings That Cannot Be Imported

The following figure shows an example of the dialog box that opens when you click **OK** in the **Import** window and the software detects that communication mappings selected for import cannot be imported. It indicates the identifier of the project and the corresponding executable, and the reason. The **Import** window is closed before the dialog box opens.



Button	Description
Yes	Closes the dialog box and imports the selected projects without communication mappings. NOTE: This may result in variable mappings being present in the target system while the required variables do not exist.
No	Closes the dialog box and cancels the import operation.

Aborting Import Tasks

The abort icon is displayed in the notification panel, page 73 at two different stages of the import process:

- After clicking **Open** when you select the export file, page 279. Clicking the icon cancels the operation to read the contents of the export file. The **Import** window does not open.
- After confirming the import by clicking **OK** in the **Import** window, the abort icon is displayed. Clicking the icon cancels the task. The project is not imported.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are importing a small project, each task may complete before you are able to click the icon.

Importing a Control Project

To import data of a Control project into an existing system, proceed as follows.

Step	Action	Result
1	In the Import window, page 279, select the items that you want to import.	-
2	Click OK .	If applicable rules are satisfied, the software: <ul style="list-style-type: none"> • Applies the import rules based on your import selection. • Proceeds with the import. • Replaces existing communication mapping if you had selected to import them. • Displays an import summary dialog box.
3	In the Project Browser , right-click the Control project into which you have made the import and select Refine .	The Control Participant window opens.
4	Click the consistency check button, page 450.	The software proceeds with a consistency check and displays a notification to report findings once the check is completed.
5	Take the necessary corrective actions, page 451.	-

NOTE: If the software detects incompatibilities during the import process, it cancels and rolls back the import operation, and displays a dialog box to inform you. In such case, take the necessary corrective actions and start the import over. For example, if the incompatibility is related to variable mappings, you can clear the check box next to **CommunicationMappings** to import Control project data but no communication mapping.

Importing a Control Expert Project

Overview

The **Import Control Project** command allows you to create a Control project in the **Project Explorer** by importing data of a project that was created by using Control Expert and which is contained in a compatible project file (.stu or .xef format).

After import, you will have access to the same commands from the project node as if you had created the project in the **Project Browser**.

Project Requirements

To be able to import a project created with Control Expert, the following requirements need to be satisfied:

- The version of Control Expert with which the export file was created needs to be identical to the version of the Control Participant. To determine the version of the Control Participant, open it (for example, by refining a Control project) and open the **About** dialog box from the **Help** menu.
- The version of the EcoStruxure Control Expert DTM Library that is installed in Control Expert needs to be the same as the one installed in the Control Participant. For information on the installed version, refer to the platform release notes.
- No application password is set for the Control project.

Imported Data

Project data contained in the file (.stu or .xef)	Contained in the Control Participant project after import	Visible in the Project Explorer after import
FBD sections	Yes	Yes Sections are created in their respective MAST and FAST task folders in the Assignment Editor .
Sections in the following programming languages: <ul style="list-style-type: none"> • Ladder Diagram (LD) • Instruction List (IL) • Structured Text (ST) • Sequential Function Chart (SFC) • Ladder Logic 984 (LL984) 	Yes	
Constituents of any section	Yes	No
Configuration	No	Refine the imported project to view the information. NOTE: Only data related to the program becomes part of the logical Control project. Other data (for example, data related to the configuration) is ignored during build although it is visible when you refine the logical project.
Other global project information (for example, variables, types, project settings...)	Yes	
Project platform	Yes	Yes Controller Family attribute, page 261.

NOTE: The name of the file containing the project data becomes the identifier of the Control project that the software creates during the import process.

Importing a Control Expert Project

To import a Control Expert project into an existing system, proceed as follows.

Step	Action
1	Right-click the system root folder in the Control Project Browser of the Project Explorer and select Import Control Project . Result: The Open dialog box opens.
2	Select the file format that you want to import.
3	Browse to the location of the Control Expert project file that you want to import and click Open . Result: The software creates a Control project and displays it in the Control Project Browser .

Managing Data Backup Files

Overview

The **Manage Data Backup Files** dialog box lets you view and delete data backup files, page 733 that exist for all the executables of a Control Participant project.

The dialog box shows the information that was associated to each backup file at the time of creation, including the type of data that was selected.

By default, backup files are listed in descending order of time stamp (most recent one on top). You can sort and filter, page 118 the information that is displayed.

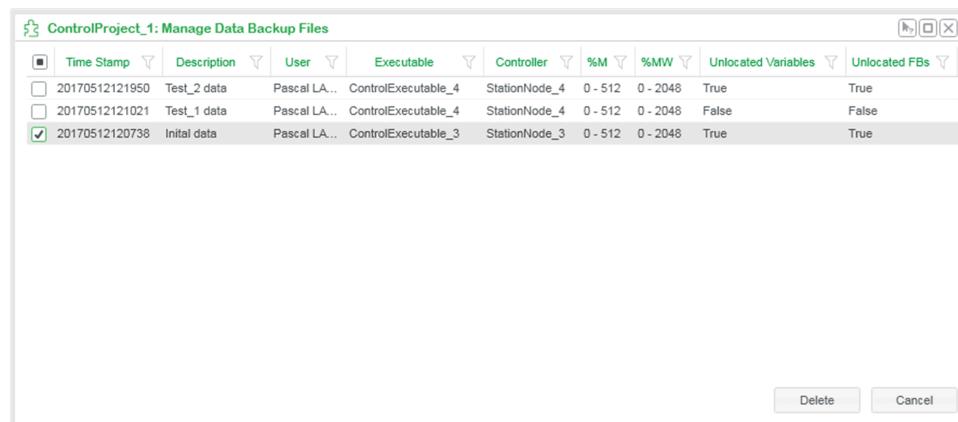
Opening the Manage Data Backup Files Dialog Box

To open the **Manage Data Backup Files** dialog box, right-click a Control Participant project in the **Project Browser** and select **Manage Data Backup Files**.

If no data backup file exists, the command is not available.

Manage Data Backup Files Dialog Box

The following figure shows an example of the **Manage Data Backup Files** dialog box, which shows the data backup files that have been created for executables of **ControlProject_1**.



Column header	Description
Time Stamp	Date and time of the backup file creation.
Description	Description that was entered when the backup file was created.
User	Name of the user who was logged on when the backup file was created.
Executable	Identifier of the Control executable whose data is contained in the backup file.
Controller	Identifier of the controller that was running the executable whose data was backed up. For redundant controllers, it also indicates the role of the controller that was selected when the backup file was created.
%M	Indicates the memory range of located variables of boolean data type that are contained in the backup file. If no such data was selected during data backup, the value shown is N/A .
%MW	Indicates the memory range of located variables of WORD data type that are contained in the backup file. If no such data was selected during data backup, the value shown is N/A .

Column header	Description
Unlocated Variables	<p>True indicates that this type of data was selected during data backup.</p> <p>Possible values:</p> <ul style="list-style-type: none">• True• False
Unlocated FBs	

Deleting Data Backup Files

To delete a data backup file, select the corresponding row and click **Delete**. The backup file is deleted after you confirm the command.

You can select more than one backup file at a time.

NOTE: When you delete a Control Participant project or the controller to which it is deployed, the corresponding data backup files are deleted as well.

Supervision Project Definition Stage

Overview

This section describes how to create Supervision projects of a system.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Internal Supervision Architecture

Task Description

As a reminder, the following figure describes the internal Supervision architecture for the execution of tasks:



1 Report scheduling and execution

2 Alarm acquisition and processing

3 Trend acquisition and storage

4 I/O communications

5 Graphical interface, events, local I/O, and client services

NOTE: Report, alarm, and trend servers also act as clients of the I/O server.

Creating Supervision Projects

Creating Supervision Projects

To create a Supervision project, proceed as follows.

Step	Action
1	<p>In the Supervision Project Browser, right-click the system root folder and select Create Supervision Project.</p> <p>Result: The Supervision is shown in the tree view.</p>

NOTE: For each system, you can create as many Supervision projects as needed to meet the requirements of the system.

Supervision Project Actions

The table describes the Supervision project context menu commands.

Command	Description	Applies to
Create Cluster , page 295	Creates a new cluster with its default contents and displays it in the tree view.	Project Definition stage
Last Action Summary , page 327	Opens a dialog box that indicates the last action that was performed and a summary of the result.	
Assign Facets	Opens the Assignment Editor , which lets you create and manage tag containers and Supervision pages of the Supervision project and assign facets to them.	Assignment Stage, page 328
Generate	Starts a process that creates and/or updates in the Supervision Participant project the data of the entire Supervision project.	Generation stage, page 383
Refine	Opens the Supervision Participant, which allows you to modify the project.	Refinement stage, page 493
Advanced Settings , page 495	Opens the Supervision Participant, which allows you to change advanced project settings that will be activated in the station nodes where you deploy the Supervision project.	Refinement stage
Include Projects , page 496	<p>Opens a dialog box, which allows you to:</p> <ul style="list-style-type: none"> • Add external included projects to be deployed to the Supervision nodes. • To remove included projects that are attached to the Supervision and stored in the contents repository. <p>NOTE: You cannot add an included project if the software has already added to the Participant project an included project with the same name.</p>	
Attach User Files , page 497	<p>Opens a dialog box, which allows you to:</p> <ul style="list-style-type: none"> • Attach external files to be deployed to the Supervision nodes. • To remove files that are attached to the Supervision project and stored in the contents repository. 	Project Definition stage
Export , page 307	Opens a Save As dialog box, which allows you to create an export file containing the Supervision project information.	
Settings	Opens the Settings window, page 263, which lets you configure various project-related settings.	Assignment stage
Delete	Deletes the Supervision project including components, assignments, and mappings.	–
Rename	Allows you to enter a new identifier for the project.	–
Properties	Opens the Properties window, which allows you to edit the properties of the Supervision project.	–

Containers Node Actions

The table describes the **Containers** node context menu commands.

Command	Description	Applies to
Assign Facets	Opens the Assignment Editor , which lets you create and manage tag containers and Supervision pages of all clusters and assign facets to them.	Assignment Stage, page 328
Create Tag Container	Lets you create a tag container, page 340 in an existing cluster. The Assignment Editor opens in the background.	
Create Page	Opens the Assignment Editor and creates a Supervision page, page 345 in the project.	
Generate	Starts a process to create and/or update in the Supervision Participant project the data for the containers of all clusters that exist in the project.	Generation Stage, page 383

Pages Node Actions

The table describes the **Pages** node context menu commands.

Command	Description	Applies to
Assign Facets	Opens the Assignment Editor , which lets you create and manage tag containers and Supervision pages of all clusters and assign facets to them.	Assignment Stage, page 328
Create Page	Opens the Assignment Editor and creates a Supervision page in the project.	
Generate	Starts a process to create and/or update in the Supervision Participant project the data for the pages that exist in the project.	

Cluster Node Actions

For a description of the context menu commands, refer to the topic describing how to create clusters, page 295.

Tags Node Actions

The table describes the **Tags** node context menu commands.

Command	Description	Applies to
Assign Facets	Opens the Assignment Editor , which lets you create and manage tag containers and Supervision pages of the parent cluster and assign facets to them.	Assignment Stage, page 328
Generate	Starts the generation process of tag containers and pages that exist in the parent cluster to populate new tags, update out-of-date tags, and/or remove deleted tags in the Supervision Participant project.	

Services Node Actions

The table describes some of the **Services** node context menu commands. For a description of the other commands, refer to the topic describing how to create Supervision servers and clients, page 296.

Command	Description	Applies to
Open	Opens the Services window, which lets you view and edit properties of the Alarm, report, and trend servers as well as clients that exist in the parent cluster.	Assignment Stage, page 328
Generate	Starts a process to create and/or update in the Supervision Participant project the data related to services.	Generation Stage, page 383

ClientEvents and ServerEvents Node Actions

The table describes the **ClientEvents** and **ServerEvents** nodes context menu commands.

Command	Description	Applies to
Open	Opens the corresponding window, which lets you view the Supervision facets that are assigned to client or server events in the parent cluster.	Assignment Stage, page 328
Generate	Starts a process to create and/or update in the Supervision Participant project the data related to services in the parent cluster.	Generation Stage, page 383

Executables Node Actions

The table describes the **Executables** node context menu commands.

Command	Description	Applies to
Create Executable	Creates one or more Supervision executables, page 619, which let you associate the Supervision project infrastructure to stations nodes and the I/O devices to controllers.	Mapping Stage

Supervision Project Properties

Right-click the Supervision project and select **Properties**.

You can edit the following items.

Item		Description
General	Identifier	The identifier of the Supervision project, which must be unique within the system.
	Description	Optional. You can enter a description of the project with free form text.

Project Settings Window

The **Settings** window opens when you select the **Settings** command in the context menu of the Supervision project.

The following table describes the settings that are specific to Supervision projects. For a description of the other settings, refer to the topic describing project settings, page 263.

Section	Description
Page Templates	<p>Lets you add and remove Supervision page templates and select a template to be used by default when you create pages.</p> <p>A page template is the combination of a style, a resolution, and a template.</p> <p>Some page templates are provided by default.</p> <p>The page templates that appear in this section and your selection apply only to this Supervision project.</p> <p>For details on how to add, remove, and use page templates, refer to the topic describing the selection of Supervision page templates, page 346.</p>

Creating Supervision Project Components

Overview

This section describes how to use the **Project Explorer** to create the components of Supervision projects.

During build, one or more OPC Factory Server and/or OPC UA Server Expert configuration files (.xml) are automatically generated, page 643 based on the configuration of properties of the Supervision project components.

For information on how to create tag containers and Supervision pages, refer to the topic describing the creation of containers, page 329 in the Assignment stage.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Creating Clusters

Creating Clusters

To create a cluster, proceed as follows.

Step	Action
1	Right-click the Supervision project and select Create Cluster . Result: The software creates a new cluster node and its child nodes.

Cluster Actions

The table describes the cluster context menu commands.

Command	Description
Generate	Starts the generation process to populate the database of the Supervision Participant project for this cluster.
Delete	Deletes the cluster including its contents and unassigns the assigned Supervision data facets.
Rename	Lets you enter a new identifier for the cluster.
Properties	Opens the Properties window, which allows you to edit the properties of the cluster.

Cluster Properties

To open the **Properties** window of the cluster, right-click the cluster and select **Properties**.

You can edit the following items,

Item		Description
General	Identifier	The identifier must be unique within the Supervision project and satisfy applicable naming rules, page 89. NOTE: Changing the identifier of the cluster changes the assignment status of facets that are assigned to a tag container of the cluster to Out Of Date .
	Description	Optional. You can enter a description of the cluster with free form text.

Creating Supervision Servers and Clients

Overview

You can create Supervision servers in the **Services** node of a cluster.

The I/O server instance represents a server that is set up on an operation server (see EcoStruxure™ Process Expert, Installation and Configuration Guide), on which Supervision and OPC Factory Server or OPC UA Server Expert software is installed. The operation server is represented in the topology of the system by a station node, page 516, which acts as an engine to run the Supervision project.

The alarm, trend, and report server instances represent servers that are set up on a single or distinct operation servers, on which Supervision software is installed.

You can create up to two alarm, report, and trend servers per cluster.

NOTE: Generate the Supervision project after you make changes related to components described in this topic.

Creating Supervision Servers

To create I/O, alarm, trend, and report servers in the cluster, proceed as follows.

Step	Action
1	<p>Right-click the Services node and select the corresponding command.</p> <p>Result: The new server is created.</p> <p>NOTE: Only I/O servers are shown in the project browser. To view other servers, open the Services node.</p>

NOTE: The software may automatically create servers during the assignment stage if you have not created them already. For more information, refer to the Supervision project Assignment stage, page 364.

Supervision I/O Server Properties

To open the **Properties** window of an I/O server, right-click the server and select **Properties**.

You can edit the following items.

Item		Description
General	Identifier	The identifier must be unique within the Supervision project.
	Description	Optional. You can enter a description of the server with free form text.
Attributes	Port	Communication port the server listens on. Default value: 2082.
	Legacy Port	Communication port that is used for legacy connections to the server. Default value: 2078.

Supervision Servers Common Properties

To view the alarm, trend, and report servers that exist in the cluster and their properties, right-click the **Services** node and select **Open**.

You can view and/or edit the following items.

Item		Description
General	Identifier	<p>The identifier must be unique within the Supervision project.</p> <p>Default value: Server_n_P where:</p> <ul style="list-style-type: none"> • Server is the name of the service provided by the server (for example, <i>Alarm</i>). • n is an incremental number starting at 1. • P stands for primary. The suffix of the second server instance that you create is _S for standby, even if you rename the first instance.
	Description	<p>Optional.</p> <p>You can enter a description of the server with free form text.</p>
Attributes	Port	<p>Communication port the server listens on.</p> <p>Default value:</p> <ul style="list-style-type: none"> • Alarm server: 2080 • Report server: 2084 • Trend server: 2085 <p>Edit the value if you are running more than one server on the station node.</p>
	Legacy Port	<p>Communication port that is used for legacy connections to the server.</p> <p>Default value:</p> <ul style="list-style-type: none"> • Alarm server: 2076 • Report server: 2075 • Trend server: 2077
	Database Port	<p>The attribute is available for alarm servers only.</p> <p>Communication port the server database listens on.</p> <p>Default value: 5482</p> <p>Edit the value if you are running more than one alarm server on the station node.</p>
	Mode	<p>You can select one of the following values from the menu:</p> <ul style="list-style-type: none"> • None • Primary: Defines the server as the primary server. Default value for the server with _P suffix. • Standby: Defines the server as the standby server. Default value for the server with _S suffix.
	Extended Memory	<p>The attribute is available for alarm servers only.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • False: Default value. • True: The alarm server operates in 64-bit extended memory mode. This allows the alarm server process to utilize memory beyond the 4GB limit. <p>For more information on this mode of operation, refer to <i>Alarm Server Process</i> in the help of the Supervision Participant.</p>

Creating Supervision Clients

You can create Supervision Clients in the **Services** node of a cluster.

The client instance can represent a Supervision client (runtime) that runs:

- On the operation server.
- On an operator station.

To create a client for the Supervision project, proceed as follows.

Step	Action
1	<p>Right-click the Services node and select Create Client.</p> <p>Result: The client is created but not shown in the project browser.</p>

Supervision Client Properties

To view the clients that exist in the cluster and their properties, right-click the **Services** node and select **Open**.

You can edit the following items.

Item	Description
General	Identifier The identifier must be unique within the Supervision project.
	Description Optional. You can enter a description of the client with free form text.

Creating I/O Devices

Overview

Typically, an I/O device is created for one controller. If the data is present in two controllers, create an additional I/O device for the second controller manually. Create a second tag container, page 340 for data of the second controller and associate this tag container to the second I/O device.

The software creates additional I/O devices, page 300 when you create calculated or disk variable tags.

NOTE: Monitoring data of a device, page 830 communicating by using the Modbus TCP protocol (or Modbus through an Ethernet gateway) from the Supervision infrastructure also requires a dedicated I/O device per Modbus device.

Prerequisites

An I/O server must exist in the cluster.

Creating I/O Devices

To create an I/O device in the cluster, proceed as follows.

Step	Action
1	Expand the Services node and the I/O server, right-click the IODevices node, and select Create IO Device . Result: The I/O device is created but not shown in the project browser.

NOTE: The software automatically creates the I/O device during the assignment stage if you have not created it already. For more information, refer to the Supervision project Assignment stage, page 364.

I/O Device Properties

To view the I/O devices that exist in the cluster and their properties, right-click the **IODevices** node and select **Open**.

You can edit the following items.

Item		Description
General	Identifier	<p>The identifier must be unique within the Supervision project.</p> <p>NOTE: Changing the identifier of the I/O device changes the assignment status of facets that are assigned to the tag container that is associated with the I/O device to Out Of Date if they were generated.</p>
	Description	<p>Optional.</p> <p>You can enter a description of the I/O device with free form text.</p>
Attributes	Address	<p>The value needs to be compatible with the protocol that is selected for the associated tag container.</p> <p>The default value is the alias name for OPC Factory Server: <i>Alias_<IODevice_n></i> where <i><IODevice_n></i> corresponds to the value of Identifier.</p> <p>NOTE: The attribute is disabled and cleared when Protocol of the associated tag container is set to OPCUA or OPCUA Embedded.</p> <p>For these two protocols, the following temporary address is created for the I/O device when you generate the Supervision project or the cluster:</p> <ul style="list-style-type: none"> • For OPCUA: <i>opc.tcp://127.0.0.x:49152/OPCUAServerExpert</i> • For OPCUA Embedded: <i>opc.tcp://127.0.0.x:4840</i> <p>During build, page 643, the IP address placeholder (127.0.0.x) and the port value are updated based on the mapping.</p>
	StartupMode	<p>Select the role of the I/O device in terms of communication with the controller:</p> <ul style="list-style-type: none"> • Primary (default value) • Standby • StandbyWrite • None
	Priority	<p>You can assign the order in which the I/O device attempts to communicate with the tag container that it is associated with in case more than one I/O device is associated with the container.</p> <p>Set the values as follows depending on the role of the I/O device:</p> <ul style="list-style-type: none"> • Primary: 1 or blank (default value) • Secondary: 2 or higher <p>NOTE: If you associate two I/O devices configured as primary to one tag container, verify that Priority is configured with a different value for each.</p>
	Memory	<p>Select one of the following values depending on how you want the I/O device to communicate with the controller:</p> <p><i>False</i>: Default value. The I/O device is online with the controller</p> <p><i>True</i>: The I/O device is not connected to the controller. Select this setting for testing purposes, for example.</p>
Relationships	TagContainer	<p>Select the tag container that is associated with the I/O device.</p> <p>NOTE:</p> <ul style="list-style-type: none"> • By default, the software associates the I/O device with the tag container for which the I/O device has been automatically created, page 364 during the assignment of Supervision data facets. • Changing the relationship of the I/O device to the tag container changes the assignment status of facets assigned to the tag containers affected by this change to Out Of Date.
	Ports	<p>Select the communication port, page 302 that is associated to the I/O device.</p> <p>NOTE: By default, the software associates the I/O device with either the communication port that has been automatically created, page 364 during the assignment of Supervision data facets or with the one that you have created manually.</p>

I/O Devices Created for Calculated and Disk Variable Tags

When you generate a Supervision data facet that references a non-disabled **Calculated Variable Tag** or **Disk Variable Tag** element (see EcoStruxure™ Process Expert, Global Templates, Reference Manual), one specific I/O device is automatically created per I/O server with the same properties as the default I/O device except for the following. These specific I/O devices are only visible when you refine the Supervision project.

Property	For a calculated variable tag	For a disk variable tag
Identifier	<i>CicodeIODevice_n</i> where <i>n</i> is an integer starting at 1 so that the identifier remains unique within the I/O sever.	<i>Disk_IODeviceID_n</i> where: <ul style="list-style-type: none"> • <i>IODeviceID</i> is the identifier of the I/O device that is associated to the tag container to which the data facet is assigned. • <i>n</i> is an optional integer starting at 1 that is added so that the identifier remains unique within the I/O sever.
Address	Blank	Blank
Ports	Blank	Blank
Memory	Same value as default I/O device	True
Persist		True

Creating Communication Ports

Overview

Before you can create a communication port, you need to create a communication board unless one already exists.

You need to create a communication board for each communication driver that you are using. For example, if you are using both OFSOPC and OPCUA, create one board for each and configure them accordingly.

To view the communication ports that exist, right-click a board and select **Open**.

Prerequisites

An I/O server must exist in the cluster.

Creating a Communication Board

To create a communication board, proceed as follows:

Step	Action
1	<p>Expand the Services node and the I/O server, right-click the Communications node, and select Create Board.</p> <p>Result: The board is created.</p>

NOTE:

- The software automatically creates one communication board and port during the assignment stage if you have not created it already. For more information, refer to the Supervision Assignment stage, page 364.
- Deleting a communication board deletes the associated communication ports.

Communication Board Properties

To open the **Properties** window of the communication board, right-click it, and select **Properties**.

You can edit the following items. For details, refer to *Board Properties* in the help of the Supervision Participant.

Item		Description
General	Identifier	<p>The identifier must be unique within the I/O server.</p> <p>16 characters maximum.</p>
	Description	<p>Optional.</p> <p>You can enter a description of the board with free form text.</p>
Attributes	Protocol	<p>Indicates the protocol for communicating with the I/O device and that is to be used for configuring the I/O device.</p> <p>When the protocol for the tag container is set to OPCUA Embedded, select OPCUA.</p> <p>When you configure a user-defined protocol for a tag container, page 340, enter a protocol for the board that is compatible and installed in the Supervision Participant. To verify which ones are installed, refine the Supervision Participant project and expand an entry in the Board Type column of Components & Mapping in the Topology activity.</p> <p>Default value:</p> <ul style="list-style-type: none"> Same as Protocol (that is, OFSOPC or OPCUA) if the value of Protocol for all the tag containers that are associated to an I/O device whose communication port is associated to the board is the same; otherwise, OFSOPC. OFSOPC when you create the board manually.

Item	Description
Address	Default value: 0 when the board is created by the software.

NOTE: The *OFSOPC* driver configuration implements an automatic and adaptive strategy, creating and deleting OPC groups and OPC items dynamically to maximize throughput. For information on other parameters affecting performance when using this driver configuration, refer to Performance Considerations (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

NOTE: For the *OPCUA* driver, the software adds driver-specific parameters to the *citect.ini* file (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide) (for example, subscription and connection parameters), which are configured with default values and apply to a given I/O device. For details, refer to *OPCUA Driver* in the *Driver Reference Help* of the Supervision Participant.

Creating Communication Ports

For each communication board, you can create several communication ports.

To create a communication port, proceed as follows.

Step	Action
1	Expand the Services node, right-click the board, and select Create Port . Result: The port is created but not shown in the project browser.

Communication Port Properties

To view the ports that exist for a board and their properties, right-click the board and select **Open**.

You can edit the following items.

Item	Description	
General	Identifier	The identifier must be unique within the I/O server.
	Description	Optional. You can enter a description of the port with free text.
Attributes	Port Number	Interrupt number of the port. Leave it blank if you are using the COM port of the computer.
	Special Opt	Lets you enter any special options that are supported by the board. 32 characters maximum. For details, refer to <i>Hardware Arrangements</i> in the help of the Supervision Participant.
Relationships	IoDevice	Select the I/O devices that are associated with the communication port from the menu. The I/O device, page 299 must exist in the Supervision project cluster to be available for selection. NOTE: By default, the software associates the port with the I/O device that has been automatically created, page 364 during the assignment of Supervision data facets. If you have created the I/O device manually, associate it with the port.

Implementing Redundancy

Overview

The Supervision Participant allows you to build redundancy into the system at different levels to help increase the degree of protection and/or reliability. The common ways to implement redundancy are:

- I/O server redundancy
- Alarm, trend, and report server redundancy
- Network redundancy

The following topics describe the additional steps that you need to complete in order to implement such redundancy by starting from a typical Supervision project, which consists of one of each:

- Cluster
- Tag container
- I/O server with its client
- I/O device
- Communication board
- Communication port

For more information, refer to the topic of the Supervision Participant help, page 93, which describes how to build redundancy into a system.

I/O Server Redundancy

To implement I/O server redundancy, proceed as follows.

Step	Action	Refer to
1	From the Services node of the Supervision project, create a second I/O server.	Project definition stage
2	From the Services node, create a client for this second I/O server.	
3	From the Communications node of the second I/O server, create a communication board.	
4	From this board, create a communication port.	
5	From the IODevices node of the second I/O server, create an I/O device.	
6	In the properties of the second I/O device, associate the I/O device with the tag container of the Supervision project, which is already associated to the first I/O device. Result: The software automatically configures the StartupMode of the second I/O device as <i>Standby</i> with Priority 2 (or the next available value).	
7	In the same window, configure the Ports property as <i>Port_2</i> , which is the port of the communication board that you have created in step 4.	
8	In the properties of the communication port of the second I/O server, verify that the port is associated to the standby I/O device under IODevice .	
9	In the properties of the communication board of the second I/O server, configure Protocol with the same values as for the tag container, which is associated to the first I/O device.	
10	From the system root folder, create a station node entity representing the second I/O server.	Configuration stage
11	In the properties of the NIC of this station node, configure the IP address of the NIC.	
12	On this station node, create one Supervision and one OFS service instances.	
13	Connect this station node to the Ethernet network.	
14	Once you have generated the complete Supervision project, from the project executable, perform the service mapping of the second I/O server to the station node that you have created in step 10.	Mapping stage

Step	Action	Refer to
15	From the project executable, map the standby I/O device to the same controller to which you have mapped the primary I/O device (communication mapping).	
16	Configure the physical computers according to the topology of the system, build, and deploy the Supervision project.	Build and Deployment stage

Alarm, Trend, or Report Server Redundancy

To implement alarm, trend and/or report server redundancy, proceed as follows.

Step	Action	Refer to
1	From the Services node of the Supervision project, create a second alarm, trend and/or report server. Result: The software: <ul style="list-style-type: none">• Creates the server with the _s suffix in its identifier.• Configures its Mode as <i>Standby</i>.	Project definition stage
2	From the system root folder, create a station node entity representing each redundant alarm, trend and/or report server.	Configuration stage
3	In the Properties window of the NIC of each redundant station node, configure the IP address of each NIC.	
4	On each redundant station node, create one Supervision service instance.	
5	Connect each redundant station node to the Ethernet network.	
6	Once you have generated the complete Supervision project, from the project executable, map each standby server to the corresponding redundant station node (service mapping).	Mapping stage
7	Configure the physical computers according to the topology of the system, build, and deploy the Supervision project.	Build and Deployment stage

Network Redundancy

Network redundancy can be implemented for any Supervision station.

To implement network redundancy, for example, on the operation server, proceed as follows.

Step	Action	Refer to
1	In the station node entity representing the operation server (I/O server), create a second NIC.	Configuration stage
2	In the Properties window of the second NIC, configure its IP address. NOTE: It is not required to connect this NIC to an Ethernet network.	
3	Configure the physical computers according to the topology of the system.	–
4	Build the Supervision project by using the Build All command. NOTE: In the service mapping, no additional mapping is required as the engine that is mapped to the <i>I/OServer</i> service is the station node.	Build and Mapping stages
5	Deploy the Supervision project either from the executable or from the station node by selecting either NIC. Result: After restoring the Supervision project on the station node (operation server), both network addresses are created in the Supervision Participant and associated to the I/O server.	Deployment and Execution stages

NOTE: No specific action is required in the Supervision project.

Managing Supervision Projects

Overview

This section describes how to use the **Project Explorer** to manage Supervision projects and project contents.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Exporting a Supervision Project

Overview

The export functionality lets you create a non-editable export file (.sbk), which contains selected Supervision project information.

You can use this data with the Supervision project import, page 271 functionality to transfer Supervision project information to a different system or merge information from different Supervision projects.

NOTE: To export all the projects (Control and Supervision) of a system at once in one export file, right-click the root folder in either **Project Browser**, select **Export**, and follow the export procedure described in this section. Refer also to the topic describing the export and import of Control projects, page 265.

Software Version Compatibility

The version of EcoStruxure Process Expert that you use to import the information contained in the export file must be the same as the version that you used to create the export file (a service pack is considered the same version); otherwise you cannot proceed with the import.

Project Related Data

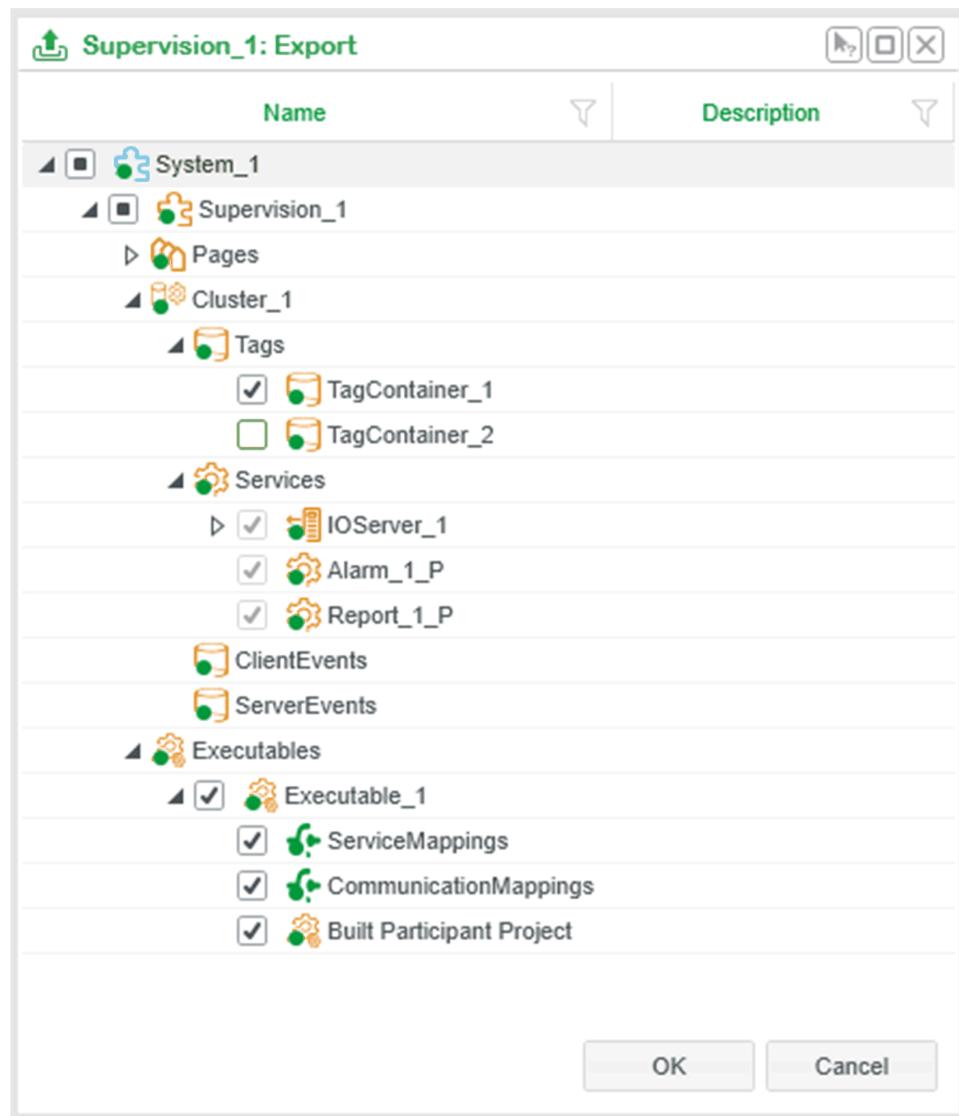
To be able to use the export file for importing a project into a different system, verify that the instances, their templates, and topological entities related to the project exist in the target system before importing the project, page 314. Use the export-import functionality, page 552 of the **Topology Explorer** first.

NOTE: The parameter values of instances that reference the exported facets are not contained in the export file. However, the export file contains information that allows the software to verify during import if the configuration of an instance is different from when its facets were exported.

Project Export Window

The Supervision project **Export** window lets you select, which information the export file will contain.

The following figure shows an example of the **Export** window where the check box for *TagContainer_2* has been cleared. To indicate that at least one item of the Supervision project is not selected, the check box of parent nodes are filled with a square.



The table indicates the meaning of the colored dots that appear in the **Export** window next to items of a Supervision project.

Color code	Description
Green	You can export the item.
Red	<p>You cannot export the item because it contains at least one facet with the following state:</p> <ul style="list-style-type: none"> • Deleted • Unassigned • Out of Date • Moved <p>Generate containers that contain facets with such states or clear their check box in the Export window in order to export the Supervision project.</p> <p>NOTE: You cannot export items that are tagged with a red dot.</p>
Yellow	<p>Indicates that the item contains at least one subitem that is tagged with a red dot.</p> <p>The item can be exported partially only.</p>

NOTE: To expand/collapse a node in the **Export** window, select the node and press **Enter**.

Export Settings

The table describes which Supervision project information you can export.

Item	Selected for export	Description
SupervisionProject_n	By default	Unselect the item to exclude the entire project from the export file. When you select the project, the associated equipment hierarchy, page 154 is exported.
Pages node	By default	You cannot unselect the item.
Pages	By default	Selecting a page automatically selects any genie facets that are assigned to it. Selecting a page automatically selects the logical Participant project.
Page Templates	By default	Default and user-created page templates, page 346. Unselecting a page template does not impact imported pages that use this template. You cannot unselect the default page templates.
Cluster_n	By default	You cannot unselect the item.
Tags node	By default	You cannot unselect the item.
Tag containers	By default	You can select tag containers individually. Selecting a tag container automatically selects any facets that are assigned to it.
Services node	By default	You cannot unselect the item.
Alarm servers	By default	Any associated boards and ports are selected by default. You cannot unselect the item.
I/O servers	By default	
Communications node	By default	
IODevices node	By default	
Trend servers	By default	
Report servers	By default	
Clients	By default	
ServerEvents	By default	Any associated I/O devices are selected by default. You cannot unselect the item.
ClientEvents	By default	
Executables	By default	
Executable_n	By default	
Service Mappings	By default	
Communication Mappings	By default	
Built Participant Project	By default	

Aborting Export Tasks

After confirming the export by clicking **Save** in the **Export** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The export file is not created.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are exporting a small project, the task may complete before you are able to click the icon.

Exporting a Supervision Project

To export a Supervision project to file, proceed as follows.

Step	Action
1	In the Supervision Project Browser , right-click the Supervision project that you want to export and select Export . Result: The Export window opens and displays the Supervision project that you have selected.
2	Select the project components that you want to export.
3	Click OK . Result: The Export window closes and the Export dialog box opens.
4	Select a location where you want to save the export file, enter a file name, and click Save . Result: The software creates the export file containing the selected Supervision project information. NOTE: If you click Cancel , the the Export dialog box closes without creating the export file and the selection you have made in the Export window is discarded.

Importing a Supervision Project

Overview

The **Import** command lets you import data of a Supervision project from an export file (.sbk), which was created by using the export, page 267 functionality of the **Project Explorer**.

If the system into which you are importing a Supervision project already contains a Supervision project with the same identifier, the software compares the data of both projects. It then proceeds with the partial import according to the import settings that you select and the import rules that are described in this topic.

NOTE: If the export file also contains information of a Control project, you can see the data that is related to the Control project, and have the choice to import it, page 271 as well.

Software Version Compatibility

The version of EcoStruxure Process Expert that you use to import the information contained in the export file must be the same as the version that you used to create the export file (a service pack is considered the same version); otherwise you cannot proceed with the import.

Import Settings

The table describes which Supervision project information you can import into a system.

Item	Selected for import ⁽¹⁾	Description
SupervisionProject_n	By default	Unselect the item to exclude the entire project from the export file.
Pages node	By default	You cannot unselect the item.
Pages	By default	<p>You can select containers (pages) individually⁽²⁾.</p> <p>Selecting a page automatically selects any genie facets that are assigned to it.</p> <p>Selecting a page automatically selects the logical Participant project.</p>
Page Templates	By default	<p>Default and user-created page templates, page 346.</p> <p>Unselecting a page template does not impact imported pages that use this template.</p> <p>You cannot unselect the default page templates.</p>
Cluster_n	By default	<p>At least one cluster must be selected.</p> <p>You cannot unselect a cluster when the export file contains only one cluster.</p>
Tags node	By default	You cannot unselect the item.
Tag containers	By default	<p>You can select tag containers individually⁽²⁾.</p> <p>Selecting a tag container automatically selects any facets that are assigned to it.</p> <p>A Constituent File is associated to each tag container.</p>
Tag container Constituent File	By default	<p>If the identifier of the imported tag container already exists in the target project, the selection of the constituent file has an impact on the import behavior, page 315.</p> <p>NOTE: The Constituent File check box is selected and disabled if the identifier of the imported tag container does not exist in the target project.</p>
Services node	By default	You cannot unselect the item.
Alarm servers	By default	
IOServer_n	By default	
Communications node	By default	
IODevices node	By default	
Trend servers	By default	
Report servers	By default	
Clients	By default	
ClientEvents	By default	
ServerEvents	By default	
Executables	By default	

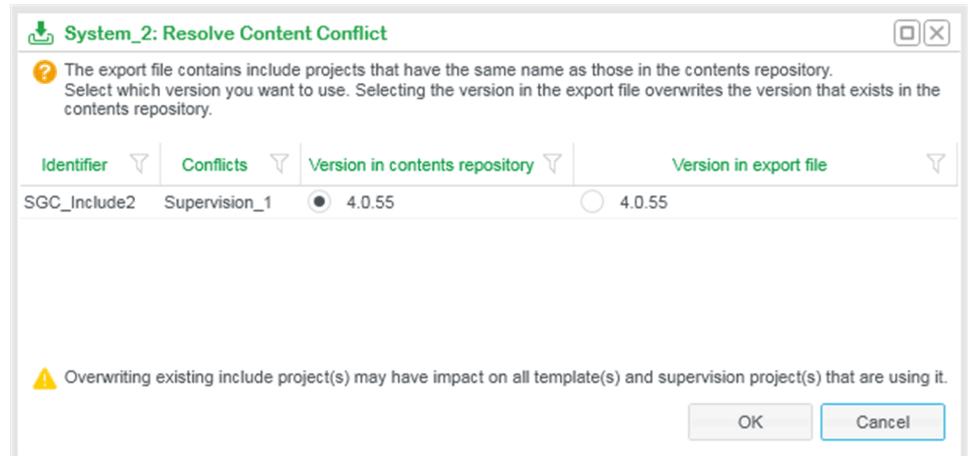
Item	Selected for import ⁽¹⁾	Description
Executable_n	By default or not selectable	If the target system already contains an executable with the same identifier, you cannot select the executable for import. Else, you can select each executable and its contents individually. Selecting an executable automatically selects any associated mapping.
	Service Mappings	If a service mapping already exists in the target system, you cannot select the service mapping for import. Contains data related to the service mapping including any required contents in the content repository.
	Communication Mappings	If a communication mapping already exists in the target system, you cannot select the communication mapping for import. Contains data related to the communication mapping.
	Built Participant Project	By default Represents the built Supervision Participant file (.ctz) (includes the OFS/OPC UA configuration files (.xml)). It appears and is selected only if the built Participant project was selected at the time of export.

(1) The information needs to be contained in the export file to be selected
(2) The item is available for selection only if it is tagged with a green dot; otherwise the item cannot be selected.

Import Rules for Included Projects

At the time of import, the software checks whether the content repository of the target system contains an included project that has the same identifier as the one contained in the export file.

The following figure shows an example of the **Resolve Content Conflict** dialog box that opens if the software detects the presence of an included project with the same identifier.



Header	Description
Identifier	The identifier of the included project for which a conflict exists.
Conflicts	The identifier of the Supervision Participant projects that exist in the target system and use the version of the included project that is in the content repository.
Version in contents repository	By selecting this check box, you use the version of the included project that is in the content repository. Selected by default.
Version in export file	By selecting this check box, you use the version of the included project that is contained in the export file, page 319 and overwrite the version that is in the repository.

Included projects contain data related to Supervision constituents (for example, genies, tags, and so on), which can be shared by several templates.

Replacing the existing included project has an impact on:

- The templates that use the contents of the included project.
- The Supervision Participant projects to which the included project is associated.

NOTICE

LOSS OF DATA

Resolve an included project conflict that arises during the import of a Supervision Participant project by selecting the correct version of the included project to be imported.

Failure to follow these instructions can result in unusable Supervision projects.

Import Rules for Project-Related Objects

Facets assigned to Supervision projects are related to instances of the application and mappings of project executables are related to entities of the topology. For the Supervision project import to complete entirely, these project-related objects need to be present in the system in which you want to import the project.

The table describes the impact on the import operation when project-related objects are not present or have a different configuration. It also describes the corrective action.

Project-related object that is not present or different	Impact if not present in the target system	Required action
The instances whose facets are assigned to the Supervision project that you are importing (same identifier, template identifier, template version, and status Valid).	<p>You can import the project but not the tag and page containers to which such facets are assigned. This requires that the project contains at least one container that can be imported, page 317, even empty.</p> <p>A dialog box opens, page 319 asking you to confirm the partial import.</p>	<p>Do either of the following first to create missing instances:</p> <ul style="list-style-type: none"> • Use the export/import functionality, page 219 of the Application Explorer to import missing instances. • Create instances. This may require importing templates first. <p>You can obtain information about the missing instances and their templates in the tooltip of the container that is tagged with a red dot.</p>
An element of the instance creating a facet that is being imported is not selected.	<p>The facet that is normally created by this element is not imported.</p> <p>The notification panel provides information about the facet that is not imported.</p> <p>For example, you import a Supervision project to which the data facet created by the Maintenance element of an instance is assigned but in the application of the target system, the Maintenance element of the same instance is not selected.</p>	Edit the instance in the Application Explorer and select the corresponding element.
The properties of an instance are different from the instance configuration at the time of export. The difference lies in the element selection (services) and/or the parameter values, page 176.	The assignment status of facets that are imported is set to Out Of Date .	Generate the project or container to apply the changes.
Topological entities to which you have mapped the executable of the project (service and/or communication) that you are importing (same identifier, same devices (for example, NIC for a station node), and services).	You can import the project but not the Supervision project executable and any of its associated mappings.	Use the export-import functionality, page 552 of the Topology Explorer first to create the missing topological entities.

Import Rules for Equipment

When you import a Supervision project and a project with the same identifier already exists in the target system, for an instance that exists in the same cluster in both the imported file and the Supervision project, the equipment, page 425 that is already generated in the Supervision project prevails. This applies independently of the selection of the **Constituent File**, page 315 of the tag container.

Example:

Identifier of instance present in imported file and Supervision project	Equipment name generated by the imported instance	Equipment name generated by the instance in the Supervision project	Equipment in the Participant project after import
MyMotor_1 ⁽¹⁾	Folder_1.MyMotor_1 ⁽²⁾	Folder_2.MyMotor_1 ⁽²⁾	Folder_2.MyMotor_1
(1) The instance is located in <i>Folder_1</i> .			
(2) Facets of the instance are assigned to <i>TagContainer_1</i> in <i>Cluster_1</i> .			

Import Rules for Pages

When you select to import a page that is contained in an export file, and a page with the same identifier exists in the target system, the contents of the existing page is overwritten with the content of the page that is in the export file.

Import Rules for Page Templates

page templates, page 346 that you import appear in the **Page Templates** section of the Supervision project settings, page 292.

The page template that was selected as default template in the exported Supervision project becomes the default one after import.

When you select to import a page template and a page template with the same identifier exists in the target system, the **Resolve Conflicts in Page Templates** window opens where you have the choice to import the page template or not (**Skip** action). Either choice has no impact on the imported pages that use this page template.

If you import it (**Create** action), a time stamp suffix in the format YYYYMMDDHHMMSS is added to the name of the imported page template.

Import Rules for Facets

The software applies import rules to facets if the system into which you are importing a Supervision project already contains a Supervision project with the same identifier.

The following are the general import rules that the software applies if facets with the same identifier contained in tag containers with the same identifier exist in both the target project and the export file that you are importing:

- If you **select the constituent file** of the tag container:
 - Facets from the export file overwrite facets of the target system and maintain the state they have in the export file.
 - Facets that exist only in the export file are imported into the target system and maintain the state they have in the export file.
 - Facets that exist only in the target system become **Assigned** and **Non Generated**. If the status of the facet is deleted, unassigned, or unlinked, the software removes the facet from the tag container after import.
- If you **unselect the constituent file** of the tag container:
 - Facets with the same identifier maintain the state that they have in the target system.

- Facets that exist only in the export file are imported into the target system and become **Assigned** and **Non Generated**.
- Facets that exist only in the target system maintain their state.

NOTE: The software changes the assignment status of any facet of an instance to **Out Of Date** if a property of an instance, page 176 is different from what it was at the time of export. This includes the location of the instance in the application. The notification panel provides information about the change of the assignment status after import is complete.

The following tables illustrate these import rules based on the different use cases. The tag container and facet names are examples, and represent tag containers and facets in a cluster of the project.

If you select the tag container **Constituent File** option.

Contents of export file selected for import		Item in Supervision project of target system		Result in Supervision project of target system after import
Tag container identifier is identical				
Tag container1	Facet1	Tag contain-er1	-	Facet1 from export file with status from export file.
Tag container1	Facet1	Tag contain-er1	Facet1	Facet1 from export file with status from export file.
Tag container1	Facet1	Tag contain-er1	Facet2	Facet1 from export file with status from export file. Facet2 with status non-generated. If the status of Facet2 is deleted, unassigned or unlinked, the facet is removed from the tag container after import.
Tag container identifier does not exist in target project				
Tag container2	Facet3	-	-	Tag container2 with Facet3 from export file with status from export file.
- The item does not exist				

If you do not select the tag container **Constituent File** option.

Contents of export file selected for import		Item in Supervision project of target system		Result in Supervision project of target system after import
Container identifier is identical				
Tag container1	Facet1	Tag contain-er1	-	Facet1 from export file with status non-generated.
Tag container1	Facet1	Tag contain-er1	Facet1	Facet1 from target system with same status.
Tag container1	Facet1	Tag contain-er1	Facet2	Facet1 from target system with status non-generated. Facet2 from target system with same status.
- The item does not exist				

Import Rules for Executables and Mappings

The software updates the **Build State**, page 634 of a **Built** executable after import depending on the scenario.

If the system into which you are importing a Supervision project does not contain a Supervision project with the same identifier, the state of **Built** executables that you import changes to **Out Of Date**.

If the system into which you are importing a Supervision project already contains a Supervision project and an executable with the same identifier, the software applies import rules to executables, service, and communication mappings. These rules are described in the following tables.

The table illustrates the import rules that apply to a **Built** executable based on different scenarios.

Contents of export file selected for import	Item in Supervision project of target system	Resulting built state in Supervision project of target system after import
Executable1	Does not exist	Out Of Date
Executable1	Executable1	Out Of Date
Executable1	Executable2	Executable1: Out Of Date Executable2: Out Of Date

The table illustrates the import rules that apply to mappings based on different scenarios.

Contents of export file selected for import	Item in Supervision project of target system	Result in Supervision project of target system after import
Service mapping	Exists	Service mapping from target system is not changed.
	Does not exist	Service mapping is created with the mapping information from the export file.
Communication mapping	Exists	Communication mapping from target system is not changed.
Communication mapping	Does not exist	Communication mapping is created with the mapping information from the export file.

Opening the Project Import Window

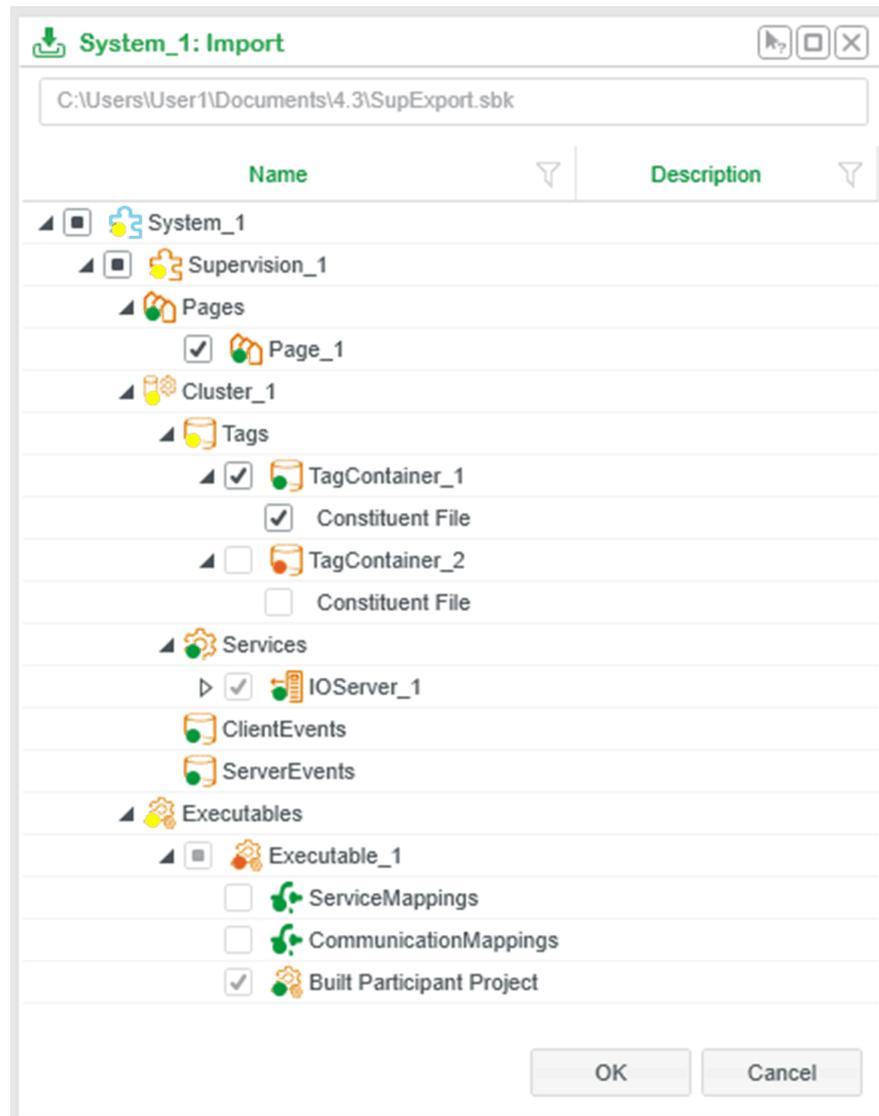
To open the **Import** window, proceed as follows.

Step	Action
1	Right-click the system root folder in a Project Browser of the Project Explorer and select Import . Result: The Import dialog box opens.
2	Browse to the export file (.sbk) that contains the project information that you want to import and click Open . Result: The Import window opens and displays the contents of the file that you have selected. NOTE: You can abort the command, page 319.

Project Import Window

The **Import** window lets you select the information of a Supervision project that is contained in an export file that you want to import into an existing system.

The following figure shows an example of the **Import** window displaying the contents of a Supervision project export file. In this example, the check boxes for *TagContainer_2* and relate mappings are cleared because the items cannot be imported. To indicate that at least one item of the Supervision project is not selected, the check box of parent nodes are filled with a square.



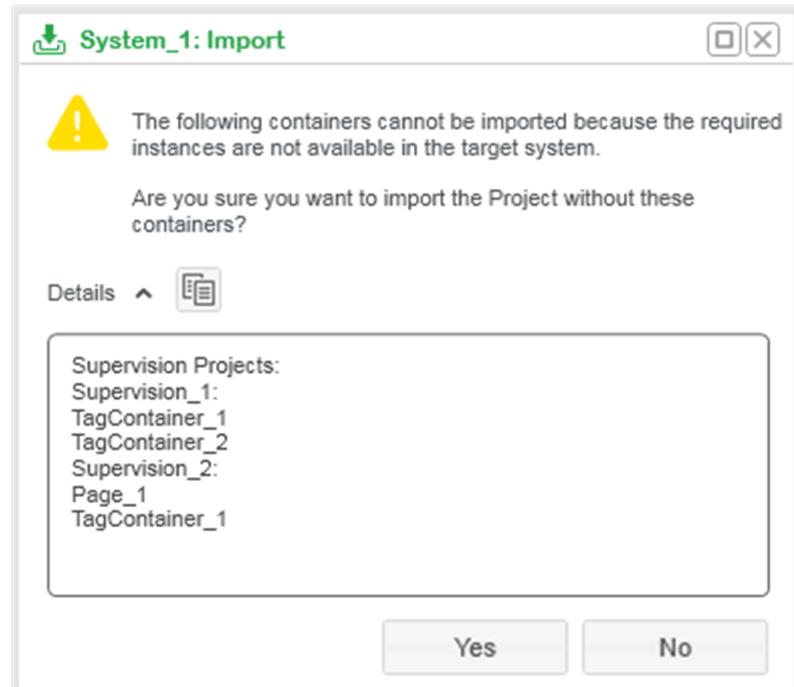
The table indicates the meaning of the colored dots that appear in the **Import** window next to items of a Supervision project.

Color code	Description
Green	You can import the item.
Red	<ul style="list-style-type: none"> At the container level: The container (for example, the tag container or the page) contains at least one facet that is referenced by an instance, which is not present in the application, page 314. At the executable level: <ul style="list-style-type: none"> An executable with the same identifier already exists in the target project. A project-related topological object, page 314 is not present in the target system. <p>A tooltip provides additional information about the issue. NOTE: You cannot import items that are tagged with a red dot.</p>
Yellow	<p>The item contains at least one subitem that is tagged with a red dot.</p> <p>The item can be imported partially only.</p>

NOTE: To expand/collapse a node in the **Import** window, select the node and press **Enter**.

Confirmation Dialog Box for Containers That Cannot Be Imported

The following figure shows an example of the dialog box that opens when a container of a Supervision project selected for import is tagged with a red dot and you click **OK** in the **Import** window. If you are also importing Control projects, containers are grouped by Participant then by project.



Button	Description
Yes	Closes the dialog box, the Import window, and imports the selected projects without containers tagged with a red dot.
No	Closes the dialog box and reverts to the Import window without importing projects. You can modify the selection of projects or cancel the import operation.

Aborting Import Tasks

After confirming the import by clicking **OK** in the **Import** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The project is not imported.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are importing a small project, the task may complete before you are able to click the icon.

Importing a Supervision Project

Included projects contain data related to Supervision constituents (for example, genies, tags, and so on), which can be shared by several templates.

Replacing the existing included project has an impact on:

- The templates that use the contents of the included project.
- The Supervision Participant projects to which the included project is associated.

NOTICE

LOSS OF DATA

Resolve an included project conflict that arises during the import of a Supervision Participant project by selecting the correct version of the included project to be imported.

Failure to follow these instructions can result in unusable Supervision projects.

To import data of a Supervision project into an existing system, proceed as follows.

Step	Action
1	In the Import window, page 317, select the items that you want to import.
2	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> • Applies the import rules based on your import selection. • If the applicable rules are satisfied, it proceeds with the import; If it detects a conflict with the name of an included project, it displays the Resolve Content Conflict dialog box, page 313. • Displays a summary of the completed actions and detected incompatibilities in the Notification Panel. <p>NOTE: Click Cancel to close the Import window without importing project information.</p>
3	<p>If the software displays the Resolve Content Conflict dialog box, do one of the following:</p> <ul style="list-style-type: none"> • Select Version in export file to overwrite the existing included project with the one contained in the export file that you are importing. • Select Version in contents repository to keep the existing included project. <p>Click OK.</p> <p>Result: The software proceeds according to your selection.</p> <p>NOTE: Click Cancel to discard your selection in the Resolve Content Conflict dialog box and revert to the Import window.</p>

Participant Project Indicators

Overview

This topic describes the various indicators that appear on components of Control and Supervision Participant projects.

Participant Project Progress Indicators

Overview

Various icons give an indication of the steps that remain before the system engineering life cycle is completed up to the deployment stage for Control and Supervision projects.

The progress icon is not visible while the checked-out icon, page 326 is shown.

NOTE: The indicator icons are visible to any user who opens the Participant project.

Meaning of Icons

For Control and Supervision projects.

Icon	Condition description
(0% progress)	Either one applies: <ul style="list-style-type: none"> • 0% progress for all project containers. • At least one facet is candidate, page 333 for assignment.
(0% progress)	The progress of at least one container is not 100%.
(0% progress)	Either one applies: <ul style="list-style-type: none"> • The progress of all containers is 100% and no executable exists. • The project progress is 75% and executables are deleted.
(0% progress)	The progress of at least one executable is not 100%.
No icon (100% progress)	The progress of at least one executable is 100%.

For sections and containers.

Icon	Condition description
(0% progress)	Either one applies: <ul style="list-style-type: none"> • No assignment is done. • All assignments have been removed.
(0% progress)	At least one facet is Non Generated .
(0% progress)	At least one facet, page 411 is Out Of Date, Deleted, UnAssigned, Moved, Inconsistent, or Invalid and no facet is Non Generated .
(0% progress)	All facets are Generated but the analysis detected one or more errors, page 324.
No icon (100% progress)	All facets are Generated or Unlinked without detected errors during analysis.

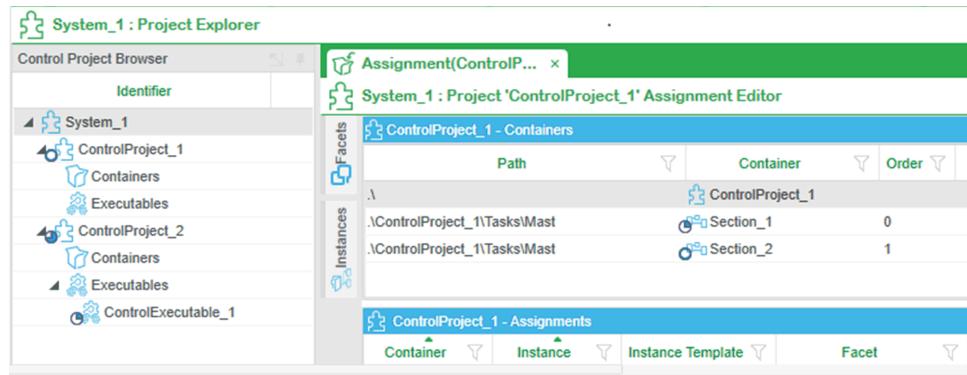
For executables.

Icon	Condition description
(0% progress)	Either one applies: <ul style="list-style-type: none"> • The executable is created. • Executable progress is 100% and service mapping, page 576 is removed.
(0% progress)	<ul style="list-style-type: none"> • Control: Service mapping is completed. • Supervision: Service mapping is done for at least one service.

Icon	Condition description
	Or, executable progress is 100% and service mapping is modified.
	<ul style="list-style-type: none"> Control: At least one communication mapping or hardware mapping is done. Supervision: Communication mapping is completed. And the executable state is Not Built .
	The executable state is Out Of Date or Built with alerts, page 327.
No icon (100% progress)	The executable state is Built .

How Icons Appear

The following figure shows an example of several Control projects and their components that have various completion states, which are indicated by progress icons.



Participant Project Validity Indicators

Overview

A dot or progress icon, page 322 that appears in red or yellow color indicates an alert or detected error, page 327 during the analysis of components of a Participant project.

When both an alert and the detection of an error occur for the same object, the detected error icon prevails.

When an icon appears on at least one component of a project (for example, a section, a container, an executable), it appears automatically at the project level.

The indicator icons are visible to any user who opens the Participant project.

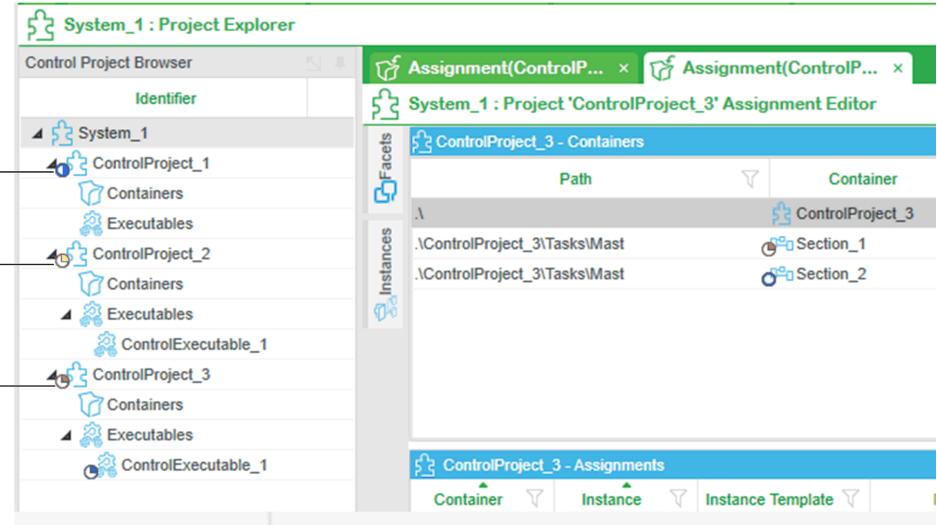
NOTE: The last action summary, page 327 can help provide information about issues that are not indicated by validity icons.

Meaning of Icons

Icon color	Level	Condition description
Yellow (progress icon or dot for objects with 100% progress) Example: 	Project	<ul style="list-style-type: none"> Control projects: The analysis, page 388 of the project or saving refinement changes, page 439 generated an alert indicating that the build operation may not complete successfully. Supervision projects: Saving changes generated an alert. All projects: At least one alert was generated for a project component.
	Section	The analysis, page 388 of the section or saving refinement changes, page 439 generated an alert indicating that the build operation may not complete successfully.
	Tag container or page	Generation completed with at least one alert.
	Page	Saving changes generated at least one alert.
	Executable	Build completed with at least one alert.
Red (progress icon or dot for objects with 100% progress) Example: 	Project	<ul style="list-style-type: none"> Control projects: The analysis, page 388 of the project or saving refinement changes, page 439 detected an error, which prevents the build operation from completing successfully. Supervision projects: Saving changes detected an error. All projects: There is at least one detected error for a project component.
	Section	The analysis, page 388 of the section or saving refinement changes, page 439 detected an error, which prevents the build operation from completing successfully.
	Tag container or page	Generation completed with at least one detected error.
	Page	Saving changes detected at least one error.
	Executable	Build did not complete successfully.

How Icons Appear

The following figure shows an example of several Control projects and their components that have various validity states, which are indicated by the color of the progress icons.



Item	Description
1	Valid project with 50% progress.
2	Project with 25% progress and at least one alert.
3	The detected error at the section level is also shown at the project level.

Participant Project Write-Lock Indicator

Overview

A lock icon lets you see which Participant project components are being used or modified by other processes or users and that you cannot work on them.



Certain components that are locked can be opened in read-only mode.

NOTE: The indicator icon is visible to any user who opens the Participant project.

Meaning of the Icon

The icon indicates that the project or a component is locked because an operation is in progress, such as:

- Assignment⁽¹⁾
- Generation
- Refinement
- Editing
- Mapping⁽¹⁾
- Export/import
- Build

(1) The lock icon appears very briefly while an performed is performed.

Where the Icon Appears

The lock icon can appear at the following levels:

- The Participant project
- Sections
- Containers (such as Supervision pages or tag containers)
- Clusters
- **Services** node and any of its subcomponents (such as servers, I/O devices)
- **ClientEvents** and **ServerEvents** nodes
- Executables

Viewing the Last Action Summary

Overview

The **Last Action Summary** command lets you view the last action that was performed on a given Participant project component and a summary of the result. The command is available in the context menu of projects and their components.

For actions that had not completed successfully or with alerts, and/or detected errors, it shows again the information that was displayed originally.

Supported Actions

An action summary is available after you perform either action:

- Generation
- Regeneration
- Build

Particular Conditions

Last action performed	Result
Not a supported action.	The command is unavailable.
A supported action was aborted.	An action summary for aborted actions is not available. The action summary for the previous action remains available (if applicable).
Import of a Participant project.	The command is unavailable. The action summary that existed at the time of export is cleared. NOTE: The last action summary of the target project is preserved when you perform an incremental import. That is, when you import a project into a system in which a project with the same identifier already exists.

Assignment Stage

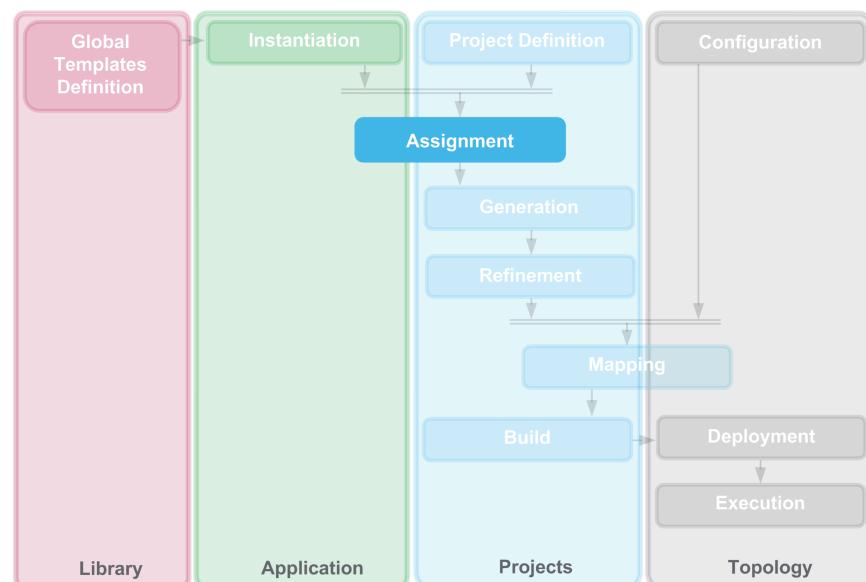
What's in This Chapter

Creating and Managing Sections, Instances, and Containers.....	329
Assigning Facets of Instances	349
Managing Assigned Facets	376

Overview

This chapter describes how to create Control and Supervision containers and assign facets of instances of the application to the corresponding projects that you have created during the Projects Definition stage.

The following figure shows the position of the **Assignment** stage within the system engineering life cycle.



Refer to the Assignment stage, page 44 for a description of the purpose of this stage.

Creating and Managing Sections, Instances, and Containers

Overview

This section describes the **Assignment Editor** and how to use it to create and manage Control sections and Supervision containers. It also describes section and container context menu commands and properties.

As part of the simplified workflow, page 57, it describes how to create instances in-place.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Assignment Editor

Overview

The **Assignment Editor** lets you create and manage Control and Supervision containers and assign the facets of instances of the application to them.

You can assign by facet, instance, folder, or entire application depending on the extent of the assignment, page 334 that you want to perform.

The editor gives you the flexibility to assign manually, automatically, or by combining both methods.

NOTE: As part of the simplified workflow, page 57, it lets you create and manage instances in-place by using context menu commands of FBD sections, page 335 and assigned Control facets, page 377.

The following table describes the two methods that you can use to assign facets to Control projects, page 362.

Section creation	Facet assignment	Prerequisite	How to use
Automatic, user-defined. With possibility to: <ul style="list-style-type: none">• Rename sections, page 359• Select which sections to create	Automatic ⁽¹⁾ , user-defined, with possibility to preview assignments, page 359.	Configure assignment mechanism, page 350.	Assign objects by dragging them to the project.
Manual	Manual	Create sections, page 335	Assign objects by dragging them to a section.
(1) By default, remaining candidate facets are assigned automatically, page 368 during generation			

The following table describes the methods that you can use to assign facets to Supervision projects, page 364.

Tag container creation	Facet assignment	Prerequisite	How to use
Automatic With possibility to: <ul style="list-style-type: none">• Rename tag containers, page 359• Select the cluster	Automatic ⁽¹⁾ with possibility to preview assignments, page 359.	Create additional clusters, page 295 as needed.	Assign objects by dragging them to the project.
Manual	Manual	Create tag containers, page 340 in the Assignment Editor and additional clusters, page 295 (as needed).	Assign objects by dragging them to the tag container.
	Automatic, page 370	You have associated a Control project to the tag container.	Assigning instances to the Control project automatically assigns the corresponding Supervision facets to the tag container.
(1) By default, remaining candidate facets are assigned automatically, page 368 during generation			

Opening the Assignment Editor

In a Control or Supervision project browser of the **Project Explorer**, double-click the **Containers** node, page 258 of the project to which you want to assign facets.

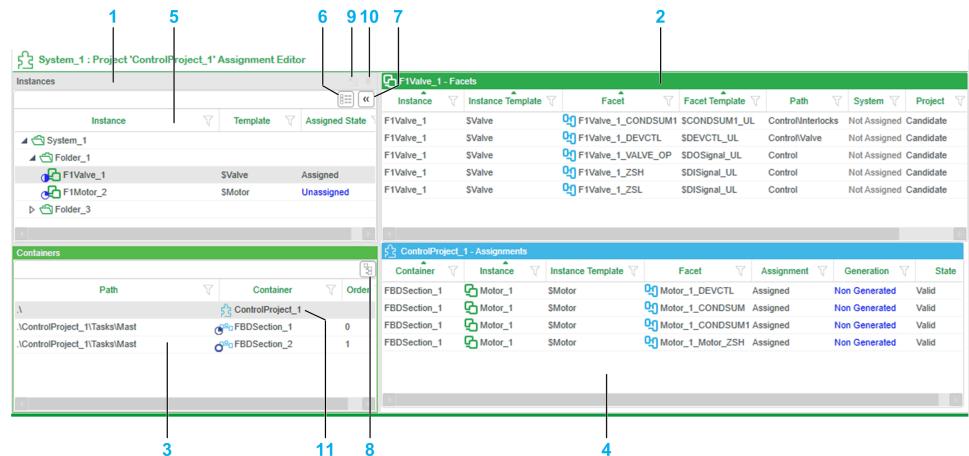
NOTE: You can also open the **Assignment Editor** by using the **View Assignments**, page 379 window.

Assignment Editor

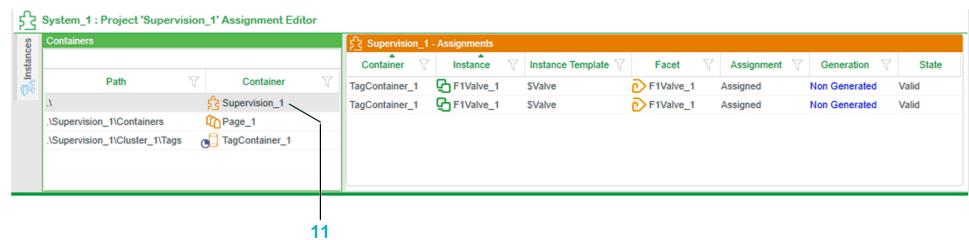
The **Assignment Editor** for Control and Supervision projects is identical except for the color scheme of the **Containers** and **Assignments** pane:

- Blue: Displays facets, containers, and assignments of Control projects.
- Orange: Displays facets, containers, and assignments of Supervision projects.

The following figure shows an example of the **Assignment Editor** for Control projects.



The following figure shows an example of the **Assignment Editor** for Supervision projects. The **Instances** and **Facets** panes are minimized to a tab by unpinning the **Instances** pane.



Item	Description
1	Instances pane. Shows the compatible instances of the application that have a valid Data and Link status.
2	Facets pane. Shows Participant-specific facets, which are referenced by the items that are selected in the Instances pane. The pane is empty by default. Supervision genie facets are not displayed.
3	Containers pane. Shows the project and its sections or containers. This pane is used to select the target to which you want to assign facets.
4	Assignments pane. Shows facets that are assigned to the project or container that is selected in the Containers pane. The title bar of the pane is where you drag objects to assign them.
5	You can select which columns, page 333 you want to see in each pane.
6	Button to toggle between grid and tree view in the Instances pane.
7	Button to show or hide facets, page 334 in the Facets pane. The Instances pane must be pinned.
8	Button to toggle between grid and tree view in the Containers pane, page 333.
9	Button to show or hide, page 111 the Instances and Facets panes
10	Button to extract or retract the Instances pane.
11	Project node, which gives you access to project-level context menu commands.

The following tables describe the information that appears in the panes of the **Assignment Editor**.

Instances pane.

Column header	Description
Instance	Identifier of the instance.
Template	Identifier of the template that the instance is using.
Version	Version of the template that the instance is using.
Assigned State	<ul style="list-style-type: none"> Assigned: At least one Control or Supervision facet of the instance is assigned to a project of the system. Unassigned: No facet of the instance is assigned to any project of the system.
Path	<p>Path to the instance in the tree view of the Application Browser.</p> <p>The column appears only if you have selected the grid view display mode.</p>
Candidate	Provides information about the assignment of facets at the instance level: <ul style="list-style-type: none"> Done: The facets of the instance are already assigned to this project. Candidate: The instance contains at least one facet that is not yet assigned to this project.
Description	Contents of the <i>\$Description</i> parameter of the instance.

Facets pane.

Column header	Description
Instance	Identifier of the instance that references the facet.
Instance Template	Identifier of the template that the instance that references the facet is using.
Facet	Identifier of the facet.
Facet Template	Identifier of the template that the facet is using.
Path	Path to the facet as it is shown in the Assignment Viewer , page 379.
System	Indicates the assignment ⁽¹⁾ status of the facet at the system level: <ul style="list-style-type: none"> Assigned: The facet is already assigned to a project of the system. Not assigned: The facet is not yet assigned to a project of the system.
Project	Indicates the assignment status of the facet at the project level: <ul style="list-style-type: none"> Candidate: You have not yet assigned the facet to this project. Done: You have already assigned the facet⁽¹⁾.
Type	Provides information on the subtype of the facet.

(1) Generating the section that contains the facet may change this status depending on the actions that you have executed on the facet (for example, unassigning it after a first generation). The **Assignment** column of the **Assignments** pane provides you with a more detailed information on the assignment status of the facet.

Containers pane.

Column header	Description
Path	<p>For a Control project, indicates whether the section belongs to the MAST or FAST task folder.</p> <p>For a Supervision project, indicates to which cluster the container belongs.</p>
Container	<p>Shows the identifier of the project and the sections or containers that it contains.</p> <p>An icon indicates the language of a section or the type of container.</p> <p>NOTE: Although Supervision pages appear, you cannot assign genie facets, page 494 to them by using the Assignment Editor.</p>
Order	<p>For Control projects, indicates the order in which the software generates the sections.</p> <p>You can change the order of sections, page 266.</p> <p>For more detailed information on how the software manages the section order, refer to the generation process description, page 390.</p> <p>NOTE: The order is not shown in tree view mode.</p>

For a description of the **Assignments** pane, refer to the topic *Assigned Facet Description*, page 366.

Customizing Column Display

In the various panes of the editor, you can choose which columns you want to see by right-clicking a column header and selecting **Customize**. Certain columns are always shown.

When you hide a column for which a filter is applied, the filter is cleared.

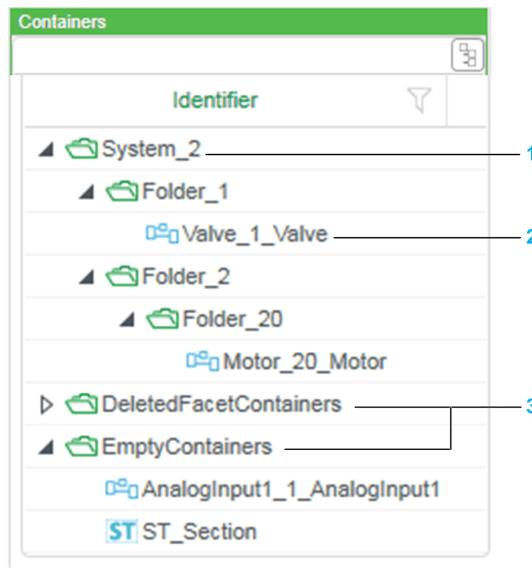
If tree and grid view are available, you may be able to configure the column display independently in each view.

Displaying Sections in Tree View Mode

The tree view mode allows you to view sections to which facets are assigned in relation to the folder structure of the application.

For more details, refer to the topic describing the representation of the application structure in Control projects, page 371.

The figure shows an example of the **Containers** pane in tree view mode once facets have been assigned to some FBD sections.



item	Description
1	System root folder and subfolders representing the folder structure of the application. NOTE: Predefined folder icons, page 170 that you may have assigned in the application to various folder levels are not represented.
2	Section identifier.
3	Special root folders.

Viewing Instances with Facets That Can Be Assigned

To view the instances of the entire application, which have facets that can be assigned to the project, proceed as follows.

Step	Action
1	Switch to grid view mode by clicking the button in the Instances pane.
2	Click the button to show the candidate column in the Instances pane. Result: The Instances pane displays the instances of the entire application. Those that have a status of Candidate in the Candidate column have facets that can be assigned to this project.

NOTE: Use the instance progress icons, page 189 to see which instances have facets that need to be assigned.

Viewing Facets That Can Be Assigned

To view the facets of the entire application that can be assigned to the project, proceed as follows.

Step	Action
1	Switch to tree view mode by clicking the button in the Instances pane.
2	In the Instances pane, select the system root folder.
3	Click the button to show facets in the Instances pane. Result: The Facets pane displays the facets of the entire application. Those that have a status of Candidate in the Project column can be assigned to this project.

NOTE:

- To view only the facets with status **Candidate**, use the filter on the **Project** column.
- To narrow the display to part of the application, select an application folder instead of the system root folder.

Object Selection

The following table describes the scope of the assignment depending on the objects that you drag from the **Instances** or **Facets** pane of either **Assignment Editor**. Each editor assigns only Participant-specific facets.

Object	Resulting assignment
One or more facets of the same or several instances.	These facets only. Remaining facets are assigned automatically, page 368.
One or more instances.	The facets of all the selected instances.
One or more folders.	The facets of all the instances that are contained in the selected folders. By dragging the system root folder, you can assign facets of the entire application at once.

NOTE: If the facet that you are trying to assign is not compatible with the selected container, you cannot create the assignment and a tooltip indicates it. If you are trying to assign multiple items at once and some of them are not compatible, the assignment is created for the compatible ones and information about the other ones is displayed in the notification panel.

Creating Control Project Sections

Overview

In the **Assignment Editor**, you can create only FBD sections in a Control Participant project.

This topic describes how to create sections manually. Sections can also be created automatically, page 350.

To create non-FBD sections, refer to the topic describing the **Project Explorer**.

Creating FBD Sections

To create an FBD section in the **Assignment Editor**, proceed as follows.

Step	Action
1	<p>Right-click the identifier of the Control project in the Containers pane and select Create FBD Section.</p> <p>Result: The Create FBD Section dialog box opens.</p>
2	<p>Edit the identifier, page 336 of the section if required, select the MAST or FAST task folder, and click OK.</p> <p>Result: A new FBD section is created with the next available order value, page 266.</p>

Inserting FBD Sections

To insert an FBD section, proceed as follows.

Step	Action
1	<p>In the Containers pane of the Assignment Editor, right-click the section where you want to insert the new one and select Insert FBD Section.</p> <p>Result: The Insert FBD Section dialog box opens.</p>
2	<p>Edit the identifier, page 336 of the section and click OK.</p> <p>Result: The newly created section is inserted and has the order, page 266 of the section that you had right-clicked. The order of the section that you right-clicked and those that follow is incremented by 1.</p>

Section Actions

The table describes the context menu commands of sections.

Command	Description
Insert FBD Section	Creates a new FBD section, page 335 at the position of this section.
Paste Instances	<p>Lets create a copy of instances that have been added to the Clipboard by using the Copy Instances facet context menu command, page 377.</p> <p>The pasted instance has the following properties:</p> <ul style="list-style-type: none"> • Its identifier is suffixed with <i>_X</i> (where <i>X</i> is an integer) • It has the same configuration as the copied instance. • It is created in the application in a folder named <i>HierarchyNotDefined</i>, which is in the root of the system. • Its facets are assigned to the section on which you have used the paste command. <p>NOTE: If the instance has Supervision data facets and the automatic assignment of Supervision facets, page 370 is enabled, these facets are assigned.</p>
Create Instance	Opens the Create Instance dialog box, page 338, which lets you select a template of the Global Templates library to create one instance and assign its facets to the section. The facet selection is based on the default Control service configuration of the template.

Command	Description
	You can also edit, link, and copy the instance, page 377. The command is only available for FBD sections.
Generate	Refer to <i>Generation Stage</i> , page 383. The command is only available for FBD sections.
Refine	Refer to <i>Refinement Stage</i> , page 435. The command is not available for LL984 sections.
ReGenerate	Refer to <i>Generation Stage</i> , page 421. The command is only available for FBD sections.
Delete	<p>Deletes the section.</p> <p>If you have assigned, page 329 facets to the section:</p> <ul style="list-style-type: none"> Unassigns, page 378 any facets contained in the FBD section that you are deleting. Changes the System status, page 411 of the facets to Not assigned. <p>NOTE: If the FBD section has been generated, deleting it also removes from the logical Control Participant project the section and constituents that were generated by the facets assigned to the section (for example, variables).</p> <p>However, derived types that were generated by these facets are not removed (for example, DFBs and DDTs). To remove these, you can either:</p> <ul style="list-style-type: none"> Regenerate, page 421 the Control project. This discards refinements. Purge the types from the Control Participant during refinement. For details, refer to <i>Purge of Unused Instances</i> in the Control Participant help. <p>NOTE: To make the command available for peer to peer communication sections, refer to the topic describing how to modify and delete peer to peer communication mapping, page 474.</p>
Rename	Lets you edit the identifier, page 336 of the section.
Properties	Opens the Properties window of the section.

Section Properties

To open the **Properties** window of a section, right-click the section and select **Properties**.

You can edit the following items.

Item		Description
General	Identifier	<p>Depending on how you create the section, the default identifier can be either:</p> <ul style="list-style-type: none"> FBDSection_n if you create the FBD section manually, using the Create FBD Section command, where <i>n</i> is an incremental number starting at 1. The concatenation of instance identifier _instance template identifier if the software creates the section automatically during assignment. <p>NOTE: For identifiers of Schneider Electric templates, the \$ prefix is omitted.</p> <p>The identifier must be unique within the Control project.</p> <p>The software allows only the use of identifiers that satisfy the naming rules, page 89.</p>
	Description	<p>Optional.</p> <p>You can enter a description of the section with free form text.</p>
	Generated On	<p>Date and time of the last successful generation or regeneration of the section.</p> <p>The property is read-only and contains information only for FBD sections.</p>

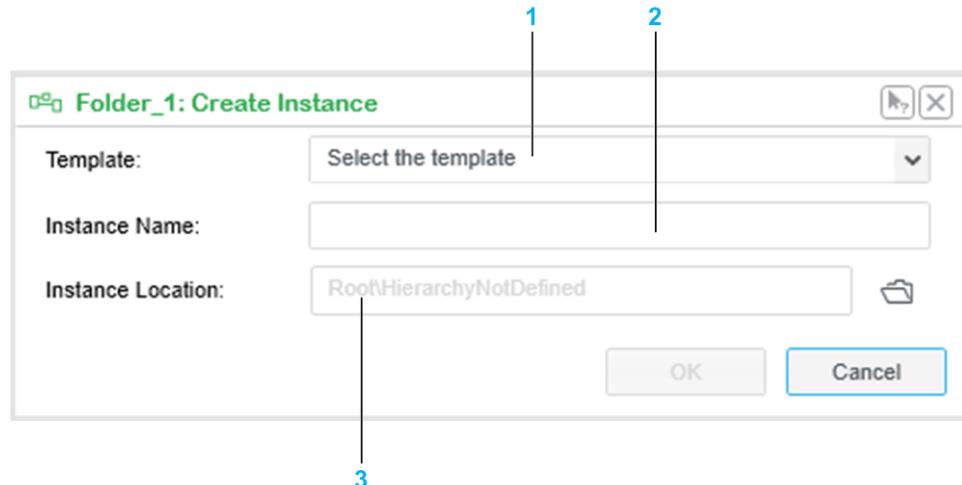
Creating Instances In-Place

Overview

As part of the simplified workflow, page 57, you can create and manage instances in Control projects from within the **Assignment Editor**.

Create Instance Dialog Box

The following figure shows an example of the **Create Instance** dialog box.



Item	Description
1	List of the composite and facet application templates that exist in the Global Templates library with usability state <i>Approved</i> .
2	Editable name of the instance, page 184 (<i>\$Name</i>).
3	Path in the application where the instance will be created. You can select an existing folder. By default, the instance is created in the <i>HierarchyNotDefined</i> folder, which is in the root of the system.

Creating Instances

Step	Action
1	In the Containers pane of the Assignment Editor , right-click an FBD section and select Create Instance . Result: The Create Instance dialog box opens.
2	In the Template menu, select the template to use to create the instance.
3	Edit the instance name. NOTE: If you select a different template, the instance name is reset.
4	Select a folder of the application where the instance is created.
5	Click OK . Result: <ul style="list-style-type: none"> The instance is created in the application with the default template configuration. Its Control facets are assigned to the section from which you have created the instance. NOTE: If the automatic generation setting, page 263 is enabled, the section is generated unless the software detects an error, page 389. NOTE: If the instance creates Supervision data facets and the automatic assignment of Supervision facets, page 370 is enabled, these facets are assigned. Follow the regular system engineering workflow to manage the Supervision project.
6	You can edit, link, and copy the instance, page 377.

Creating Supervision Tag Containers

Overview

This topic describes how to create tag containers manually by using the **Assignment Editor**, their properties, and context menu commands. The software creates tag containers automatically when you assign facets, page 330.

You need to create, at least, one tag container per controller to assign Supervision data facets of instances of the application.

These data facets represent the tags that are managed by Supervision components and assigned to the I/O device variables to be associated to the Supervision Participant project:

- Variable tags
- Trend tags
- Alarm tags

Monitoring data of a device, page 830 communicating by using the Modbus TCP protocol (or Modbus through an Ethernet gateway) from the Supervision infrastructure requires creating, in addition, one tag container for each device. Supervision *Data* facets representing device-specific data are assigned to each container. Also, a specific communication protocol needs to be defined for these tag containers.

Creating Tag Containers

To create a tag container in the **Assignment Editor**, proceed as follows.

Step	Action
1	<p>Right-click the identifier of the Supervision project in the Containers pane and select Create Tag Container.</p> <p>Result: The Create Tag Container dialog box opens.</p>
2	<p>Edit the identifier, page 341 of the tag container if required, select an existing cluster, and click OK.</p> <p>Result: A new tag container is created.</p>

Using a User-Defined Protocol

When you enter a protocol in the **Protocol** parameter of the tag container, page 341 instead of selecting a predefined one, verify the following:

- The protocol is supported by the platform of the controller that is the data source of the variable tags; otherwise, the variables cannot be read and/or written to by Supervision software.
- It is supported by the Supervision Participant and installed; otherwise, you cannot build the Supervision executable.

To verify which protocols are installed in the Supervision Participant, refine the Supervision Participant project and expand an entry in the **Protocol** column of **I/O Devices** in the **Topology** activity.

If the protocol is not installed, contact Schneider Electric support.

When using a user-defined protocol, some data generated at the template level, such as variable tag addresses may not be in the required protocol-specific format depending on the template configuration. After generating the Supervision Participant project, refine it and edit the data as needed. No configuration file (.xml) is generated when you build the Supervision Participant project.

NOTE: For communication with the I/O server, ensure that the protocol of the communication board, page 302 is correctly configured.

Tag Container Actions

The table describes the tag container context menu commands.

Com-mand	Description
Create Tag Container	Lets you create a new tag container in an existing cluster.
Generate, page 413	Starts the generation process to populate new tags, update out-of-date tags, and/or remove deleted tags in the Supervision Participant project.
Delete	<p>Deletes the tag container.</p> <p>If you have assigned facets to the container, the software:</p> <ul style="list-style-type: none"> • Unassigns the facets contained in the container that you are deleting. • Changes the system status of the facets to Not assigned. <p>NOTE: The software does not delete the servers that it may have created during the assignment of Supervision facets.</p> <p>NOTE: If the container has been generated, deleting it also deletes the data in the logical Participant project when you generate the project again.</p>
Rename	Lets you enter a new identifier for the tag container.
Proper-ties	Opens the Properties window, which allows you to edit the properties of the tag container.

Tag Container Properties

To open the **Properties** window of the tag container, right-click the container and select **Properties**.

You can edit the following items.

Item		Description
General	Identifier	<p>The identifier must be unique within the Supervision project.</p> <p>Default value: TagContainer_n where <i>n</i> is an incremental number starting at 1.</p> <p>NOTE: When Protocol for the tag container is set to OPCUA, the identifier is used as follows:</p> <ul style="list-style-type: none"> During generation, it becomes part of the variable tag address in the Supervision Participant project. The format of the address is <i>ns=2;s=0:<tag container ID>!<variable name></i>, where <i>ns</i> indicates the namespace and 2 corresponds to the OPC UA server as defined by Schneider Electric. For example, <i>ns=2;s=0:Controller1!InputVar1</i> During build, it becomes the alias for the device in the OFS configuration file, page 643. <p>Therefore, it may be useful to rename the tag container with the identifier of the controller that is mapped to the associated I/O device, page 624.</p>
	Description	<p>Optional.</p> <p>You can enter a description of the container with free text.</p>
	Generated On	Date and time of the last successful generation of the tag container (read-only).
Attributes	ScanTime	<p>The acquisition time for the tags that are assigned to the container.</p> <p>msec</p> <p>Default value: 0</p>
	Protocol	<p>The protocol that is used for communication with the I/O device.</p> <p>The menu contains the following values:</p> <ul style="list-style-type: none"> OFSOPC (default) OPCUA <p>NOTE: When using the OPC UA protocol with templates of a Schneider Electric library, clear the Use Native Types checkbox of the Unity/Control Expert symbols parameter (selected by default) so that the correct Control Participant types are used by the OPC UA server. For details, refer to <i>PLC Software Settings</i> in the OPC UA Server Expert help.</p> <ul style="list-style-type: none"> OPCUA Embedded: When the OPC UA server is embedded in a module (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) of the controller. Sets the protocol of the associated I/O device to OPCUA in the Supervision Participant project. User-defined protocol, page 340 whose name can be entered by using free-form text. 16 characters maximum. Values entered are added to the menu. <p>When you instantiate Schneider Electric templates and assign their Supervision facets to the Supervision project, only OFSOPC. OPCUA, and OPCUA Embedded are supported. This is because, by default, the facets generate Supervision data (variables and tags) that is specific to these protocols. To use another protocol, you can add the protocol-specific Supervision data to the project either by instantiating and configuring the \$VariableTag_CD template or by refining the Supervision project.</p> <p>NOTE: For each additional tag container that you create to monitor data of a device directly from the Supervision runtime, change the value to OPC.</p> <p>For information on the corresponding driver, refer to the driver help. To locate driver help files, look up <i>Driver Reference Help</i> in the help of the Supervision Participant, page 93</p>

Item		Description
Attributes	Direct TCP IP Addressing Mode⁽¹⁾	<p>Sets the addressing mode for the <i>TCP IP</i> driver in the OFS/OPC UA configuration file. Select the mode depending on the Modbus TCP topological entity, page 624 that is mapped to the associated I/O device in the Communication Mapping tab.</p> <p>The parameter is disabled when Protocol for the tag container is set to OPCUA Embedded.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Unallocated/Allocated Data, PAC Modicon (/U) (default): Corresponds to selecting <i>UNITY / Control Expert</i> in the PLCs section of the Device address wizard in the OFS/OPC UA configuration tool. Use this setting when mapping a controller or station node. • Allocated Data, Offset 0 (/T): Corresponds to selecting <i>PL7</i> in the PLCs section. • Allocated Data, Offset 0 with minimum communication (/NT): Corresponds to selecting <i>PL7</i> in the PLCs section and <i>Minimum communication profile</i>. • Allocated Data, Offset 1 (/Q): Corresponds to selecting <i>CONCEPT</i> in the PLCs section. • Allocated Data, Offset 1 with 32 bits swapping (/QW): Corresponds to selecting <i>CONCEPT</i> in the PLCs section and <i>Swap 32-bit Values</i>. • Allocated Data, Offset 1 (/NQ): Corresponds to selecting <i>MODBUS Device</i> in the PLCs section. <p>The value is available only when Protocol is set to OPCUA.</p> <p>NOTE: The value shown in brackets corresponds to the suffix that is appended to the device address in the configuration file.</p> <p>For more information, refer to:</p> <ul style="list-style-type: none"> • <i>Editing the Device Network Address</i> in the help of OPC Factory Server, page 101. • <i>Managing Device Aliases</i> in the help of OPC UA Server Expert.
Attributes	Supports Optimized Modbus Frames⁽¹⁾	<p>Corresponds to the <i>Modbus Request Optimization</i> parameter of Communication Settings in the OFS configuration tool.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • <i>Off</i> (default): The check box of the parameter is cleared. • <i>On</i>: The check box of the parameter is selected. <p>The parameter is not available for the OPCUA protocol.</p> <p>NOTE: Because in OFS, this parameter applies to all aliases of the OFS configuration file, if you configure several tag containers in a Supervision project and for one of them you select <i>Off</i>, the check box of the parameter in the OFS configuration file is cleared, overriding the <i>On</i> setting that you may have configured for any other tag container.</p>
Relationships	IoDevices	<p>Select the I/O device that you want to associate with the container.</p> <p>The I/O device appears in the menu once you have created it or assigned facets.</p> <p>Default value: <code>IODevice_1</code></p> <p>NOTE:</p> <ul style="list-style-type: none"> • By default, the software associates the tag container with the I/O device that it automatically creates when you assign Supervision facets to the container, page 364. If you create the I/O device manually, page 299, associate it with the tag container before or after assigning facets. • Changing the relationship after assigning facets to the tag container changes their assignment status to Out Of Date.
	Associated Control Project	<p>Lets you select one existing Control project that will trigger the automatic assignment, page 370 of Supervision facets to the tag container when you assign Control facets.</p> <p>You can associate a Control project to several tag containers given they belong to a different Supervision project.</p> <p>When the associated Control project is renamed, the value is updated. When it is deleted, the value is reset.</p> <p>Default value: Not Assigned</p>

Item	Description
(1)	The parameter becomes read-only when <i>Protocol</i> is different than OFSOPC or OPC UA.

Creating Supervision Pages

Prerequisites

- Generate the Supervision project before creating pages.
- If you want to use a specific style for the page, verify that it has been added in the Supervision project settings, page 292 and that it is selected as the default style.

Creating Supervision Pages

To create a page in the **Assignment Editor**, proceed as follows.

Step	Action
1	Right-click the identifier of the Supervision project in the Containers pane and select Create Page . Result: A new page is created.

Supervision Page Actions

The table describes the page context menu commands.

Command	Description
Create Page	Lets you create a new Supervision page.
Edit , page 495	Opens the Supervision Participant, which allows you to assign to the page animated graphics that are associated with the instances of the Supervision project. It also allows you to add graphic elements to the page to represent process hardware.
Generate , page 413	Starts the generation process to update out-of-date genies, and/or remove deleted genies in the Supervision Participant project.
Delete	Deletes the page including assignments of animated graphics.
Rename	Lets you enter a new identifier for the page.
Properties	Opens the Properties window, which allows you to edit the properties of the page.

Supervision Page Properties

To access the **Properties** window of the Supervision page, right-click the page and select **Properties**.

You can edit the following items.

Item		Description
General	Identifier	The identifier must be unique within the Supervision project. The software allows only the use of identifiers that satisfy the naming rules, page 89.
	Description	Optional. You can enter a description of the page with free form text.
	Generated On	Date and time of the last successful generation of the page (read-only).

Selecting Supervision Page Templates

Overview

By using the **Page Templates** section, page 292 of the Supervision project settings, you can add and remove page templates and select the template to be used by default for pages of the Supervision project.

The settings apply to each Supervision project of a system individually.

The following page templates are provided with the software:

- *Classic Templates*: Best suited for templates of the GPL Classic.
- *Situational Awareness*: Best suited for templates of the GPL.

Page templates that you want to add need to be contained in an archive file (.zip).

Creating Page Template Archive Files

To create an archive file (.zip) that contains a page template, proceed as follows.

Step	Action
1	Release (see EcoStruxure™ Process Expert, Supervision Participant Services User Guide) the Supervision Participant and open AVEVA Plant SCADA Studio outside of EcoStruxure Process Expert.
2	Create a new project, make it active, and open the Graphics Builder.
3	Click File > New... > Page .
4	In the Use Template dialog box, select a style, resolution, and a template to use for the page. Click OK . Result: The new page opens.
5	Save the page in the new project with the name <i>page</i> (Page text box) and click OK . NOTE: You will not be able to use the page template if you save it with a different name.
6	Close the Graphics Builder.
7	In the menu bar of AVEVA Plant SCADA Studio, click Backup . Result: The Backup Project dialog box opens.
8	In the Backup file text box, click Browse... , change the file extension to <i>zip</i> , click Open , and OK to backup the project as an archive in zip format.
9	By using the File Explorer, browse to the location where you have backed up the project and open the archive.
10	Delete all the files in the archive except for the following: <ul style="list-style-type: none"> • Page.ctf • Page.ctg • pgadv.DBF • pgbutton.DBF • pgdynobj.DBF • pgkey.DBF • pgtouch.DBF
11	Save the archive file.
12	To create another page, reopen the Graphics Builder and repeat steps 3 to 11. Ensure to save the new page with the name <i>page</i> , which replaces the page that you had created previously.

Adding Page Templates to a Supervision Project

Step	Action
1	Right-click the Supervision project and select Settings .
2	Select the Page Templates section.
3	Click the Add new template button. Result: The Browse Page Templates dialog box opens.
4	Select an archive file (.zip) containing the page template that you want to add and click Open . Result: The page template is shown in the Page Templates section.
5	Double-click the page template identifier and description to edit them.
6	Click OK to close the Settings window and Yes to save changes.
7	If the page template that you have added is based on an external included project that is not present in the content repository, add the included project to the Supervision project by using the Included Projects context menu command, page 494 of the project. Otherwise, if the page template is used by a page, you cannot build the Supervision project. NOTE: You only need to add the included project the first time you add a page template to the Supervision project of a system. Thereafter, when you add the page template (or another page template using the same include) to another Supervision project, the required included project is already available.

Using a Page Template

Step	Action
1	In the Page Templates section of the Settings window, select the page template that you want to use by selecting its Is Default property.
2	Click OK to close the Settings window and Yes to save changes. Result: Pages that you create from now on in this Supervision project use the page template. If you have added an external included project, it is shown when you use the Included Projects command, page 290 after you create the first page.

Removing Page Templates

To remove a page template, verify that its **Is Default** property is not selected in the **Page Templates** section and click the delete button in the **Action** column. Save changes when closing the **Settings** window.

You cannot remove the page templates provided with the software.

NOTE: Removing a page template does not impact existing pages that were created with this template and does not remove from the content repository the included project associated to this template.

Changing the Page Templates of Existing Pages

You cannot change the page template of an existing page by using the **Page Templates** section.

You need to edit the page and change the appearance manually.

Exporting and Importing Projects With Page Templates

Page templates are exported when you export a Supervision project and appear in the **Export** window.

For details on the import rules that apply to page templates, refer to the topic describing the import of Supervision projects, page 311.

Assigning Facets of Instances

Overview

This section describes how to use the **Assignment Editor** to assign facets of instances to their respective Control and Supervision containers. It also details the information that the software provides on the resulting assignments.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Defining the Control Facet Assignment Mechanism

Overview

The **Container Naming Rules** window allows up to four expressions and lets you use either one to define the assignment mechanism of Control facets, which includes:

- The name of sections that are automatically created when you assign facets.
- Which facets are assigned to which section.

Expressions are project-specific.

By default, one preconfigured expression, page 352 is selected in each project.

The expression is applied when you drag instances or folders containing instances to the Control project in the **Assignment Editor**.

If the resulting container identifier does not satisfy applicable naming rules, it is indicated, page 360.

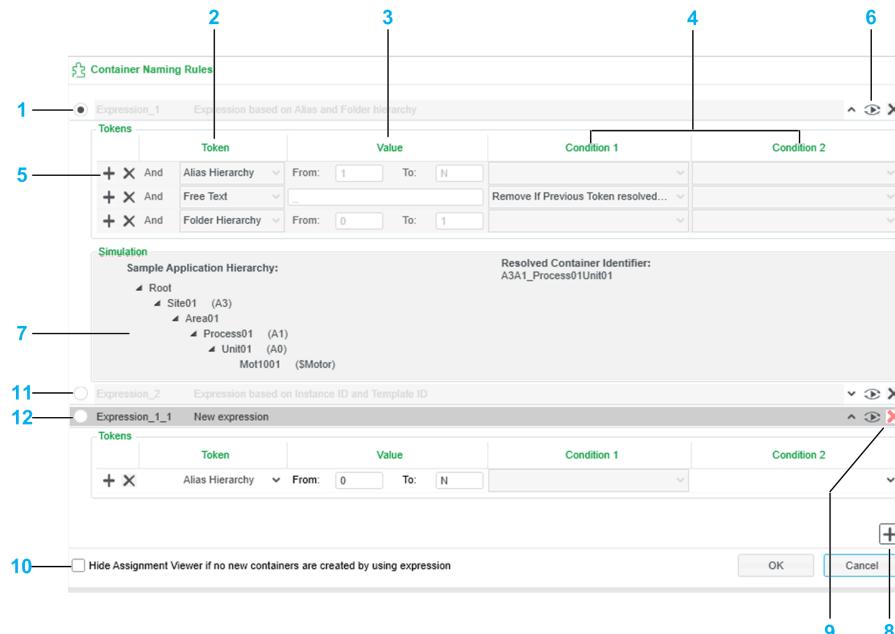
NOTE: You can preview and modify, page 359 the result or override the mechanism by assigning facets to an existing section.

Opening the Container Naming Rules Window

To open the **Container Naming Rules** window, select **Settings** in the context menu, page 260 of the Control project.

Container Naming Rules Window Description

The following example shows the **Container Naming Rules** window for a new Control project with its two preconfigured expressions of which the first one is selected.



Item	Description
1	Preconfigured expression, page 352 (read-only). Select the expression to use it to create sections and assign facets. You can copy expressions, page 353 and edit the name and description of expressions that you add.
2	To create the expression, you can configure up to six tokens. To create the section identifiers, the software concatenates the values that it has calculated for each of the configured tokens.

Item	Description
	<p>The tokens that you include in the expression have an impact on the number of sections, page 352 that are created.</p> <p>The available tokens are:</p> <ul style="list-style-type: none"> • Alias Hierarchy: Corresponds to the alias of the folder, page 166. • Folder Hierarchy: Corresponds to the identifiers of folders that make up the path of the instance. • Free Text: Free-form text that you can enter. It lets you add a fixed value to all sections that are created or insert a separator. Underscore is the only special character that is allowed. • Instance ID: Corresponds to the identifier of the instance, page 171. • Template ID: Corresponds to the identifier of the template, page 171 that is used by the instance.
3	<p>For Alias Hierarchy and Folder Hierarchy, you can select which levels in the path of the instance to consider.</p> <p>0 corresponds to the folder that contains the instance. 1 corresponds to its parent folder, and so on.</p> <p>N corresponds to the highest folder in the structure. The system root folder is excluded (except for instances that have the system root folder as direct parent).</p> <p>Maximum value: 99</p> <p>Example:</p> <p>For an instance located in <i>System_1\Site_1(S1)\Area_1(A1)\Cell_1(C1)\Unit_1(U1)</i> (where the value in parentheses represents the folder alias):</p> <ul style="list-style-type: none"> • 0 = <i>Unit_1</i> (or its alias, <i>U1</i>) • 1 = <i>Cell_1</i> (or its alias, <i>C1</i>) • 2 = <i>Area_1</i> (or its alias, <i>A1</i>) • N = <i>Site_1</i> (or its alias, <i>S1</i>) <p>Configuring the value as 1 to 2 for each token generates the following result:</p> <ul style="list-style-type: none"> • For Folder Hierarchy: <i>Cell_1Area_1</i> • For Alias Hierarchy: <i>C1A1</i>
4	<p> Optionally, you can define up to two conditions, page 353 for each token (except for the first and last token, which support only one condition each).</p> <p>The following conditions are available:</p> <ul style="list-style-type: none"> • Condition 1: <ul style="list-style-type: none"> ◦ Remove if previous token resolved value is empty ◦ Remove if all previous tokens resolved value is empty ◦ Remove if previous token resolved value is NOT empty ◦ Remove if all previous tokens resolved value is NOT empty • Condition 2: <ul style="list-style-type: none"> ◦ Remove if next token resolved value is empty ◦ Remove if all next tokens resolved value is empty ◦ Remove if next token resolved value is NOT empty ◦ Remove if all next tokens resolved value is NOT empty
5	Buttons to add and remove tokens in an expression.
6	<p>Lets you simulate the resulting section identifier by using the expression that is configured based on a sample folder structure and instance identifier if you were to assign any part of this sample hierarchy to the Control project in the Assignment Editor.</p> <p>To refresh the result after modifying the expression, click the button again.</p>
7	<p>Pane showing the sample folder structure and the resulting section identifier.</p> <p>Values in bracket represent the alias of the folder and the identifier of the template.</p>
8	<p>Lets you add a custom expression.</p> <p>You can add up to two expressions but only one expression can be selected at a time.</p>
9	You can delete expressions that you have added.
10	When selected, opens the Assignment Viewer window, page 359 only if the assignment creates at least one new section based on the selected expression.

Item	Description
11	Additional preconfigured expression, page 352 (read-only).
12	User-created expression that can be configured and selected.

Container Creation Principle

The table describes the number of sections, page 356 that are created depending on the token that the expression contains. When several tokens are included, the one with the lowest granularity determines the number of sections. Tokens are listed in increasing order of granularity.

Token	Number of sections created
Instance ID	As many sections as there are instances in the object that you are dragging.
Folder Hierarchy⁽¹⁾	As many sections as there are folders containing instances in the object that you are dragging (when Value = 0 to N). This token lets you assign facets of instances located in several folders of a same level, page 357 to one section.
Alias Hierarchy⁽¹⁾	As many sections as there are folders that have an alias and that contain instances (instances of subfolders included) in the object that you are dragging. This token lets you assign facets of instances located in several folders of different levels, page 358 to one section.
Template ID	As many sections as there are templates used by instances in the object that you are dragging. This token lets you assign facets of several instances that use the same template, page 355 to one section.
(1) The value that you configure impacts the number of sections that are created and the section identifier	

NOTE: The conditions, page 353 that are configured may impact the results.

Preconfigured Expressions

The following preconfigured expressions exist and are read-only. The section identifier that they create is composed of three parts where each part is the result of a calculated token.

To use a different expression, create one and select it.

For **new** Control projects:

The expression creates a section for each folder containing instances.

Example:

Alias1Alias2Alias3_Folder3Folder4

The table describes how the tokens of the expression are used to calculate the different parts of the section identifier.

Token	Value	Result
Alias Hierarchy	From 1 to N	Concatenation of the folder aliases from the highest level folder down to the level-1 folder in the path of the instance.
Free Text	_ (underscore)	An underscore. NOTE: The condition removes the underscore if no aliases have been configured because a section identifier cannot start with an underscore.
Folder Hierarchy	From 0 to 1	Concatenation of the identifiers of the level-1 and level-0 folders in the path of the instance starting with the level-1 folder.

For new and migrated Control projects:

The expression creates a section for each instance. The section identifier is the same as the one that was automatically created when assigning facets by using earlier versions of the software.

It is selected by default for migrated projects.

Example:

MyMotor_1_Motor

The table describes how the tokens of the expression are used to calculate the different parts of the section identifier.

Token	Value	Result
Instance ID	n/a	The identifier of the instance.
Free Text	_ (underscore)	An underscore.
Template ID	n/a	The identifier of the template that is used by the instance (without \$ prefix).

Using Conditions in Expressions

You can use conditions to modify container identifiers. Conditions remove a token from an expression based on the value of one or all previous tokens and/or the value of one or all following tokens.

It lets you, for example, show the folder identifier instead of the folder alias in case no alias is defined.

Example of condition based on previous token:

You have configured an expression as `<token1><token2><token3>` and an underscore is defined as free text for `token2`.

If the value for `token1` is blank, the resulting container identifier starts with an underscore. To avoid that, you can configure the condition **Remove if previous token resolved value is empty** for `token2`. The container identifier is then only `token3`.

Example of condition based on next token:

You have configured an expression as `<token1><token2><token3>` and `Section_A` is defined as free text for `token1`. However, you want this text to appear only when `token2` and `token3` contain no value.

You can configure the condition **Remove if all next tokens resolved value is not empty** for `token1` to achieve this.

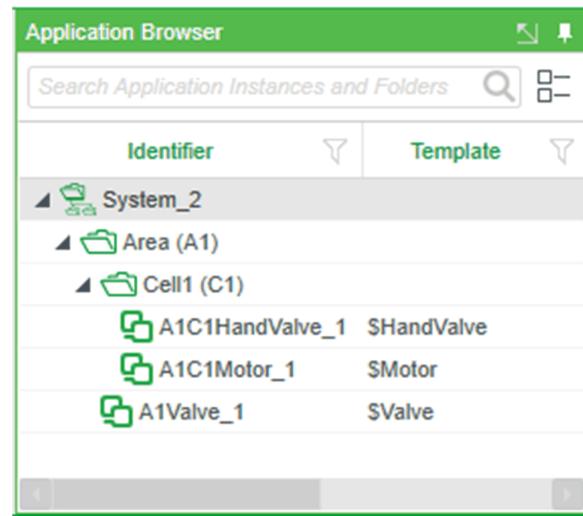
Copying Expressions

You can copy expressions to paste them in the same window or in the window of another project.

Step	Action
1	Right-click the header of the expression and select Copy .
2	Right-click the empty space below an expression and select Paste . Result: The expression is added to the Container Naming Rules window. NOTE: You can only paste as many expressions as is allowed to remain within the limit of expressions per project.

Sample Application for Examples

The following figure shows the sample application that is used in the following examples. In each one, **Area** is dragged to the Control project in the **Assignment Editor**



Example With Instance ID Token

Only **Instance ID** is configured as token without conditions.

Tokens	Token	Value	Condition 1	Condition 2
1.	Instance ID			
2.				
3.				

After dragging **Area**, one section for each instance is proposed. The section identifier is the instance identifier.

ControlProject_1 : Assignment Viewer					
Expression1 : Default Expression for the newly created project					
	Identifier	Facets	State	Path	
1. □ A1C1Motor_1	A1C1Motor_1	A1C1Motor_1_DEVCTL A1C1Motor_1_CONDSUM1	New	MAST	▼
2. □ A1C1HandValve_1	A1C1HandValve_1	A1C1HandValve_1_HVALVE A1C1HandValve_1_HVALVE_ZSH A1C1HandValve_1_HVALVE_ZSL	New	MAST	▼
3. □ A1Valve_1	A1Valve_1	A1Valve_1_DEVCTL A1Valve_1_ZSL A1Valve_1_ZSH	New	MAST	▼

Example With Folder Hierarchy and instance ID Tokens

Folder Hierarchy is added to **Instance ID** as tokens with an underscore as separator. No conditions.

Tokens	Token	Value	Condition 1	Condition 2
1.	Folder Hierarchy	From: 0 To: 1		
2.	Free Text	_		
3.	Instance ID			

Compared to the previous example, page 354, the number of sections remains the same. Only the identifier of the sections is changed by adding the identifier of the folder containing the instance (level 0) and its parent (level 1) as prefix.

ControlProject_1 : Assignment Viewer				
	Identifier	Facets	State	Path
▲	AreaCell1_A1C1Motor_1		New	MAST
	A1C1Motor_1	0 A1C1Motor_1_DEVCTL		
	A1C1Motor_1	0 A1C1Motor_1_CONDSUM1		
▲	AreaCell1_A1C1HandValve_1		New	MAST
	A1C1HandValve_1	0 A1C1HandValve_1_HVALVE		
	A1C1HandValve_1	0 A1C1HandValve_1_HVALVE_ZSH		
	A1C1HandValve_1	0 A1C1HandValve_1_HVALVE_ZSL		
▲	Area_A1Valve_1		New	MAST
	A1Valve_1	0 A1Valve_1_DEVCTL		
	A1Valve_1	0 A1Valve_1_ZSL		
	A1Valve_1	0 A1Valve_1_ZSH		

Example With Template ID Token

Instance *Motor_1* that is using template \$Motor is added to the application so that two instances of the same template exist.

Application Browser	
Search Application Instances and Folders	
Identifier	Template
▲ System_2	
▲ Area (A1)	
▲ Cell1 (C1)	
A1C1HandValve_1	\$HandValve
A1C1Motor_1	\$Motor
A1Motor_1	\$Motor
A1Valve_1	\$Valve

Only **Template ID** is configured as token without conditions.

Tokens	Token	Value	Condition 1	Condition 2
1.	Template ID			
2.				
3.				

One section for each template that is used in the application is proposed. The section identifier is the template identifier (without \$ character).

ControlProject_1 : Assignment Viewer					
Expression1 : Default Expression for the newly created project					
	Identifier	Facets	State		Path
Motor	A1C1Motor_1 A1C1Motor_1 A1Motor_1 A1Motor_1 A1Motor_1	A1C1Motor_1_DEVCTL A1C1Motor_1_COND_SUM1 A1Motor_1_DEVCTL A1Motor_1_COND_SUM A1Motor_1_COND_SUM1	New	MAST	MAST
HandValve	A1C1HandValve_1 A1C1HandValve_1 A1C1HandValve_1	A1C1HandValve_1_HVALVE A1C1HandValve_1_HVALVE_ZSH A1C1HandValve_1_HVALVE_ZSL	New	MAST	MAST
Valve	A1Valve_1 A1Valve_1 A1Valve_1	A1Valve_1_DEVCTL A1Valve_1_ZSL A1Valve_1_ZSH	New	MAST	MAST

Example With Template ID and Alias Hierarchy Tokens

Alias Hierarchy is added to **Template ID** as tokens with an underscore as separator. No conditions.

Tokens					
	Token	Value	Condition 1	Condition 2	
1.	Alias Hierarchy	From: 0 To: 2			
2.	Free Text	_			
3.	Template ID				

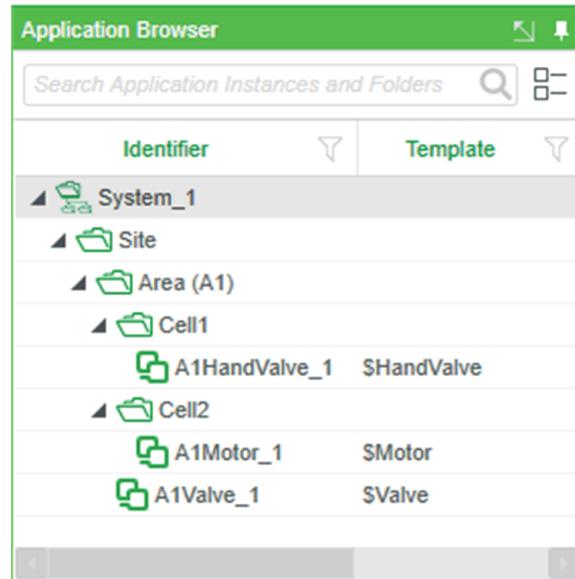
Compared to the previous example, page 355, one section has been added because the two instances of template \$Motor are located in folders that have a different alias. The section identifier is the template identifier with the aliases of Cell1 and Area added as prefix.

ControlProject_1 : Assignment Viewer					
Expression1 : Default Expression for the newly created project					
	Identifier	Facets	State		Path
A1C1_Motor	A1C1Motor_1 A1C1Motor_1	A1C1Motor_1_DEVCTL A1C1Motor_1_COND_SUM1	New	MAST	MAST
A1C1_HandValve	A1C1HandValve_1 A1C1HandValve_1 A1C1HandValve_1	A1C1HandValve_1_HVALVE A1C1HandValve_1_HVALVE_ZSH A1C1HandValve_1_HVALVE_ZSL	New	MAST	MAST
A1_Valve	A1Valve_1 A1Valve_1 A1Valve_1	A1Valve_1_DEVCTL A1Valve_1_ZSL A1Valve_1_ZSH	New	MAST	MAST
A1_Motor	A1Motor_1 A1Motor_1 A1Motor_1	A1Motor_1_DEVCTL A1Motor_1_COND_SUM A1Motor_1_COND_SUM1	New	MAST	MAST

NOTE: The same number of folders would have been created if **Folder Hierarchy** had been selected as token instead of **Alias Hierarchy**.

Example to Group Facets of Instances Located in Several Folders of Same Level

This example illustrates how to use the **Folder Hierarchy** token to assign to one section the facets of instances located in folders of a same level in the application (*Cell1* and *Cell2*). The following figure shows the sample application that is used.



Folder Hierarchy is configured as token with value 1 to 1. Because the value represents the parent folder, it creates a section for each folder that contains instances by using the identifier of its parent. When two folders have the same parent, only one section is created. No conditions.

Tokens	Token	Value	Condition 1	Condition 2
1.	Folder Hierarchy	From: 1 To: 1		
2.				
3.				

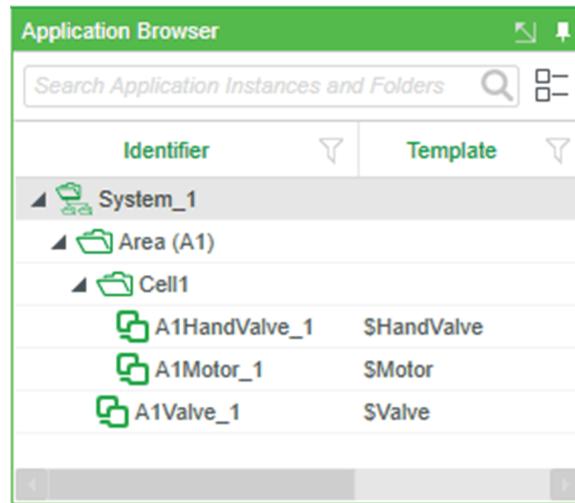
After dragging *Area* to the Control project, two sections are proposed. Section *Area* contains the facets of *A1HandValve_1* and *A1Motor_1* because it is the parent of *Cell1* and *Cell2*.

	Identifier	Facets	State	Path
<input checked="" type="checkbox"/>	Area	New	MAST	
<input checked="" type="checkbox"/>	A1HandValve_1	A1HandValve_1_HVALVE		
<input checked="" type="checkbox"/>	A1HandValve_1	A1HandValve_1_HVALVE_ZSH		
<input checked="" type="checkbox"/>	A1HandValve_1	A1HandValve_1_HVALVE_ZSL		
<input checked="" type="checkbox"/>	A1Motor_1	A1Motor_1_DEVCTL		
<input checked="" type="checkbox"/>	A1Motor_1	A1Motor_1_CONDSUM1		
<input checked="" type="checkbox"/>	Site	New	MAST	
<input checked="" type="checkbox"/>	A1Valve_1	A1Valve_1_DEVCTL		
<input checked="" type="checkbox"/>	A1Valve_1	A1Valve_1_ZSL		
<input checked="" type="checkbox"/>	A1Valve_1	A1Valve_1_ZSH		

NOTE: If the section identifier is not satisfactory, you can edit it in-place, page 360 or add a **Free Text** token to the expression.

Example to Group Facets of Instances Located in Several Folders of Different Levels

This example illustrates how to use the **Alias Hierarchy** token to assign to one section the facets of instances located in folders of different levels in the application (*Area* and *Cell1*). The following figure shows the sample application that is used.



Alias Hierarchy is configured as token with value 0 to 1. This creates a section for each folder that has an alias and contains instances. No conditions.

Tokens	Token	Value	Condition 1	Condition 2
1.	Alias Hierarchy	From: 0 To: 1		
2.				
3.				

After dragging *Area* to the Control project, one section is proposed, which contains the facets of the three instances of the application. *Cell1* contains instances but has no alias. *Area* contains an instance and has an alias. Thus, its alias is used to create the section identifier to which facets of *A1Valve_1* are assigned. Facets of any instances contained in subfolders (*Cell1*) are assigned there as well.

ControlProject_1 : Assignment Viewer					
Expression_1 : Default expression					
	Identifier	Facets	State		Path
1. <input checked="" type="checkbox"/> A1			New		MAST
	A1HandValve_1	A1HandValve_1_HVALVE			
	A1HandValve_1	A1HandValve_1_HVALVE_ZSH			
	A1HandValve_1	A1HandValve_1_HVALVE_ZSL			
	A1Motor_1	A1Motor_1_DEVCTL			
	A1Motor_1	A1Motor_1_COND_SUM1			
	A1Valve_1	A1Valve_1_DEVCTL			
	A1Valve_1	A1Valve_1_ZSL			
	A1Valve_1	A1Valve_1_ZSH			

NOTE: If the section identifier is not satisfactory, you can edit it in-place, page 360 or add a **Free Text** token to the expression.

Previewing Facets Assignments

Overview

The **Assignment Viewer** opens when you drag one or more facets to a project in the **Assignment Editor**.

It does not open if you drag facets to a container that you have created manually.

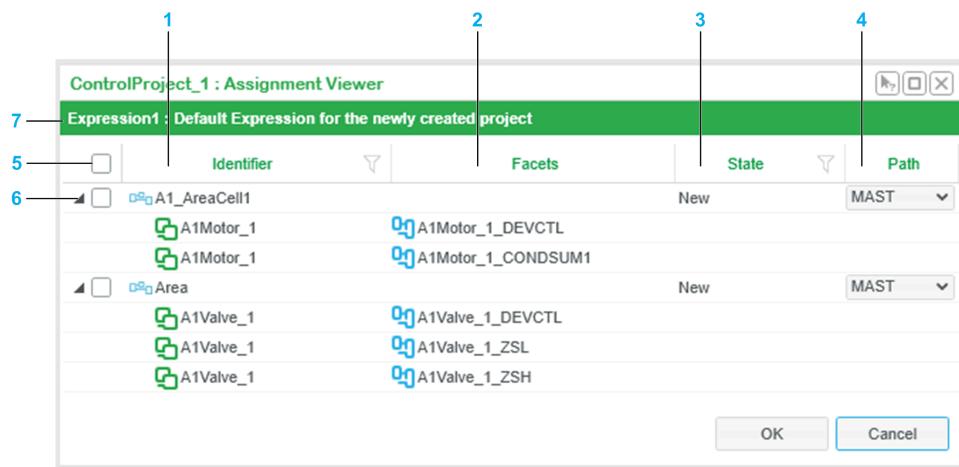
The window lets you:

- For Control projects, preview the identifier of newly created sections, which is the result of the expression that you have selected, page 350.
- Preview which facets will be assigned to which container.
- Edit the identifier, page 360 of containers in-place.
- Select in which task folder or cluster to create new containers.
- Select which assignments to perform.

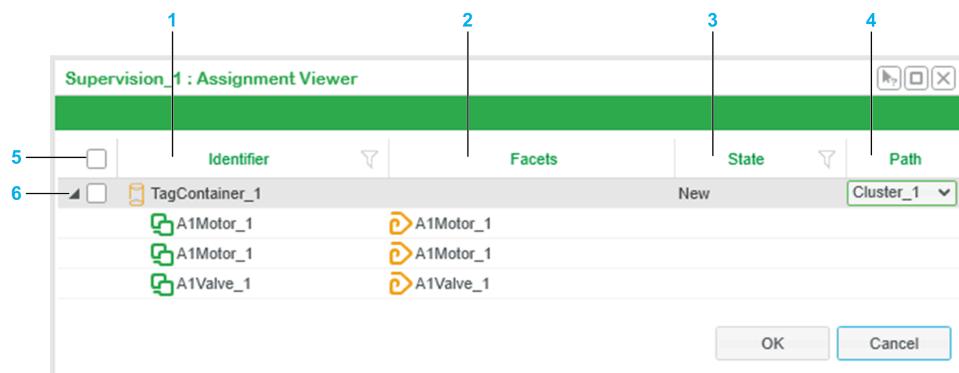
NOTE: For Control projects, you can configure it, page 350 to open only if the facets that you are assigning create a new section.

Assignment Viewer Description

The following figure shows an example of the **Assignment Viewer** when you drag the top level folder of a folder structure containing two instances⁽¹⁾ to a Control project while the default expression, page 352 for a new project is selected.

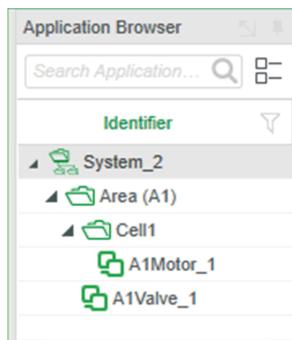


The following figure shows an example of the **Assignment Viewer** when you drag the top level folder of a folder structure containing two instances⁽¹⁾ to a Supervision project.



Item	
1	<p>Shows the identifier of the containers to which the facets of the instances that you have dragged will be assigned. For each container, shows the identifier of the instances whose facets will be assigned.</p> <p>By default, the order of containers and instances corresponds to the hierarchy of the application.</p> <p>For Control projects, the number of sections and their identifier depends on the expression that is selected, page 350 and the object that you have dragged (for example, an instance, a folder, or a folder structure).</p> <p>For Supervision projects, one tag container is created each time you drag an object to the project independently of the number of instances inside the object.</p> <p>You can edit the identifier of new sections.</p>
2	Identifiers of the facets that will be assigned to the container.
3	Indicates whether the container already exists in the project or if it will be created.
4	Indicates or, for new containers, lets you select: <ul style="list-style-type: none"> For Control projects: The task folder of the section. For Supervision projects: The cluster of the container.
5	Check box to select all the containers that appear in the window.
6	Select the check box to create the container (if new) and assign the facets. Containers whose check box is cleared are ignored.
7	For Control projects, indicates the name of the expression, page 350 that is selected.

(1) The following figure shows the application that was used for the two **Assignment Viewer** examples.



Editing Container Identifiers

You can edit the identifier of new containers only.

By editing the identifier of a container so that it is the same as an existing container or another container that is proposed in the **Assignment Viewer**, you can assign the facets of both to a single container.

Container Identifier Validation

Either of the following dots can appear on the container icon:

- A red dot to indicate that the container identifier that is the result of the expression is blank.
- A yellow dot to indicate that the container identifier has been modified to satisfy applicable naming rules, page 88 (for example, maximum length or allowed characters).

Modifying the Application While the Assignment Viewer is Open

If the application is modified while the **Assignment Viewer** is open and the changes impact any result of the expression that is selected, the information in the window is not updated. Also, when you proceed with the assignments, they are performed based on the information that is displayed in the window.

Assigning Control Facets of Instances

Before Starting

To assign Control facets of instances, you must have completed:

- The Instantiation stage, page 152
- The Control Project Definition stage, page 259

You can only assign facets of instances that have a valid Data and Link status, page 171.

You can assign facets only to FBD sections.

You cannot assign the same facet to more than one FBD section in the same project but you can assign the same facet to several Control Participant projects of the system.

NOTE: You can associate a Control project to a tag container to assign the corresponding Supervision facets automatically, page 370.

Facet Assignment Order

Depending on the method that you use to select instances for assignment, the order of the referenced facets varies. The following applies when you assign all facets to the same FBD section.

Assignment method	Facet order
Instance by instance	The order in which you drag instances to assign them defines the order of the referenced facets. Instances selected first for assignment have facets with the lowest order.
Multiple instances at once	The order in which you select instances defines the order of the referenced facets. Instances selected first have facets with the lowest order.
By folder	The order of instantiation defines the order of the referenced facets. Instances created first have facets with the lowest order.
Multiple folders at once	The order in which you select folders defines the order of the referenced facets. Folders selected first have facets with the lowest order. If the folders contain multiple instances, the order in which instances were created is not relevant anymore.

NOTE: You can change the order of facets, page 381.

Assigning Facets

This procedure describes how to assign facets of instances to sections of the Control project by using an expression, page 350.

Step	Action	Result
1	Verify that the instances are displayed in tree view mode in the Instances pane, else click the toggle button, page 330. NOTE: Alternatively, you can assign instances using the grid view mode. The assignment principle remains unchanged. You can switch between views at any time during assignment by clicking the toggle button.	The Assignment Editor displays the valid, page 171 instances of the application in tree view mode.
2	Configure the expression depending on the assignment mechanism that you want to use and select it.	–
3	In the Containers pane, select the Control project.	• The name of the project appears in the title bar of the Assignments pane.

Step	Action	Result
		<ul style="list-style-type: none"> Any existing sections and the facets already assigned to them are displayed.
4	<p>From the Instances pane, drag one or more objects, page 334 to the title bar of the Assignments pane.</p> <p>NOTE: Alternatively, you can drag objects directly to project in the Containers pane.</p>	The Assignment Viewer window, page 359 opens.
5	Review the information that is displayed, edit section identifiers as needed, and select the task folder for each section.	–
6	Click OK .	<ul style="list-style-type: none"> Sections that do not exist yet are created after existing sections. The Control facets that have a status of Candidate are assigned to the respective FBD sections and displayed in the Assignments pane. The software assigns an order, page 366 to each assigned facet. The status of the assigned facets in the System column of the Facets pane changes to Assigned if the status was Not assigned. The status in the Project column of the Facets pane changes from Candidate to Done.

NOTE: By default, remaining facets will be assigned automatically, page 368 when you generate the project or a section.

Assigning Supervision Facets of Instances

Before Starting

To assign Supervision data facets of instances, you must have completed:

- The Instantiation stage, page 152
- The Supervision Project Definition stage, page 288

You can only assign facets of instances that have a valid Data and Link status, page 171.

You cannot assign the same facet to more than one tag container in the same project but you can assign the same facet to several Supervision projects of the system.

NOTE: To assign Supervision genies facets to a page, page 345, refine the Supervision project.

Automatic Supervision Project Component Creation

Each time you drag an object from the **Instances** pane to the Supervision project, one tag container is created, page 359.

In addition, depending on the facets that you assign to the tag container, the software may automatically create and configure the following components under the **Services** node, page 291 of the Supervision project cluster:

- One I/O server including:
 - One I/O device
 - One communication board and its port
- One alarm and/or one trend server (if facets that create alarm and/or trend tags have been assigned).

NOTE: The I/O server and its components are created only for the first tag container of a cluster whether you create the tag container manually or if the software creates it automatically during the assignment. The other servers are created also when you assign corresponding facets to additional tag containers.

Assigning Facets

This procedure describes how to assign data facets of instances to a Supervision project.

Step	Action	Result
1	<p>Verify that the instances are displayed in tree view mode in the Instances pane, else click the toggle button, page 330.</p> <p>NOTE: Alternatively, you can assign instances using the grid view mode. The assignment principle remains unchanged. You can switch between views at any time during assignment by clicking the toggle button.</p>	The Assignment Editor displays the valid, page 171 instances of the application in tree view mode.
2	In the Containers pane, select the Supervision.	<ul style="list-style-type: none"> • The name of the project appears in the title bar of the Assignments pane. • Any existing containers and the facets already assigned to them are displayed.
3	<p>From the Instances pane, drag one or more objects, page 334 to the title bar of the Assignments pane.</p> <p>NOTE: Alternatively, you can drag objects directly to the project in the Containers pane.</p>	The Assignment Viewer window, page 359 opens.

Step	Action	Result
4	Review the information that is displayed, edit the container identifier as needed, and select the cluster.	–
5	Click OK .	<ul style="list-style-type: none"> Containers that do not exist yet are created. Supervision data facets that have a status of Candidate are assigned to the respective container and displayed in the Assignments pane. The status of the assigned facets in the System column of the Facets pane changes to Assigned if the status was Not assigned. The status in the Project column of the Facets pane changes from Candidate to Done.

NOTE: If an instance contains a facet of type *ClientEvent* or *ServerEvent*, the software assigns the facet to the **ClientEvents** and **ServerEvents** container, page 292 respectively.

NOTE: By default, remaining facets will be assigned automatically, page 368 when you generate the project or a section.

Assigned Facet Description

Facet Description

The **Assignments** pane, page 330 of the **Assignment Editor** provides information on the facets that are assigned to an FBD section or tag container.

To see a list of facets that are assigned to a project, select the project in the **Containers** pane.

The following information is displayed in the **Assignments** pane.

Column header	Description
Container	Identifier of the section/container to which the facet is assigned.
Instance	Identifier of the instance that references the facet.
Instance Template	Identifier of the template that the instance is using.
State	<p>State of the facet in the context of the instance.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Valid • Invalid, page 180 • Deleted: The section is generated, page 383 and you have unselected the corresponding element of the instance in the Instance Editor or deleted the instance itself.
Facet	Unique identifier of the facet.
Facet Template	Identifier of the template that the facet is using.
Path	Path to the facet in the composition of the instance, page 176.
Order	<p>This column applies to Control projects only.</p> <p>Indicates the order in which the software generates constituents of the facets inside the section and as such, the order in which the Control Participant executes the constituents inside the section.</p> <p>You can change the order of facets, page 381 inside a section.</p> <p>The software applies the order only during:</p> <ul style="list-style-type: none"> • A first generation, page 388 • A regeneration, page 421 <p>For more detailed information on how the software manages the facet order, refer to the description of the generation functionality, page 390.</p>
Assignment	<p>Indicates the status of the facet in relation to the instance that references the facet.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Assigned • Unassigned • Out Of Date • Deleted <p>Refer to <i>Assignment Status of Generated Facets</i>, page 411 for a description of the assignment status.</p>
Generation	<p>Indicates the status of the facet in relation to the generation of its constituents in the logical Participant project.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Non Generated • Generated • Moved • Inconsistent • Unlinked <p>Refer to <i>Generation Status of Facets</i>, page 411 for a description of the generation status.</p>

Changing the Element Selection of Instances After Assignment

If you change the Control or Supervision element selection of an instance in the **Instance Editor** window after assigning facets of the instance (before generating), the changes are reflected in the respective **Assignment Editor**.

The table describes how the changes to the element selection of an instance are reflected in the **Assignment Editor**.

When you...	Then...
Select a new element of an instance	The corresponding facets are added to the existing facets of the instance and you can display them in the Facets pane of the Assignment Editor . Proceed with the assignment as described in the assignment procedure.
Unselect an element of an instance	The corresponding facets are removed from the list of facets in the Facets pane of the Assignment Editor . If you had already assigned the facets, then they are also removed from the corresponding container.

NOTE: Changes impacting the assignment status of facets that you have generated are described in *Assignment Status of Generated Facets*, page 411.

Automatic Assignment of Remaining Facets

Overview

The **Project Explorer** features a functionality that allows you to assign remaining Control and Supervision candidate facets of instances, which have at least one facet that is already assigned to a Participant project. The assignment is done automatically by the software when you select a generation command, page 386. The container to which these facets are assigned depends on where the other facets of the instance are already assigned.

There can be candidate facets remaining because you have:

- Not assigned all the facets to a Participant project yet.
- Unassigned facets.
- Changed the element selection of the instance, page 182 after assignment, which has created new facets.

Information on facets that the software has assigned automatically is displayed in the notification panel.

NOTE: Automatic assignment does not apply to Supervision genie facets.

Using and Disabling the Functionality

The functionality is enabled by default and remains enabled until you disable it in the settings of a Control project, page 263 or Supervision project, page 290.

Assignment is triggered, page 386 when you use any generation command at the project or project container level.

NOTE: You can still assign facets manually while the functionality is enabled.

Automatic Assignment Criteria

The table describes how the software assigns remaining candidate facets of an instance, depending on where other facets of the instance are already assigned.

Candidate facet type	Facets of same instance are already assigned to one container	Facets of same instance are already assigned to several containers
Control facet	Candidate facets are assigned to the same FBD section.	Candidate facets are assigned to the FBD section to which other facets of the instance are already assigned and that has the lowest Order number.
Supervision facet	Candidate facets are assigned to the same tag container.	<ul style="list-style-type: none"> • Supervision project with one cluster: Candidate facets are assigned to the tag container that was created first. • Supervision project with several clusters: Candidate facets are assigned to each cluster. In each cluster, candidate facets are assigned to the tag container that was created first.

NOTE: Only facets that are available are assigned. If a facet is locked, for example, because the instance referencing it is open in the **Instance Editor**, it remains unassigned.

Viewing Assignments

To verify to which container facets of an instance are assigned, use the **View Assignments**, page 379 window.

Examples

In this example, three facets of instance *Motor_1* are already assigned to section *FBDSection_1*. A modification of the element selection of *Motor_1* has created facet *Motor_1_DEVMT*, which is candidate to be assigned. The software assigns *Motor_1_DEVMT* automatically to section *FBDSection_1* when you generate the project. (Information has been highlighted in yellow for this example.)

The screenshot shows the ControlProject_1 Assignment Editor with three main windows:

- ControlProject_1 - Assignment Editor**: Shows the project structure with Auto_assign, Folder_1, and Motor_1.
- ControlProject_1 - Instances**: Shows four Motor_1 instances under \$Motor.
- ControlProject_1 - Facets**: Shows facets assigned to Motor_1 instances. One facet, Motor_1_DEVMT, is highlighted in yellow and marked as a candidate.
- ControlProject_1 - Containers**: Shows FBDSection_1 selected.
- ControlProject_1 - Assignments**: Shows assignments for FBDSection_1. It lists Motor_1 instances and their assigned facets. Motor_1_DEVMT is listed as unassigned.
- ControlProject_1 - Facets**: Shows facets for Motor_1 instances again, with Motor_1_DEVMT now assigned to Motor_1 instance 1.
- ControlProject_1 - Assignments**: Shows assignments for Motor_1 instances. Motor_1_DEVMT is now assigned to Motor_1 instance 1.

In this example, three facets of instance *Motor_1* have been assigned to section *FBDSection_1*. The section was generated and the three facets unassigned. Thereafter, modification of the element selection of *Motor_1* has created facet *Motor_1_DEVMT*, which is candidate to be assigned. The software assigns *Motor_1_DEVMT* automatically to section *FBDSection_1* when you generate the project again while the three unassigned facets are removed from it. (Information has been highlighted in yellow for this example.)

The screenshot shows the ControlProject_1 Assignment Editor with the same structure as the previous example, but after generating the project:

- ControlProject_1 - Assignment Editor**: Shows the project structure.
- ControlProject_1 - Instances**: Shows four Motor_1 instances under \$Motor.
- ControlProject_1 - Facets**: Shows facets assigned to Motor_1 instances. Motor_1_DEVMT is now marked as a candidate.
- ControlProject_1 - Containers**: Shows FBDSection_1 selected.
- ControlProject_1 - Assignments**: Shows assignments for FBDSection_1. Motor_1_DEVMT is listed as unassigned.
- ControlProject_1 - Facets**: Shows facets for Motor_1 instances again, with Motor_1_DEVMT now assigned to Motor_1 instance 1.
- ControlProject_1 - Assignments**: Shows assignments for Motor_1 instances. Motor_1_DEVMT is now assigned to Motor_1 instance 1.

Automatic Assignment of Supervision Facets

Overview

This topic describes the functionality that is enabled when you associate a Control project to a tag container by using the *Associated Control Project* property of the tag container, page 340.

Working Principle

When you associate a tag container to a Control project, whenever you drag instances (or a folder) to the **Containers** pane in the **Assignment Editor** to assign their Control facets to this Control project, the Supervision data facets of these instances are automatically assigned to the associated tag container.

You can see the details of the Supervision facet assignment in the notification panel.

The automatic assignment of Supervision facets is not performed in the following cases:

- When you drag Control facets from the **Facets** pane.
- When remaining candidate Control facets are automatically assigned by the software, page 370.
- When Control facet assignments are created by means of import.
- The instance has no Supervision data facets. In such case, an entry for unsuccessful facet assignment appears in the notification panel.

Visual Indicator

The following partial view of the **Assignment Editor** shows two tag containers where only *TagContainer_1* is associated to a Control project. It is indicated by a blue icon.

Supervision_1 - Containers	
Path	Container
\	Supervision_1
.\Supervision_1\Cluster_1\Tags	TagContainer_1
.\Supervision_1\Cluster_1\Tags	TagContainer_2

Further, when you hover with the pointer over the tag container, a tooltip indicates the identifier of the Control project.

Locked Supervision Projects

If the Supervision project that contains the tag container is locked (for example, because it is open for refinement) when the software attempts to assign the Supervision facets, a dialog box opens. It gives you the following choices:

- **Yes:** The dialog box closes and the software will attempt to assign both the Control and the Supervision facets again after waiting for approximately 1 min.
 - If, at that moment, the Supervision project is still locked, the dialog box opens again.
 - If the lock has been released, both facets types are assigned without further notification. You can see the details of the facet assignment in the notification panel.

During the wait time, certain actions that you or other users perform on the Control and the Supervision projects are queued without further notification. The software will perform these actions automatically once the initial facet assignment is completed. You can see the details of these actions in the notification panel once they are completed. Such actions can be, for example, assigning additional facets to existing tag containers.

Other actions may be forbidden and a dialog box opens to inform you. For example, refining either project or creating new containers.

- **No:** Only the Control facets are assigned and you can resume working normally. You need to assign the corresponding Supervision facets manually. If you click **No** the second time the dialog box opens, actions that are queued will be performed in sequence.
- **Cancel:** No assignment is performed.

NOTE: If the instance whose Control facets you are assigning has no Supervision data facets to be assigned and the Supervision project is locked, the dialog box still opens. In this case, click **No**.

NOTE: If the dialog box opens and the Supervision project has already been released when you click **Yes**, the wait time still needs to elapse before the software performs the assignment of facets.

Unassigning Control Facets

When you unassign Control facets, the corresponding Supervision facets that were assigned automatically remain assigned.

Impact on Supervision Project Export and Import

The value that is selected for the *Associated Control Project* property is exported.

When you import into a system a Supervision project, which has a tag container that is associated to a Control project, the value of the *Associated Control Project* property after import is as follows:

- If the system contains no Control project with the same identifier, the value is reset.
- If the system contains a Control project with the same identifier, the value is kept and automatic assignment of Supervision facets is enabled.

Viewing the Application Structure in Control Projects

Overview

Switching to tree view mode in the **Containers** pane, page 330 of the **Assignment Editor** allows you to view FBD sections based on the location of instances whose Control facets are assigned to these sections.

It shows the relationship between the folder structure of the application and the sections of a Control Participant project. This is useful, for example, when you want to view the logic related to a specific area or subdivision of the application or verify that the Control facets of a group of instances are assigned to the right sections of a project.

Any FBD section to which a Control facet is assigned appears in the tree view. A section can appear several times depending on the facets it contains.

Independently of where and how many times an FBD section appears, its contents, which is visible in the **Assignments** pane, is the same.

Changing the application folder structure, the location of instances, facet assignments, and/or the status of facets has an impact on the position of sections in the tree view.

Empty FBD sections and non-FBD sections are shown as well.

NOTE: You can view the application structure also when refining Control Participant projects, page 438 and making changes online, page 728.

Root Folder Description

In tree view mode, up to three root folders can appear in the **Containers** pane.

Root folder identifier	Description
<system identifier> (for example, <i>System_1</i>)	<p>The <system identifier> (for example, <i>System_1</i>) root folder is the parent folder for the application structure. This structure contains sections to which facets with the following status are assigned:</p> <ul style="list-style-type: none"> • <i>Assigned</i> • <i>Out Of Date</i> • <i>UnAssigned</i> • <i>Non Generated</i> • <i>Generated</i> • <i>Moved</i> • <i>Inconsistent</i>
EmptyContainers	<p>Parent folder for sections to which no facet is assigned because you have performed one or more actions such as:</p> <ul style="list-style-type: none"> • Created the section manually. • Unassigned all non-generated facets from the section. If the facets were already generated, you have unassigned all facets and generated the section. • Disabled a service, which removed the corresponding non-generated facet. If the facet was already generated, you have disabled the service and generated the section. • Deleted any instance referencing a non-generated facet assigned to the section. If the facets were already generated, you have deleted the instances and generated the section. • Moved all facets to another section. • The section is a non-FBD section.
DeletedFacetContainers	<p>Parent folder for sections to which at least one facet is assigned with the status <i>Deleted</i>.</p> <p>If the section contains at least one facet with another status, the section also appears under the <system identifier> root folder.</p> <p>Once you generate the section and the <i>Deleted</i> facet is removed from the section, the section appears only under either of the two other root folders depending on the facets that are assigned to it.</p> <p>No application folders are shown under this root folder.</p>

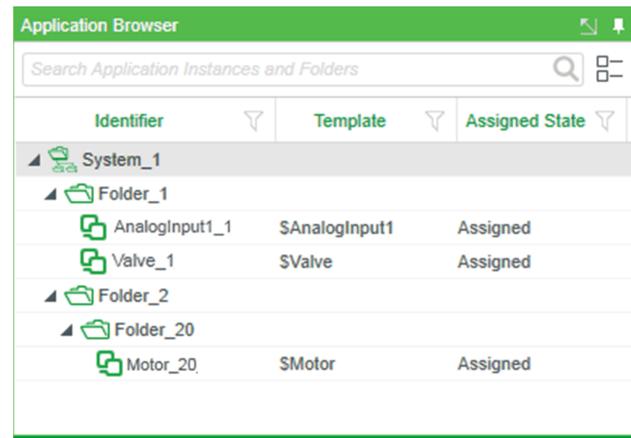
NOTE: A root folder appears only if there is a section in its folder structure. For example, if there is no empty section, the **EmptyContainers** root folder is not shown.

Relation Between Folder Structure and FBD Sections

A folder structure can contain several sections if facets referenced by instances in that folder structure are assigned to multiple sections.

A section can be shown multiple times under different folder structures if the facets assigned to this section are referenced from instances located in various folders.

In this example, the application contains three folders. The Control facets referenced by each instance are assigned to separate FBD sections. *Folder_2* is empty.



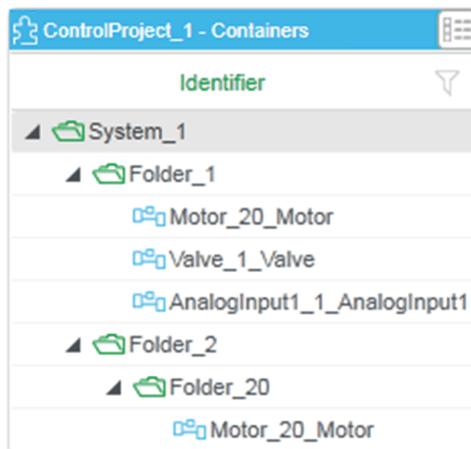
The following is an extract of the **Assignment Editor**, which shows the corresponding FBD sections in the **Containers** pane of the Control project in *grid view mode*. The project node is not shown.

Identifier	Order
Valve_1_Valve	0
AnalogInput1_1_AnalogInput1	1
Motor_20_Motor	2

The following figures show how the FBD sections are displayed in relation to the folder structure of the application in the **Containers** pane in *tree view mode*.

Identifier
System_1
Folder_1
Valve_1_Valve
AnalogInput1_1_AnalogInput1
Folder_2
Folder_20
Motor_20_Motor

If during assignment, you had assigned one facet of instance *Valve_1* also to section *Motor_20_Motor*, this section would appear twice. Under *Folder_20* because of facets of instance *Motor_20* and under *Folder_1* because of the facet of *Valve_1*. The following figure shows this.



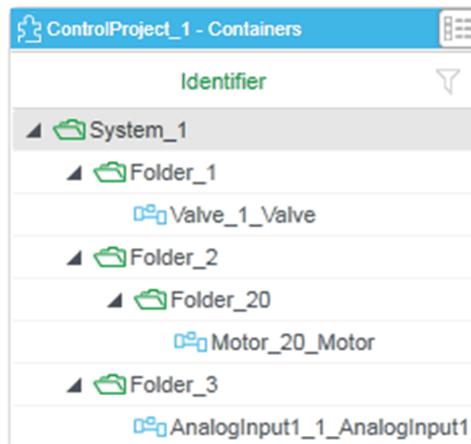
Impact of Changes in the Application on the Tree View

In tree view mode, changes to the application are reflected.

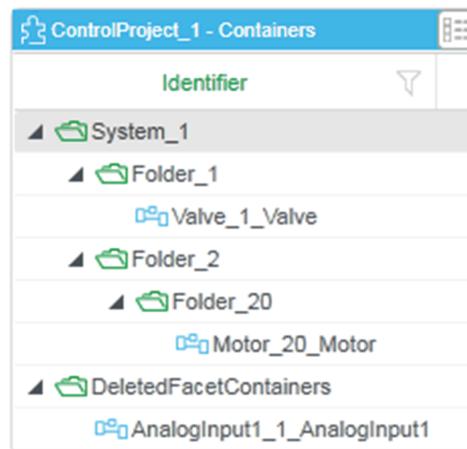
Changes that you make to the folder structure of the application are reflected in folder structure shown in the **Containers** pane.

Changes that you make to the location of instances whose facets are assigned to sections are reflected in the position of these sections in the **Containers** pane.

Based on the application of the previous example, if you create *Folder_3* and move instance *AnalogInput_1* to this folder, the tree view is modified as shown in the figure.



If you delete instance *AnalogInput_1* and its facets were generated, the status of the facets changes to deleted. When this happens, section *AnalogInput_1_AnalogInput* appears under the **DeletedFacetContainers** root folder as shown in the figure.

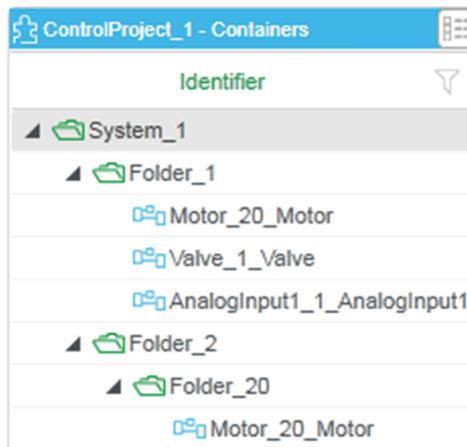


NOTE: *Folder_3* is empty and therefore it does not appear in the **Containers** pane.

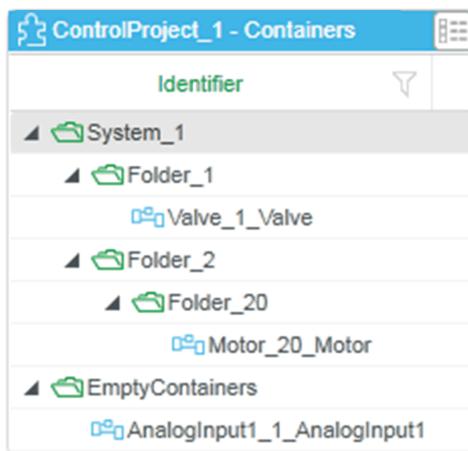
Impact of Changes to Assignments on the Tree View

Changes that you make to the assignment of facets are reflected in the position of sections in the **Containers** pane.

If without changing the application, you move a facet of the *Valve_1* instance to the *Motor_20_Motor* section, the section appears also under *Folder_1* because the instance is located in *Folder_1*.



If you had generated section *AnalogInput_1_AnalogInput*, delete the instance, and generate the section again, the *Deleted* facets are removed from the section and the empty section now appears under the **EmptyContainers** root folder. The result is shown in the following figure.



Managing Assigned Facets

Overview

This section describes the actions that you can perform on assigned facets by using the **Project Explorer** and the **Application Explorer**.

It also describes how to access containers to which facets are assigned and how to open the **Assignment Editor** from facets that you have not yet assigned to a project.

As part of the simplified workflow, page 57, it describes how to edit, link, and copy instances by using context menu commands of their assigned facets.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Facet Actions After Assignment

Overview

This topic applies to facets that are assigned but not yet generated.

However, instance-related commands that pertain to the simplified workflow, page 57 are available also for already generated facets.

For information on facets that are generated, refer to the topic describing management of generated facets, page 430.

Commands and Actions

Once you have assigned Control or Supervision data facets of instances to the respective project, you can execute the following commands and actions on these facets from the **Assignments** pane of the **Assignment Editor**.

Command/Action	Description
Unassign	<p>Removes the facet from the section or container.</p> <p>You cannot unassign facets that have the Assignment status of deleted.</p> <p>Refer to <i>Unassigning Facets</i>, page 378.</p> <p>NOTE: The actions that the software executes when you select the Unassign command for a non-generated facet is different than if you execute the command on a <i>generated facet</i>, page 432.</p>
Edit Instance	<p>Opens the instance that references the facet in the Instance Editor inside the Project Explorer, which lets you edit the instance.</p> <p>When you save, changes to Control services are automatically reflected in the section. That is, if, for example, you enable/disable a service, the corresponding facet is automatically assigned/unassigned.</p> <p>The command is only available for Control facets.</p>
Edit Links	<p>Opens the instance that references the facet in the Links Editor inside the Application Explorer, which lets you link the instance to other instances of the application.</p> <p>When you close the Links Editor, changes pertaining to links that you have created, modified, or removed are automatically reflected in the section.</p> <p>The command is only available for Control facets.</p>
Go To Instance	<p>Shows the instance that references the facet in the Application Explorer.</p> <p>The navigation feature is not available if the Assignment status of the facet is deleted.</p> <p>Refer to the topic describing how to <i>navigate from facets to the instance</i>, page 382.</p>
Copy Instances	<p>Adds the instance that references the facet to the Clipboard. This lets you create copies of the instance and assign their facets automatically by using the Paste Instances FBD section context menu command, page 335.</p> <p>If you select several facets belonging to different instances, each instance is added to the Clipboard.</p> <p>The command is only available for Control facets.</p>
Move the facet (action)	<p>Changes the assignment of a facet to another FBD section or tag container. Does not apply to genie facets, page 495.</p> <p>Refer to <i>Moving Facets</i>, page 378.</p>

NOTE: The **Reassign**, page 432 and **Unlink**, page 432 commands that you can access from the facet context menu become available only once you have completed the generation stage.

Unassigning Facets

To unassign a non-generated facet from a section or container, proceed as follows.

Step	Action	Result
1	In the Assignments pane of the Assignment Editor , right-click the facet and select Unassign . NOTE: You can select multiple facets to unassign them at once.	The facet is removed from the Assignments pane and: <ul style="list-style-type: none"> • The status of the unassigned facet displayed in the System column in the Facets pane changes from Assigned to Not assigned if the facet is not assigned to any other project of the system. • The status of the Project column changes from Done to Candidate.

Moving Facets

To move a non-generated facet from one FBD section/tag container to another, proceed as follows.

Step	Action	Result
1	In the Assignments pane of the Assignment Editor , right-click any one of the column headers to clear filters and sorting, page 118.	-
2	Drag the facet to another FBD section of the same task folder or tag container that is shown in the Containers pane. NOTE: You can select multiple facets to move them at once.	<ul style="list-style-type: none"> • The facet is removed from its original container and assigned to the target one. • The assignment status of the facet is unchanged.

NOTE: Control facets that you move are assigned to the new section with the highest order value in the **Order** column of the **Assignments** pane, page 330. When you move several facets at once, the facet that you have selected first gets the lowest order out of the ones that are moved.

Viewing Facet Assignments

Overview

By opening the **View Assignments** window for any instance of the application, you can verify if Control facets and Supervision (data and genie) facets are assigned to a project of the system.

For each facet, you can:

- If the facet is not yet assigned: Open the corresponding **Assignment Editor**, page 330 for existing projects of the system. For genie facets, you can open the **Pages** container of existing Supervision projects.
- If the facet is already assigned: Open each container to which it is assigned and locate the facet.

Opening the View Assignments Window

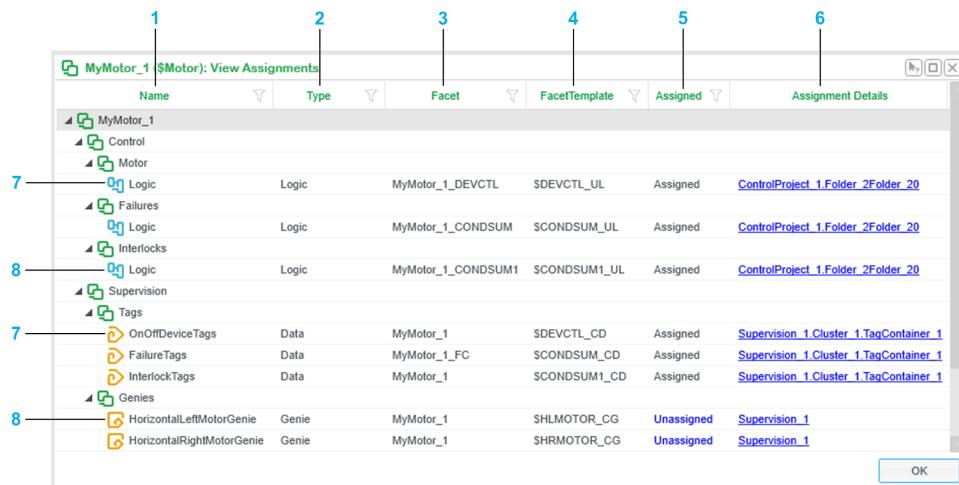
In the **Application Browser** pane, page 155, right-click an instance and select **View Assignments**.

NOTE: You cannot open the **View Assignments** window if both conditions are true for the instance:

- **Assigned** status is **Unassigned**.
- **Data** or **Link** status is **Invalid**.

View Assignments Window

The following figure shows an example of the **View Assignments** window for instance *MyMotor_1*, whose Control and Supervision data facets are assigned but not its genie facets.



Item	Description
1	Displays the elements of the selected instance grouped by Participant. For each Participant, elements are grouped by one or more levels of categories. For example, category Control and subcategory Interlocks contain element Logic .
2	Indicates the subtype of the facet.
3	Identifier of the facet that corresponds to the element.
4	Identifier of the template that the facet is using.
5	Facet assignment status. Possible values: <ul style="list-style-type: none">• Unassigned: The facet is not yet assigned to any project of the system.• Assigned: The facet is assigned to, at least, one project of the system.

Item	Description
6	<p>For Control facets, the following information is displayed:</p> <ul style="list-style-type: none"> • For unassigned facets: A list of existing Control projects in the system to which you can assign the facet. Click a project in the list to open its Assignment Editor. • For assigned facets: The list of sections to which the facet is assigned. Section identifiers are shown in the format: ProjectIdentifier. SectionIdentifier. Click a section to show or open the Assignment Editor. The section and facet are highlighted. <p>NOTE: For unassigned facets, if no Control project exists in the system, the column is empty.</p>
6	<p>For Supervision data and genie facets, the following information is displayed:</p> <ul style="list-style-type: none"> • For unassigned facets: A list of existing Supervision projects in the system to which you can assign the facet. Click a project in the list to open its Assignment Editor. • For assigned facets: The list of containers to which the facet is assigned. Container identifiers are shown in the following format format: <ul style="list-style-type: none"> ◦ For data facets: ProjectIdentifier.ClusterIdentifier. ContainerIdentifier. ◦ For genie facets: ProjectIdentifier.PageIdentifier. Click a container to show or open the Assignment Editor. The container and facet are highlighted. <p>NOTE: For unassigned facets, if no Supervision project exists in the system, the column is empty.</p>
7	Elements of the instance, which generate a facet and that are shown when you open the Instance Editor .
8	Elements of the instance, which generate a facet and that are shown when you inspect the instance, page 204.

NOTE: The **View Assignments** window is closed automatically when you navigate to the **Assignment Editor**.

Order of Control Facets

Overview

The software assigns an order, page 366 to facets that you assign to an FBD section, starting at 0, and displays it in the **Assignments** pane. The values appear in the **Order** column.

The order corresponds to the execution order of the constituents of the facet by the Control Participant inside the section after you generate the section.

You can change the order of facets inside an FBD section.

The software applies the default order or changes to this order only until the first generation, page 390 of the section. After you have generated the section a first time, you can still change the order of facets inside the section. However, the change has no effect on the order of already generated facet constituents if you generate the section again. You need to regenerate, page 421 the section for the software to apply the new facet order.

Changing the Facet Order

To change the order of Control facets inside an FBD section, proceed as follows.

Step	Action
1	In the Containers pane, select the FBD section that contains the facets that you want to rank. Result: The facets that are assigned to the selected FBD section appear in the Assignments pane.
2	In the Assignments pane, right-click any one of the column headers and clear filters and sorting, page 118. Result: The facets are displayed in increasing order from top to bottom.
3	Drag a facet up or down to a new position until you see a tooltip, which indicates the position that the facet will occupy when you release the mouse button. Result: The facet appears in the selected position with a new order value.

NOTE: To change the execution order of the FBD sections, refer to the topic describing how to *change the order of sections*, page 266.

Navigating from Facets to the Instance

Overview

From a facet that is shown in the **Assignments** pane of the **Assignment Editor**, you can open the **Application Explorer** to show the instance that references this facet.

The navigation feature is available for:

- Control facets.
- Supervision data and genie facets.

NOTE: This functionality complements the commands that let you edit and link instances, page 377 from their Control facets as part of the simplified workflow, page 57.

Conditions Impacting Navigation to the Instance

The table describes the scenarios where navigation to the instance is limited or not possible.

Condition	Result
The Assignment status of its facet is Deleted .	The Go To Instance command is not available.
The instance is not shown in the Application Browser because a filter, page 118 is applied.	The Application Explorer opens or is shown but a dialog box informs you that the instance is not visible.
The Application Browser pane is minimized or in the background (if tabbed).	The Application Browser pane remains minimized or in the background. If the instance is open in the Instance Editor , the corresponding tab is shown.

Navigating to the Instance

To navigate from a facet assigned to a project container to the instance that references this facet, proceed as follows.

Step	Action
1	<p>Right-click the facet and select Go To Instance.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> • Opens the Application Explorer window if it is not yet open. If it is, the window is shown. • Expands the application folder containing the instance. • Selects the instance that references the facet.

Generation Stage

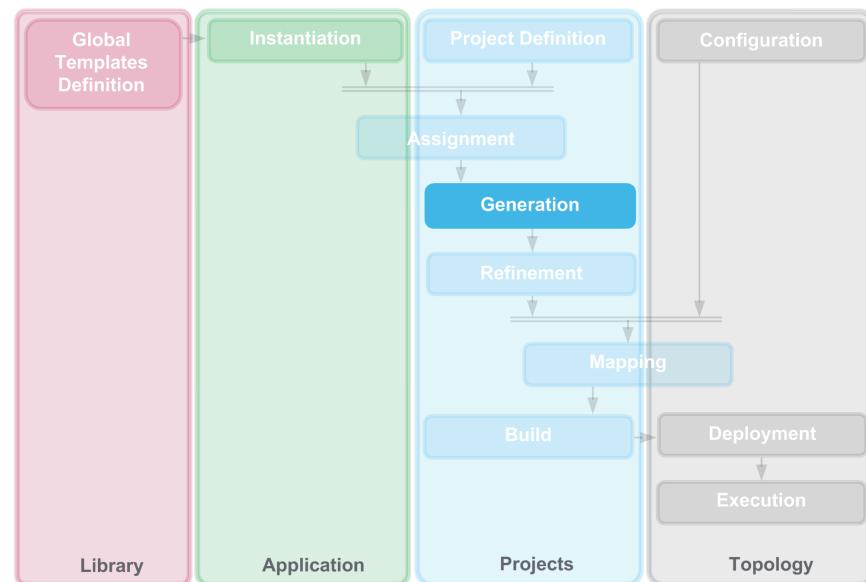
What's in This Chapter

General Description.....	384
Generating a Project for the First Time	413
Generating Projects After Changes.....	416
Generating Equipment in Supervision Projects	424
Generation Diagnostic Messages	427
Managing Generated Facets	430

Overview

This chapter describes how to create the logical Control and Supervision Participant projects of the system.

The following figure shows the position of the **Generation** stage within the system engineering life cycle.



Refer to the Generation stage, page 46 for a description of the purpose of this stage.

General Description

Overview

This section gives a general description of the generation process of Control and Supervision Participant projects.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Generation Process Description

Overview

Once you have assigned Control and/or Supervision facets to the containers of their respective projects, you need to generate these projects.

During the system engineering life cycle, the generation process of Participant projects can encompass up to three steps:

- A first-time generation, page 413.
- A generation after changes, page 416.
- A Control Participant project regeneration, page 421.

You can generate the entire projects or parts of it.

Control projects can be generated automatically, page 394.

The generation process creates the logical Participant project.

For Control projects the generation process:

- Creates the derived function blocks (DFBs) and variables based on the constituents that are encapsulated inside the facets according to the assignment of the facets in the structure of the project and their properties, page 176.
- Creates links between function blocks generated from different facets, which are assigned to the same FBD section, page 389. In addition to links that are defined at the template level, links created in an editor of the **Application Explorer** are processed.

For Supervision projects, the generation process populates the native Supervision Participant database files for:

- Tags from Supervision data facets.
- Report events from Supervision server event facets.
- Events from Supervision client event facets.
- Supervision project components.
- User interface messages.
- Equipment, page 425.

NOTE: You can abort the generation of projects, page 414 and their containers, page 415 as well as the regeneration, page 422.

Data Validation Before Generating

When you select the **Generate** or **ReGenerate** command for a project or a container, the software first verifies if the constituents of the assigned facets satisfy the applicable naming rules, page 88.

The software behaves as follows when it detects that the applicable naming rules are not satisfied:

- For Control facets, it does not generate containers nor constituents in the logical Participant project even those that satisfy the naming rules.
Refer also to the topic describing the impact of detected errors on generation, page 389.
- For Supervision facets, it generates only the constituents that satisfy the applicable naming rules.
- For both facet types, it displays a diagnostic message, which contains the identifiers of the containers, assigned facets, and their parameter values that do not satisfy a naming rule.

Generating Facets of the Same Instance Assigned to Several Projects

You can generate several projects simultaneously even when facets of the same instance are assigned to these project.

The projects can be Control and/or Supervision projects.

Generating Automatically Assigned Facets

By default, the software assigns candidate facets automatically, page 368 when you select any generation command at the project or project container level.

If among the instances whose facets are already assigned to the Participant project, at least one facet remains with the status **Candidate**, the software opens a dialog box to remind you that the functionality is enabled before executing the command. You have the possibility to cancel generation and disable the automatic assignment of these facets.

If the software is not able to proceed with the automatic assignment of a facet, generation of assigned facets completes but the notification panel displays information on the facets that it could not assign and generate.

If generation does not complete successfully, facets that the software has assigned automatically are unassigned.

NOTE: Generation takes a little longer to complete when the functionality is enabled.

Generating During Build Stage

To save time or if you have made minor changes to a logical Participant project, instead of using the **Generate** command, you can use the **Generate and Build** command from the Control executable, page 578 or Supervision executable, page 619 context menu. This performs a generation followed by a build. You need to have completed the system engineering life cycle up to the build stage.

If you generate a complete Participant project and generation of any section, page 389 or generation of any container does not complete, build is not started.

Information about the sections/containers that were not generated is displayed in the notification panel.

If build is not started or does not complete successfully, the result of the generation process is the same as if you had selected the **Generate** command for the Participant project. This means, for example, that for a Control Participant project, sections that were successfully generated remain generated.

The **Generate and Build** command:

- Cannot be used to perform a regeneration, page 421.
- Can be used together with the automatic facet assignment functionality, page 368.
- Automatically performs a **Build** or **Build All** command, page 628 depending on the type of change that you have made to the Participant project.

NOTE: After selecting the command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. Only the user who has selected the command is allowed to abort it.

If you cancel the task:

- Before generation completes, the entire generation task is canceled and no sections are generated. Build does not start.
- After generation completes, only the build task is canceled and the software displays a notification to inform you.

If you are generating and building a small project, the whole task may complete before you are able to click the icon.

Generating Facets with State Invalid

If the **State** of a facet, page 366 has become **Invalid** after you have assigned it, generating the container or Participant project removes the facet from the container.

This can be the case, for example, when you link instances A and B (where instance A needs to be linked to be valid), assign their facets to a Control project, and then delete the link between the two instances. The facet of instance A becomes invalid because the instance is not linked anymore.

Generating the section to which the invalid facet is assigned removes it from the section.

Skipping Facets During Generation

If a facet is locked, the constituents that it encapsulates cannot be generated. In such case, the locked facets are skipped but generation of the section or the Participant project completes for the other facets. This may happen, for example, when an instance is open in the **Instance Editor**.

The generation status of a facet that is skipped does not change.

The notification panel contains information about locked facets that the software could not generate.

Control Project Specifics

Overview

This topic describes the default behavior when generating Control Participant projects and the settings that you can configure to customize the generation process.

Automatic Generation Setting

The **Generation** setting, page 263 of each Control project lets you enable the automatic generation of sections of the project when the facets that are assigned to them undergo a change.

Identifying Code Generated by Control Facets

Constituents that are generated from facets assigned to Control projects are locked in the Control Participant project. The locked code is color coded in the DFB editor of the Control Participant when you refine the project, page 442.

Validity Indicators

When you execute any generation command on a logical Control Participant project or any of its containers (including the **Generate and Build** command) and project analysis is enabled, page 263, the software performs an analysis.

If the result of the analysis raises an alert or detects an error, the software displays a validity icon, page 324.

In such case, refine, page 436 the Control Participant project, identify, and fix the issue. When you save your refinement changes, the software performs a new analysis. The result is shown in the **Analyze Project** tab of the Control Participant window.

NOTE: It is possible that the software detects inconsistencies that are linked to the default **configuration** in the Control Participant, which is part of every logical Control project but that is not relevant in the context of the **Project Explorer** (except for M580 safety Control projects, page 761). These detected errors do not affect the capability to build the project successfully if you configure the topological entities correctly in the **Topological Explorer**.

This can be the case, for example, when the application contains an instance of a communication port template, which requires that you enter a %MW holding register starting address. During generation, this address is checked against the default address in the CPU of the default **configuration**. If both addresses are not consistent, a validity icon is displayed. But when you build the logical Control Participant project, this address is checked against the address that you configure in the CPU of the entity in the **Topological Explorer**, which needs to be consistent.

It can also happen because the logical Control Participant project is missing IODDT variables (for example, when the configuration contains HART modules, page 536) or Device DDT variables, page 725, which exist only in the configuration project until you perform the build or you create them, page 445.

If fixing the inconsistency requires that you change a value in the CPU of the default **configuration**, you still have to verify that the value in the CPU is configured correctly in the **Topological Explorer** by using the **Configure** command for the entity to which the Control Participant project will be mapped.

Generating Graphical Links

Inside the same FBD section, the software generates graphical links between function blocks that use the same variable and that are encapsulated inside different facets only if the following conditions are satisfied:

- The variable is created by a facet (not created manually by the user).
- The resulting link:
 - Links the output pin and input pin of separate function blocks.
 - Is not a circular link.

The software breaks any graphical circular link and creates it as variable link.

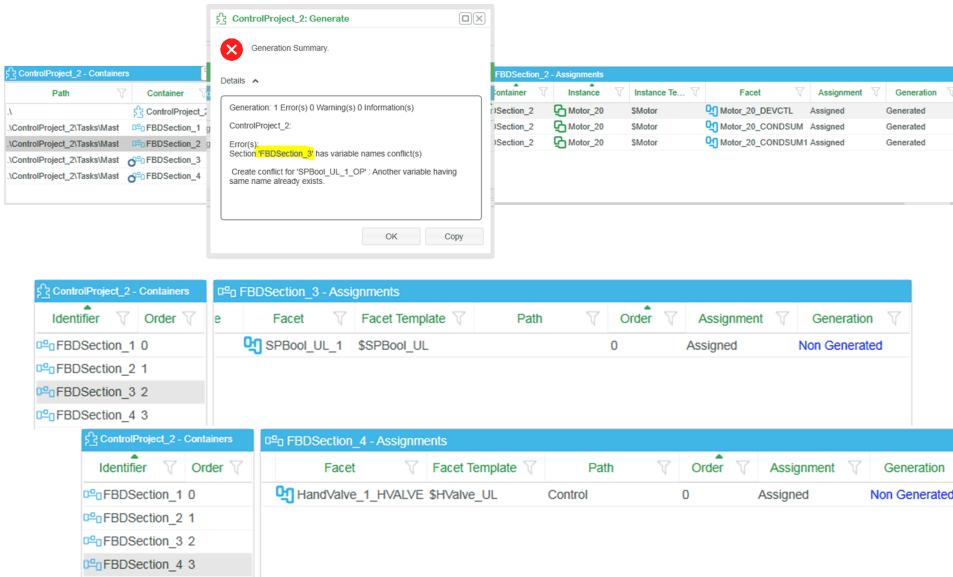
Impact of Detected Errors or User Action During Generation

When you select the **Generate** or **Generate and Build** command for a project containing several sections and the software detects an error, the impact on the generation process depends on the detected error or the user action that is performed.

NOTE: The behavior is the same when you regenerate, page 421 a Control Participant project.

Detected error or user action	Impact on the generation process
Variable name conflict ⁽¹⁾	Generation stops. Code of sections that have been successfully generated remains in the logical Control Participant project.
Type version conflict ⁽¹⁾	The section which contains the detected error and any subsequent sections are not generated.
Naming rule conflict, page 88	A summary dialog box indicates the identifier of the section where generation stopped as well as the reason.
Clicking Cancel in the Type Version Conflict dialog box, page 396.	
Operation in the Control Participant did not succeed (for example, injecting data in a section through import).	
.NET exception occurred (for example, an <code>Object reference not set</code> diagnostic message is shown).	
Section overcrowded ⁽¹⁾	Generation completes but any overcrowded section is not generated. Details are provided in the summary dialog box.
(1) Refer to the topic describing diagnostic messages, page 428.	

In the following example, *ControlProject_2*, which consists of four sections is being generated for the first time but the software stops the generation process because of a variable name conflict in *FBDSection_3* (the section name is highlighted for the purpose of the example). As a result, *FBDSection_1* and *FBDSection_2* are generated in the logical Control Participant project. *FBDSection_3* is not generated because it contains the conflicting variable and *FBDSection_4* is not either because it is generated only once generation of *FBDSection_3* completes



NOTE: If you had selected the **Generate and Build** command, build would not start because at least one section did not generate.

Control Project Section Size

During generation, FBD sections are created in the Control Participant project with a maximum initial size of 360 rows x 240 columns. You can modify this size, page 392.

If additional space is required to fit the function blocks that are generated from the facets assigned to a section, the software adapts the section size by:

- Deleting empty columns on the outer right of the section to add rows at the bottom of the section.
- Deleting empty rows at the bottom of the section to add columns on the outer right of the section.

The software displays a message to inform you of the change in section size.

If you want to reserve some space in the section, you can insert a comment box on the right-hand side of the section and/or at the bottom of the section so that the software does not delete the rows that are left and/or above the comment box when resizing the section.

When the space requirements of the function blocks exceed the maximum section size, the software displays a diagnostic message, page 428.

NOTE: The section resizing process applies to a first generation, subsequent generations, and regenerations.

Execution Order and Function Block Layout

When you generate a Control project for the first time, the software processes the facets that you have assigned to the project and that have an **Assignment** status, page 366 of **Assigned**. It takes into consideration the order of sections, page 266 at the project level and the order of facets, page 366 at the section level.

Refer also to the topic giving an example of section layout management (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

The table describes the process that the software applies during a first generation to manage the order of sections and facets, and the position of function blocks inside a section.

Stage	Description
1	The software detects the order, page 330 of the project sections and processes the sections starting with the one ranked 0; then the section ranked 1; and so on, until it has processed the sections of the entire project.
2	For each section, the software detects the order, page 366 of the first facet of each instance that is assigned to the section.
3	<p>The software generates the function blocks encapsulated in the facets according to the order that it has detected in stage 2, starting with the facet ranked 0. However, the software groups the function blocks according to the layout, page 839 of the composite referencing the facets.</p> <p>For example, if a section contains two instances, Instance1 and Instance2, referencing three and one facets respectively, and ordered as follows:</p> <ul style="list-style-type: none"> Facet1_Instance1 - Order 0 Facet1_Instance2 - Order 1 Facet2_Instance1 - Order 2 Facet3_Instance1 - Order 3 <p>The software generates Facet1_Instance1 before Facet1_Instance2 but groups function blocks of facets 1, 2 and 3 of Instance1 at the top of the section to respect the layout of the composite template used by Instance1.</p> <p>Therefore, function blocks of Facet1_Instance2 are placed at the end in spite of the order of the facet, which is 1.</p>
4	<p>If a function block is generated in a position that is already occupied by another function block, creating an overlap, the function block generated last is moved to the end of the section content.</p> <p>The block keeps its horizontal position (X axis) as defined by the layout of them template referencing it.</p> <p>In such case, the software displays a message advising you about the change in position and a possible change in the execution order.</p>

Facet Status After Unsuccessful Generation

The table describes the effect on the status of Control facets when the generation of a section or any facet is not successful.

Generated component	Facet status
Entire Control project	The generation status of facets assigned to the entire project reverts to the value it was before generating.
Control project section	The generation status of facets assigned to the section reverts to the value it was before generating.

Removing Types from the Logical Control Participant Project

Types generated from Control facets remain in the logical Control Participant project even when you unassign generated facets, page 431 encapsulating them, delete, page 411 the facets by deleting the instance, or delete the section, page 335 to which these facets were assigned.

To remove these types, you can either:

- Regenerate, page 421 the Control Participant project. This discards refinements you have made.
- Purge the types from the Control Participant during refinement. For details, refer to *Purge of Unused Instances* in the Control Participant help.

Configurable Generation Settings

By editing the GenerationSettings.xml file, you can customize the behavior of the generation process of Control Participant projects.

The file is located on the computer on which the system server is installed at the path C:\Program Files\Schneider Electric\EcoStruxure\Process Expert\System Server.

You need to have administrator rights to be able to edit this file.

NOTE: Changes to the file are applied only after a system server restart, page 68.

You can configure the following parameters.

Parameter name	Default value	Description
<i>ReGenerateNonFbdSections</i>	<i>True</i>	<p><i>True</i> = The behavior during a regeneration is the following:</p> <ul style="list-style-type: none"> Non-FBD sections that were created during refinement and their contents are retained. Refinements that are made in FBD sections to which facets are assigned are discarded. Refinements made in FBD sections that were created in the Projects Explorer are retained if no facets have been assigned to these sections. <p><i>False</i> = The behavior during a regeneration is the following:</p> <ul style="list-style-type: none"> Non-FBD sections that were created during refinement and their contents are discarded. Refinements that are made in FBD sections are discarded whether facets have been assigned to them or not. <p>NOTE: The value of this parameter has no impact on the regeneration of data and logic of the safety program of M580 safety Control projects, page 767.</p>
<i>UnityProIgnoreFbdComments</i>	<i>True</i>	<i>True</i> = Comments of FBD sections are not generated.
<i>UnityProMinSpaceTop</i>	0	Minimum space that the software leaves above function blocks that are positioned at the top of the section.
<i>UnityProMinSpaceLeft</i>	10	Minimum space that the software leaves at the left of any function block that it generates.
<i>UnityProMinDistanceX</i>	5	Minimum space that the software leaves between two consecutive function blocks along the x axis.
<i>UnityProMinDistanceY</i>	2	Minimum space that the software leaves between two consecutive function blocks along the y axis.
<i>UnityProUseEffectiveSize</i>	<i>False</i>	<i>True</i> = The minimum space around function blocks is decreased to help utilize the space of the section more effectively.
<i>UnityProInsertToEnd</i>	<i>True</i>	<ul style="list-style-type: none"> <i>True</i> = The software puts function blocks that are generated after a first generation at the end of sections independently of the position that they would normally occupy. <i>False</i> = The software lays out function blocks according to the default behavior when generating Control Participant projects after changes, page 420.
<i>CheckConsistency</i>	<i>True</i>	<p>This is not a user-configurable parameter.</p> <p>Changing the value affects the behavior of the software.</p>
<i>SaveAfterNumberSectionGenerated</i>	30	<p>Number of FBD sections that need to generate successfully to trigger a save command in the Control Participant.</p> <p>If you are generating a number of sections lower than the value, a save command is executed once generation completed successfully.</p> <p>NOTE: Reducing this number may increase generation time because the number of save operations is higher. The setting does not alter the impact of detected errors during generation, page 389.</p>
<i>NumberOfUnitySectionRows</i>	36 ⁽¹⁾	<p>Number of rows/columns that the section contains when it is created by generating the Control Participant project or section for the first time or regenerating it.</p> <p><i>NumberOfUnitySectionRows</i> allowed range: 24 to 144</p> <p><i>NumberOfUnitySectionColumns</i> allowed range: 6 to 36</p>
<i>NumberOfUnitySectionColumns</i>	24 ⁽¹⁾	

Parameter name	Default value	Description
		<p>NOTE: If you modify the default section size or if the Control Participant creates a section that is different from your settings, generation or regeneration may not complete successfully because constituents cannot be positioned, page 390 as required.</p> <p>(1) $\text{NumberOfUnitySectionRows} \times \text{NumberOfUnitySectionColumns} \leq 864$. If this value is exceeded, either 24 or 144 is retained for the number of rows (the value that comes closest to <i>NumberOfUnitySectionRows</i>) and the number of columns adjusted in consequence, independently of the setting of <i>NumberOfUnitySectionColumns</i>. For example, if you enter 200 and 7 respectively, a section of size 1440 x 60 is created because 200 exceeds the allowed number of rows and is closer to 144 than to 24.</p>

Automatic Generation of Sections

Overview

When the automatic generation setting, page 263 is enabled, certain actions trigger the generation of sections. These are actions that have an impact on facets, which are already assigned to or that will be assigned to the corresponding sections.

The behavior in case of a detected error, page 389 is the same as when you generate manually.

The following table indicates which actions trigger the generation of a section when the automatic generation of sections is enabled. If you perform several of these actions in parallel, generation may be queued until the trigger point of the last action is reached.

Action	Automatic generation trigger point
Actions on instances that have facets already assigned to the section	
Actions that modify the identifier of an instance: <ul style="list-style-type: none"> • Moving the instance to another folder⁽¹⁾. • Changing the hierarchy of the instance⁽¹⁾. • Changing the alias of a folder in the hierarchy of the instance⁽¹⁾. • Renaming the instance in-place. • Disabling hierarchical naming of the instance. 	The completion of the action (or saving changes when required).
Editing an instance.	Saving changes and closing the editor.
Deleting an instance.	The completion of the action.
Editing links between instances (creating, moving, deleting a link, removing an instance from the editor), which changes the status of assigned facets to <i>Out Of Date</i> . Does not apply to RTNS links, page 201.	Closing the editor.
Moving an instance from the application of a system (source) to the application of another system (target) ⁽²⁾ .	The completion of the action.
Updating or replacing the template of an instance.	The completion of the action.
Import operation in the application, which changes the status of assigned facets to <i>Out Of Date</i> , <i>Inconsistent</i> , or <i>Deleted</i> .	
Actions in the Control project	
Assigning facets to the section manually.	The completion of the action.
Moving facets from one section to another.	
Unassigning and relinking a facet, which changes the status of the facet to <i>Inconsistent</i> .	
Import operation leading to assigned facets having the status <i>Non Generated</i> .	
Changing the status of a facet to <i>Inconsistent</i> after refining the project offline or online and updating the project.	Closing the Control Participant refinement window or completion of the project update.
Creating an instance from the Control project section.	The completion of the facet assignment. Facets of invalid instances ⁽³⁾ are generated when the instance status becomes <i>Valid</i> .
Copying and pasting an instance in the Control project section.	The completion of the facet assignment.
(1) With hierarchical naming enabled for the instance.	
(2) The section that belongs to the source system is generated.	
(3) Facets of an invalid instance can be assigned if you have created the instance in-place, page 338 or if the instance becomes invalid after its facets were assigned.	

Selecting Versions of Types in Control Facets During Generation

Overview

Because in a logical Control Participant project, only one version of a given type can exist, either of the following situations require that you select which version is generated:

- You are assigning to a Control Participant project facets that encapsulate the same type but different versions of it.
- You have updated or replaced the template, page 212 that is used by an instance whose facets are already assigned to a Control Participant project and generated. The new template contains the same type (DFBs and/or DDTs) but of a different version. The assignment status of one or more of these facets has been set to **Out Of Date**.

If such a case occurs, when you generate the logical Control Participant project to apply the changes, the software detects the presence of several versions for a same type.

As a result, the software opens the **Global Constituent Type Version Conflicts** dialog box in which you can select either command to select the version that you want to use in the logical Control Participant project:

- **Update**
- **Skip**
- **Cancel**

The software maintains a list of the various versions of a type that it detects during each generation cycle and applies your initial selection when it detects a version again. If the software detects a new version of the same type when it generates the next section, it opens the **Global Constituent Type Version Conflicts** dialog box again.

The detection of the version change is performed at the facet level and it is irrelevant to which instance a facet belongs.

However, if two or more versions of the same type are detected in the same section, generation stops. In this case, the software informs you about the facets, the type, and the versions that create the conflict. You need to resolve the version conflict at the instance or facet level manually.

This topic contains several examples illustrating the main scenarios that you may encounter.

NOTE: Although you select the version at the section level, updating the version of a type impacts all its instances across all sections of the Control Participant project. Also, the last action that you select for a given type is the one that is applied.

NOTE: The **Global Constituent Type Version Conflicts** dialog box also opens if the software detects the same version of a type but with a different signature, page 399.

Relation Between Type Versions in the Application and in Control Projects

The version of the type that you select to use in the logical Control Participant project is not necessarily the same as the version that is encapsulated in the template used by instances of the application. Indeed, in the application, instances can use several versions of a same template encapsulating different versions of a type. The version of the type that is used in the logical Control Participant project can only be one of these versions.

This example shows the different versions of a template that are used by three instances of the application. Each template encapsulating a different version of a DFB. It also shows the single type version that is generated in the Control project.

Template version and encapsulated type version in the application	Type version used in the logical Control project
Template_A [1.0.0], type version 1.2	Type version 1.3
Template_A [1.0.1], type version 1.3	
Template_A [1.0.2], type version 2.0	

This second example shows that different versions of a type can be used in different Control projects, each project using only a single version.

Template version and encapsulated type version in the application	Type version used in logical Control project 1	Type version used in logical Control project 2
Template_A [1.0.0], type version 1.2	Type version 1.3	Type version 2.0
Template_A [1.0.1], type version 1.3		
Template_A [1.0.2], type version 2.0		

Impact of Type Version Selection on Links

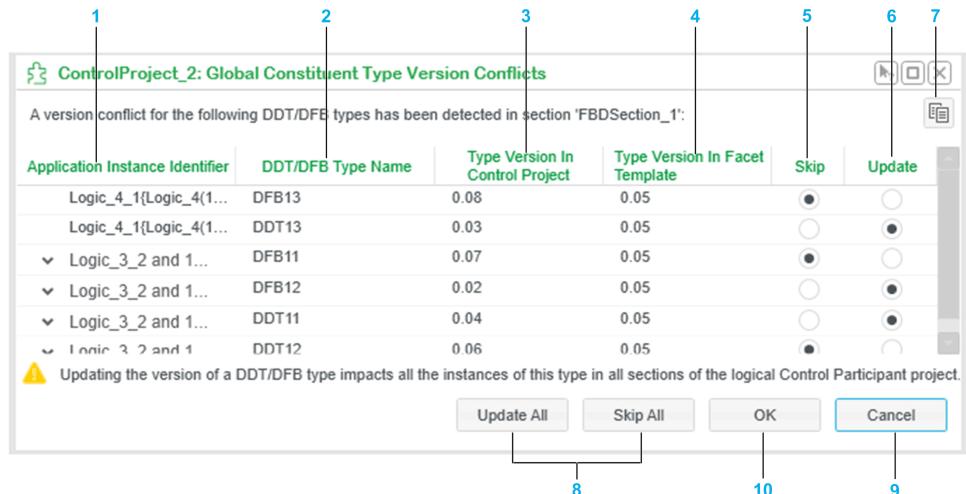
Selecting **Update** or **Skip** for a facet removes links with any instance of the encapsulated type if a pin for which a link exists is impacted in either way:

- The pin is removed.
- The pin is renamed.

The software displays a message informing you of the links that have been removed after generation completes.

Global Constituent Type Version Conflicts Dialog Box

The following figure shows an example scenario where several DFB and DDT types create a version conflict in *FBDSection_1*.



Item	Description
1	Identifier of an instance referencing a facet that is assigned to this section and that encapsulates the type. If other facets encapsulating the type and assigned to the section are referenced by different instances, the number of instances is indicated. Expand the row to view a list of these instances with the template identifier and version each one uses. A tooltip displays the same information when the row is not expanded.
2	Name of the type encapsulated in the facets for which a version conflict is detected.
3	Version of the type that is currently used in the logical Control Participant project.
4	Version of the type that is encapsulated in the facets being generated.
5	Button that lets you select the Skip action for individual types.

Item	Description
	<p>Does not update the type⁽¹⁾ that is currently used in the logical Control project (version X) to version Y, which is the version encapsulated in the facet. Also, it does not generate the facet.</p> <p>For each generation cycle, the software maintains a list of the various versions of a same type that it has detected. Based on this list, if other facets encapsulate version X or y of the same type, no dialog box opens and version X is maintained automatically.</p> <p>If, in another section, the software detects a facet that encapsulates the same type but of a version that is different from version X and version Y (for example, version Z), the Global Constituent Type Version Conflicts dialog box opens again. Version Z is added to the list of versions already detected.</p> <p>The generation status of facets may be set to inconsistent. In such case, when generation completes, the software provides a summary with the name of the facets that are affected and the name of the section they are assigned to.</p> <p>NOTE: Skip only skips the generation of the facet that encapsulates the type. Other facets belonging to the same instance are generated if their status, page 417 requires it (for example, if it is Out Of Date).</p>
6	<p>Button that lets you select the Update action for individual types.</p> <p>Generates the facet and updates the type⁽¹⁾ currently used in the logical Control project (version X) to the version that is encapsulated in the facet (version Y). The status of the facet becomes Generated.</p> <p>For each generation cycle, the software maintains a list of the various versions of a same type that it has detected. Based on this list, if other facets encapsulate version X or y of the same type, no dialog box opens and version Y is used automatically.</p> <p>If, in another section, the software again detects a facet that encapsulates version X of the same type and the facet is not generated, it remains non-generated.</p> <p>If, in another section, the software detects a facet that encapsulates the same type but of a version that is different from version X and version Y (for example, version Z), the Global Constituent Type Version Conflicts dialog box opens again. This allows you to update the current version of the type (for example, version Y) to version Z. Version Z is added to the list of versions already detected.</p> <p>The generation status of facets may be set to inconsistent. In such case, when generation completes, the software provides a summary with the name of the facets that are affected and the name of the section they are assigned to.</p> <p>NOTE: version Y can be a version later or earlier than version X. The same applies to version Z compared to the other versions.</p>
7	Copies the content of the dialog box to the Clipboard.
8	<p>Buttons that let you select an action for the conflicting type versions as a whole.</p> <p>The actions are the same as when you select the action for types individually.</p> <p>NOTE: Clicking these buttons overrides the selection you have made for individual types.</p>
9	<p>Generation of the Control Participant project stops. Code of sections that have been generated successfully remains in the logical Control Participant project. The type⁽¹⁾ version that was used after the generation of the previous section is maintained.</p> <p>The section in which the version conflicts have been detected and any subsequent sections are not generated.</p>
10	Generates the section by applying your selection of actions for conflicting types.

(1) Includes all the types encapsulated in the facet if more than one, for example the DFB and its DDT. You cannot select to update or skip one and not the other.

NOTE: The status of facets is updated only after the entire generation process is completed.

Setting the Status of Facets to Inconsistent

Updating or skipping the version of a type may set the generation status of a facet to **Inconsistent**. The assignment status does not change.

The table describes under which conditions the generation status is changed.

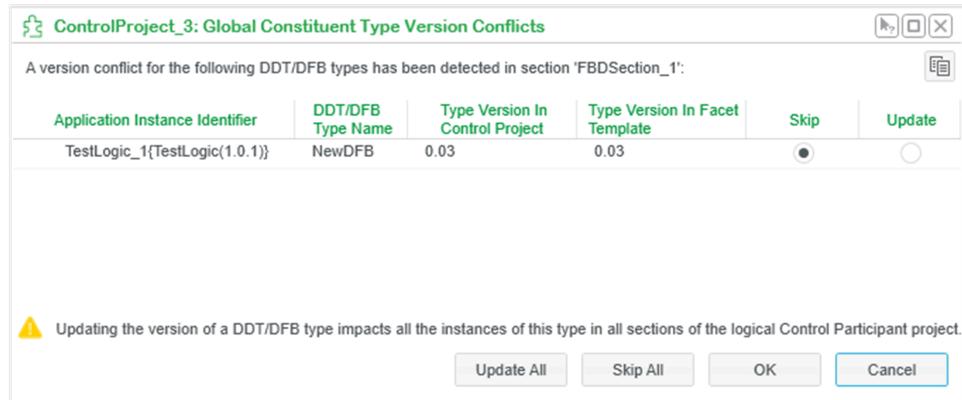
Status of the type version	Required facet status	Required user action
The version has already been detected during this generation cycle but is different from the one currently used in the Control project.	Assignment: Out Of Date	Because the version has already been detected, the Global Constituent Type Version Conflicts dialog box does not open; therefore, no action is required. Refer to example 3, page 404.
The version has not yet been detected during this generation cycle.	Assignment: Out Of Date	You are selecting Skip for the proposed version change after having already selected Update for the same type in another section during the same generation cycle. Refer to example_7, page 408.

NOTE: If the status of the facet is **Non Generated**, it does not change.

Detecting Types of Same Version But With Different Signature

The **Global Constituent Type Version Conflicts** dialog box also opens if the software detects in a facet assigned to the Control project and which needs to be generated, the same version of a type as it is already used in the logical Control project but with a different signature.

The following figure shows an example of detection of a type of the same version but with a different signature. A tooltip indicates the nature of the conflict when you position the pointer over the version of the type.



- **Update:** The type in the logical Control project is updated with the type and signature that is encapsulated in the facet.
- **Skip:** The type that is used in the logical Control project continues to be used; no update is performed. The generation status of facets may be set to inconsistent, page 398.

Independently of the action that you select, the **Global Constituent Type Version Conflicts** dialog box opens again if in another section of the same Control project another facet is detected, which encapsulates the same version of the same type and with a signature that is the same as in the previous facet or with yet another signature.

If two or more facets encapsulating the same version of a type but each with a different signature are detected in the same section, generation stops, page 389.

Performing a Consistency Check After Version Conflict Detection

It is recommended to perform a consistency check, page 449 when the software detects several versions of the same type or the same version with different signature during generation of a Control Participant project.

A consistency check detects which generated facets encapsulate a type whose version or signature is different from the version/signature of the type used in the Control Participant project. The status of such facets is set to **Inconsistent**.

The following example illustrates a scenario where a consistency check allows identifying a facet that contains a version of a type that is different from the version used in the Control project after the software detected a version conflict. In this example, v1.0 of DFB_A type was generated in the Control project by Facet_2. A second facet (Facet_1) containing v2.0 of the same type is assigned to a different section and the project is generated again.

Section	Facet	DBF_A type version	Facet assignment/generation status before generating	Facet status after generating and performing a consistency check	Comment
Section_1	Facet_1	v2.0	Out Of Date/Generated	Assigned/Generated	The software detects a version conflict and you update the type used in the Control project to v2.0 by selecting Update .
Section_2	Facet_2	v1.0	Assigned/Generated	Assigned/Inconsistent	Because the facet encapsulates v1.0 of the type, the consistency check has set the status of the facet to Inconsistent .

Regenerating a Control Participant Project After Type Version Changes

When you select the **Regenerate** command after generating a Control Participant project or section in which version conflicts were detected, the behavior is different depending on what you regenerate.

If you regenerate the entire Control Participant project, the software removes the types from the logical Control project and generates the ones that are encapsulated in facets assigned to the Control Participant project sections. If various versions of a same type are detected again, the **Global Constituent Type Version Conflicts** dialog box opens.

If you regenerate a section of a Control Participant project, the software maintains existing types, page 421 in the logical Control project and starts over the generation process. During regeneration, it compares:

- For the first facet of the section that encapsulates a type, the version of the type.
- The version of this type that exists in the logical Control project.

If they are different, it opens the **Global Constituent Type Version Conflicts** dialog box.

If they are identical, it performs the comparison with the second facet of the section encapsulating a type, and so on.

In case several versions of the same type are detected in the same section, page 400, generation stops.

Examples

The following examples illustrate various type version conflict scenarios when facets are assigned to the same section.

Description	Command used	Refer to
A facet is assigned to an already generated section.	–	Example1a, page 401

Description	Command used	Refer to
A section contains two facets that are not generated.		Example 1b, page 402
A section contains two facets that are generated and one of them is updated to use a different version of the type.		Example 1c, page 403
A generated facet is updated while the same section contains a non-generated facet. This example illustrates a possible solution to examples 1.	Update and Skip	Example 2, page 403

The following examples illustrate various type version conflict scenarios when facets are assigned to different sections.

Description	Command used	Refer to
Two different versions of the same type in different sections of a Control project.	Update	Example 3, page 404
Several different versions of the same type in different sections of a Control project.		Example 4, page 405
Two different versions of the same type in different sections of a Control project.	Skip	Example 5, page 406
Several different versions of the same type in different sections of a Control project.		Example 6, page 407
Several different versions of the same type in different sections of a Control project.	Update and Skip	Example 7, page 408
Two different versions of the same type in different sections of a Control project.	Cancel and Update The Regenerate command is also described.	Example 8, page 409

Example 1a: Two Different Versions of the Same Type in the Same Section

A facet is assigned to an already generated section.

The starting point is the following:

- One facet is assigned to Section_1 and generated.
- The facet encapsulates DFB_A.

Facet ID	DFB_A version encapsulated in facet	Facet assignment/generation status	DFB_A version used in Control project
Facet_0	v1.0	Assigned/Generated	v1.0

Then, another facet encapsulating DFB_A but of a different version is assigned to the same section. The section is generated again to apply the changes.

The table explains the behavior during generation of Section_1 and which version of DFB_A is used in the end.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status after generation	DFB_A version used in Control project after generation	Comment
Facet_0	v1.0	Assigned/Generated	v1.0	The software detects v2.0 while v1.0 is already generated. Because the conflict arises in the same section, generation stops, page 389 and DFB_A version v1.0 is maintained.
Facet_1	v2.0	Assigned/Non Generated		<p>The status of facets is unchanged.</p> <p>To be able to generate the section, you can either:</p> <ul style="list-style-type: none"> • Unassign one of the facets from the section. • Update the template used by one of the instances so that both facets use the same version of DFB_A, page 403. • Unlink Facet_0, generate the section, select Update when the Global Constituent Type Version Conflicts dialog box opens, then relink Facet_0.

(1) The order of facets in the section is not relevant.

(Back to example summary table, page 400.)

Example 1b: Two Different Versions of the Same Type in the Same Section

A section contains two facets that are not generated.

The starting point is the following:

- Two facets of two different instances are assigned to Section_1.
- Both facets encapsulate DFB_A but of a different version.
- DFB_A does not exist in the Control project.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status
Facet_0	v1.0	Assigned/Non Generated
Facet_1	v2.0	
(1) The order of facets in the section is not relevant.		

Then, the section is generated.

The table explains the behavior during generation of Section_1 and which version of DFB_A is used in the end.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status after generation	DFB_A version used in Control project after generation	Comment
Facet_0	v1.0	Assigned/Non Generated	None	<p>The software detects v1.0 and v2.0 in the same section. Because of this, generation stops, page 389 and no version of DFB_A is generated.</p> <p>The status of facets is unchanged.</p> <p>To be able to generate the section, you can either:</p> <ul style="list-style-type: none"> • Unassign one of the facets from the section. • Update the template used by one of the instances so that both facets use the same version of DFB_A, page 403.
Facet_1	v2.0			

(1) The order of facets in the section is not relevant.

(Back to example summary table, page 400.)

Example 1c: Two Different Versions of the Same Type in the Same Section

A section contains two facets that are generated and one of them is updated to use a different version of the type.

The starting point is the following:

- Two facets of two different instances are assigned to Section_1.
- Both facets encapsulate the same version of DFB_A.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status	DFB_A version used in Control project
Facet_0	v1.0	Assigned/Generated	v1.0
Facet_1	v1.0		
(1) The order of facets in the section is not relevant.			

Then, the template used by instance referencing Facet_0 is updated. This results in Facet_0 encapsulating a different version of DFB_A. The section is generated again to apply the change.

The table explains the behavior during generation of Section_1 and which version of DFB_A is used in the end.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status after generation	DFB_A version used in Control project after generation	Comment
Facet_0	v2.0	Out Of Date/Generated	v1.0	The software detects v2.0 while v1.0 is already generated. Because the conflict arises in the same section, generation stops, page 389 and DFB_A version v1.0 is maintained.
Facet_1	v1.0	Assigned/Generated	v1.0	The status of facets is unchanged. To be able to generate the section, you can either: <ul style="list-style-type: none"> Unassign one of the facets from the section. Update the template used by the instance referencing Facet_1 so that both facets use the same version of DFB_A, page 403. Unlink the already assigned and generated facet (Facet_1), generate the section, select Update when the Global Constituent Type Version Conflicts dialog box opens, then relink Facet_1.
(1) The order of facets in the section is not relevant.				

(Back to example summary table, page 400.)

Example 2: Two Different Versions of the Same Type in the Same Section

A generated facet is updated while the same section contains a non-generated facet.

The starting point is the following:

- Two facets are assigned to Section_1.
- One facet is generated, the other not.
- Both facets encapsulate DFB_A but of a different version.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status	DFB_A version used in Control project
Facet_0	v1.0	Assigned/Generated	v1.0

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status	DFB_A version used in Control project
Facet_1	v2.0	Assigned/Non Generated	
(1) The order of facets in the section is not relevant.			

Then, the template used by instance referencing Facet_0 is updated. This results in Facet_0 encapsulating the same version of DFB_A as Facet_1 (v2.0). The assignment status of Facet_0 is set to **Out Of Date**. The section is generated again to apply the change.

The table explains the behavior during generation of Section_1 and which version of DFB_A is used in the end.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Facet assignment/generation status after generation	DFB_A version used in Control project after generation	Comment
Facet_0	v2.0	Assigned/Generated	v2.0	The software detects v2.0 and opens the Global Constituent Type Version Conflicts dialog box. Update is selected. The facet is generated and the version of DFB_A used in the logical Control project is updated from v1.0 to v2.0.
Facet_1	v2.0	Assigned/Generated		
(1) The order of facets in the section is not relevant.				

NOTE: If you select **Skip** instead, none of the facets in Section_1 are generated and the version of DFB_A in the logical Control project remains v1.0.

(Back to example summary table, page 400.)

Example 3: Two Different Versions of the Same Type in Different Sections of a Control Project

This example illustrates the behavior during generation when facets encapsulating two different versions of the same type are assigned to various sections.

The starting point is the following:

- Four facets of four different instances are assigned to separate sections.
- Each facet encapsulates DFB_A.
- All sections are part of the same Control project and generated.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project
Facet_0	v1.0	Section_0	v1.0
Facet_1	v1.0	Section_1	
Facet_2	v1.0	Section_2	
Facet_3	v1.0	Section_3	
No facet assigned	–	Section_4	
(1) The order of the sections corresponds to the section number.			

Then, the following changes are performed:

- The template used by one instance is updated. As a result, one facet encapsulates DFB_A of a different version and its status is set to **Out Of Date**.
- A new facet, Facet_4, is assigned to Section_4. It encapsulates version v1.0 of DFB_A and its statuses are **Assigned/Non Generated**.

The Control project is generated again to apply the changes.

The table explains the behavior during generation of the Control project and which version of DFB_A is used in the end.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
Facet_0	v1.0	Section_0	v1.0	No change to be generated for the facet.
Facet_1	v2.0	Section_1	v2.0 ⁽²⁾	The software detects v2.0 and opens the Global Constituent Type Version Conflicts dialog box. Update is selected. The facet is generated and the version of DFB_A used in the logical Control project is updated from v1.0 to v2.0.
Facet_2	v1.0	Section_2		Because v1.0 has already been detected, the previously selected Update command is applied automatically and DFB_A version v2.0 is maintained.
Facet_3	v1.0	Section_3		
Facet_4	v1.0	Section_4		Facet_4 is not generated; its statuses remain Assigned/ Non Generated . NOTE: If the assignment status of Facet_2 or Facet_3 is Out Of Date , its generation status is set to Inconsistent .

(1) The order of the sections corresponds to the section number.
(2) Version of DFB_A that is used in the end after successful generation of the Control project.

(Back to example summary table, page 400.)

Example 4: Several Different Versions of the Same Type in Different Sections of a Control Project

This example illustrates the behavior during generation when facets encapsulating different versions of the same type are assigned to the various sections.

The starting point is the following:

- Four facets of four different instances are assigned to separate sections.
- Each facet encapsulates DFB_A.
- All sections are part of the same Control project and generated.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project
Facet_0	v1.0	Section_0	v1.0
Facet_1	v1.0	Section_1	
Facet_2	v1.0	Section_2	
Facet_3	v1.0	Section_3	

(1) The order of the sections corresponds to the section number.

Then, the template used by some instances is updated with various versions. As a result, the facets encapsulate various versions of DFB_A and their status is set to **Out Of Date**. The Control project is generated again to apply the changes.

The table explains the behavior during generation of the Control project and which version of DFB_A is used in the end.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
Facet_0	v1.0	Section_0	v1.0	No change to be generated for the facet.
Facet_1	v3.0	Section_1	v3.0	The software detects v3.0 and opens the Global Constituent Type Version Conflicts dialog box. Update is selected. The facet is generated and the version of DFB_A used in the logical Control project is updated from v1.0 to v3.0.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
Facet_2	v1.0	Section_2		Because v1.0 has already been detected, the previously selected Update command is applied automatically and DFB_A version v3.0 is maintained. NOTE: If the assignment status of the facet is Out Of Date , its generation status is set to Inconsistent .
Facet_3	v4.0	Section_3	v4.0 ⁽²⁾	The software detects v4.0 and opens the Global Constituent Type Version Conflicts dialog box. Update is selected. The facet is generated and the version of DFB_A used in the logical Control project is updated from v3.0 to v4.0.

(1) The order of the sections corresponds to the section number.
(2) Version of DFB_A that is used in the end after successful generation of the Control project.

NOTE: It is possible to update the version of a type to an earlier one. For example, if Facet_3 would encapsulate DFB_A of version v2.0, the version used in the end would be v2.0.

(Back to example summary table, page 400.)

Example 5: Two Different Versions of the Same Type in Different Sections of a Control Project

This example illustrates the behavior during generation when facets encapsulating two different versions of the same type are assigned to various sections and the **Skip** command is used.

The starting point is the following:

- Four facets of four different instances are assigned to separate sections.
- Each facet encapsulates DFB_A.
- All sections are part of the same Control project and generated.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project
Facet_0	v1.0	Section_0	v1.0
Facet_1	v1.0	Section_1	
Facet_2	v1.0	Section_2	
Facet_3	v1.0	Section_3	

(1) The order of the sections corresponds to the section number.

Then, the template used by two instances is updated. As a result, two facets encapsulate DFB_A of a different version and their status is set to **Out Of Date**. The Control project is generated again to apply the changes.

The table explains the behavior during generation of the Control project and which version of DFB_A is used in the end.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
Facet_0	v1.0	Section_0	v1.0 ⁽²⁾	No change to be generated for the facet.
Facet_1	v2.0	Section_1		The software detects v2.0 and opens the Global Constituent Type Version Conflicts dialog box. Skip is selected. The facet is not generated and the version of DFB_A used in the logical Control project remains v1.0.
Facet_2	v1.0	Section_2		No change to be generated for the facet.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
Facet_3	v2.0	Section_3		Because v2.0 has already been detected, the previously selected Skip command is applied automatically. The facet is not generated and DFB_A version v1.0 is maintained.

(1) The order of the sections corresponds to the section number.
(2) Version of DFB_A that is used in the end after successful generation of the Control project.

NOTE: The assignment status of Facet_1 and Facet_3 remains **Out Of Date**.

(Back to example summary table, page 400.)

Example 6: Several Different Versions of the Same Type in Different Sections of a Control Project

This example illustrates the behavior during generation when facets encapsulating different versions of the same type are assigned to the various sections and the **Skip** command is used.

The starting point is the following:

- Four facets of four different instances are assigned to separate sections.
- Each facet encapsulates DFB_A.
- All sections are part of the same Control project and generated.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project
Facet_0	v1.0	Section_0	v1.0
Facet_1	v1.0	Section_1	
Facet_2	v1.0	Section_2	
Facet_3	v1.0	Section_3	

(1) The order of the sections corresponds to the section number.

Then, the template used by some instances is updated with various versions. As a result, the facets encapsulate various versions of DFB_A and their status is set to **Out Of Date**. The Control project is generated again to apply the changes.

The table explains the behavior during generation of the Control project and which version of DFB_A is used in the end.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
Facet_0	v1.0	Section_0	v1.0 ⁽²⁾	No change to be generated for the facet.
Facet_1	v2.0	Section_1	v1.0 ⁽²⁾	The software detects v2.0 and opens the Global Constituent Type Version Conflicts dialog box. Skip is selected. The facet is not generated and the version of DFB_A used in the logical Control project remains v1.0.
Facet_2	v4.0	Section_2	v1.0 ⁽²⁾	The software detects v4.0 and opens the Global Constituent Type Version Conflicts dialog box. Skip is selected. The facet is not generated and the version of DFB_A used in the logical Control project remains v1.0.
Facet_3	v2.0	Section_3	v1.0 ⁽²⁾	Because version v2.0 has already been detected (for Facet_1), the previously selected Skip command is applied automatically. The facet is not generated and the version of DFB_A used in the logical Control project remains v1.0.

(1) The order of the sections corresponds to the section number.
(2) Version of DFB_A that is used in the end after successful generation of the Control project.

NOTE: The assignment status of Facet_1, Facet_2, and Facet_3 remains **Out Of Date**.

(Back to example summary table, page 400.)

Example 7: Several Different Versions of the Same Type in Different Sections of a Control Project

This example illustrates the behavior during generation when facets encapsulating different versions of the same type are assigned to the various sections and the **Update** and **Skip** commands are used.

The starting point is the following:

- Six facets of six different instances are assigned to separate sections.
- Each facet encapsulates DFB_A.
- All sections are part of the same Control project and generated.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project
Facet_0	v1.0	Section_0	v1.0
Facet_1	v1.0	Section_1	
Facet_2	v1.0	Section_2	
Facet_3	v1.0	Section_3	
Facet_4	v1.0	Section_4	
Facet_5	v1.0	Section_5	

(1) The order of the sections corresponds to the section number.

Then, the template used by the instances is updated with various versions. As a result, the facets encapsulate various versions of DFB_A and their status is set to **Out Of Date**. The Control project is generated again to apply the changes.

The table explains the behavior during generation of the Control project and which version of DFB_A is used in the end.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
Facet_0	v4.0	Section_0	v4.0	The software detects v4.0 and opens the Global Constituent Type Version Conflicts dialog box. Update is selected. The facet is generated and the version of DFB_A used in the logical Control project is updated from v1.0 to v4.0.
Facet_1	v2.0	Section_1		The software detects v2.0 and opens the Global Constituent Type Version Conflicts dialog box. Skip is selected. The facet is not generated and version v4.0 of DFB_A is maintained. NOTE: The generation status of the facet is set to Inconsistent because the version of DFB_A has been previously updated when generating Facet_0.
Facet_2	v1.0	Section_2		Because version v1.0 has already been detected when generating Facet_0, the previously selected Update command is applied automatically and DFB_A version v4.0 is maintained. NOTE: If the assignment status of the facet is Out Of Date , its generation status is set to Inconsistent .
Facet_3	v4.0	Section_3		The facet is generated but without the detection of a version change because the version of DFB_A used in the Control project is already v4.0 due to the update when generating Facet_0.
Facet_4	v2.0	Section_4	v4.0 ⁽²⁾	The previously selected Skip command (for Facet_1) is applied automatically. The facet is not generated and DFB_A version v4.0 is maintained.

Facet ID	DFB_A version encapsulated in facet	Section ID ⁽¹⁾	DFB_A version used in Control project after facet generation	Comment
				NOTE: The generation status of the facet is set to Inconsistent because the version of DFB_A has been previously updated when generating Facet_0.
Facet_5	v3.0	Section_5		The software detects v3.0 for the first time and therefore opens the Global Constituent Type Version Conflicts dialog box. Skip is selected. The facet is not generated and the version of DFB_A used in the logical Control project remains v4.0. NOTE: The generation status of the facet is set to Inconsistent because the version of DFB_A has been previously updated when generating Facet_0.

(1) The order of the sections corresponds to the section number.
(2) Version of DFB_A that is used in the end after successful generation of the Control project.

(Back to example summary table, page 400.)

Example 8: Two Different Versions of the Same Type in Different Sections

This example illustrates the behavior during generation when facets encapsulating different versions of the same type are assigned to the different sections and the **Cancel** and **Update** commands are used.

The starting point is the following:

- Two facets of two different instances are assigned to one section each.
- Each facet encapsulates DFB_A.
- Facets are not yet generated.

Facet ID ⁽¹⁾	DFB_A version encapsulated in facet	Section ID	DFB_A version used in Control project
Facet_0	v1.0	Section_1	None
Facet_1	v2.0	Section_2	

(1) The order of the facet in the section corresponds to the facet number.

Then, the Control project is generated.

The table explains the behavior during generation of the Control project and which version of DFB_A is used in the end.

Facet ID	DFB_A version encapsulated in facet	Section ID	DFB_A version used in Control project after generation	Comment
Facet_0	v1.0	Section_1	1.0	Version v1.0 of DFB_A is generated in the logical Control project.
Facet_1	v2.0	Section_2	1.0 ⁽¹⁾	The software detects v2.0 and opens the Global Constituent Type Version Conflicts dialog box. Cancel is selected. Generation stops. The facet is not generated and the version of DFB_A used in the logical Control project remains v1.0.

(1) Version of DFB_A that is used in the end after successful generation of the section.

Then, the Control project is generated a second time.

Facet ID	DFB_A version encapsulated in facet	Section ID	DFB_A version used in Control project after generation	Comment
Facet_0	v1.0	Section_1	1.0	No change to be generated for the facet.
Facet_1	v2.0	Section_2	2.0 ⁽¹⁾	The software detects v2.0 and opens the Global Constituent Type Version Conflicts dialog box. Update is selected. The facet is generated and the version of DFB_A used in the logical Control project is updated from v1.0 to v2.0.
(1) Version of DFB_A that is used in the end after successful generation of the section.				

NOTE: After the second generation, if you regenerate:

- The entire Control project: A version conflict is detected in Section_2 like during a first generation because type DFB_A is removed from the Control before regenerating.
- Section_1: A version conflict is detected because the version of DFB_A in the Control project (v2.0) is different from the version encapsulated in Facet_0 (v1.0).
- Section_2: No version conflict is detected.

(Back to example summary table, page 400.)

Statuses of Facets

Overview

After the generation process, the generation and assignment statuses of facets provide information on the facet state.

NOTE: For information on the behavior when generating such facets again, refer to the topic describing the generation of facets after changes, page 417.

Generation Status of Facets

The generation status indicates the status of the facet in relation to the generation of its constituents in the logical Participant project.

You can see the generation status in the **Generation** column of the **Assignments** pane, page 330 in the **Assignment Editor**.

The table describes the possible generation statuses of facets following a generation.

Status	Description
Non Generated	<p>You have assigned the facet to the FBD section or container after generating that section/container. Generate, page 417 the section/container again to create the constituents of the facet in the logical Participant project.</p> <p>It can also indicate that the generation process of a facet or its container was unsuccessful.</p>
Generated	<p>The software has successfully generated the constituents of the facet into the logical participant project.</p> <p>NOTE: The software automatically sets the status of Supervision genie facets that you assign to a page during page edit to Generated.</p>
Moved	<p>You have moved, page 378 the facet to another FBD section of the Control project or another container of the Supervision.</p> <p>NOTE:</p> <ul style="list-style-type: none"> If you move the facet back to its original FBD section or container, it keeps the status Moved. The status does not apply to genie facets.
Inconsistent	<p>You have refined, updated, or imported the logical Control Participant project, or relinked a facet, which was unlinked and the software has detected an inconsistency in the facet during the consistency check, page 449:</p> <ul style="list-style-type: none"> You have not yet generated the section containing the facet again in order for the software to resolve the inconsistency, page 419 and change the status, or The software could not solve the inconsistency during a subsequent generation and the status of the facet remains Inconsistent. <p>NOTE: By default, the status of Supervision genie and data facets changes to Inconsistent when you reassign or relink, page 431 them.</p> <p>NOTE: This status can also be the result of a type version conflict that the software detected during generation of Control facets, page 396.</p>
Unlinked	<p>You have unlinked the facet from its already generated constituents in the logical Participant project:</p> <ul style="list-style-type: none"> By using the Unlink, page 432 command. By unlinking the facet through the Check Consistency dialog box: <ul style="list-style-type: none"> After refining a Control Participant project, page 436. Before updating a Control Participant project, page 754. <p>By unlinking a facet, the software does not change already generated constituents when you generate it again if the status the facet is such that a generation would change its status (for example, if the status of the facet is out of date or inconsistent, generating it would change its status to generated). This allows you to preserve changes that you have made in the logical Participant project to constituents of the facet through refinement or refinement online.</p>

Assignment Status of Generated Facets

The assignment status indicates the status of the facet in relation to the instance that references the facet.

The software may update the **Assignment** status of the facet when you change the properties of the instance that references the facet.

You can see the assignment status in the **Assignment** column of the **Assignments** pane, page 330 in the **Assignment Editor**.

The table describes the possible assignment statuses of a facet following a generation.

Status	Description
Assigned	The facet is assigned to the container.
Unas-signed	You have unassigned, page 431 a generated facet from the container to which you had assigned it. NOTE: The facet can still be assigned to other projects of the system.
Out Of Date	You have changed at least 1 of the following parameter configurations of the instance that references the facet: <ul style="list-style-type: none"> • \$System parameters (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) that affect the facet. • Element selections and their parameters. • Instance identifier (including changes made by the software in relation to the hierarchical naming function). • Links. • Updated or replaced the template that is used by the instance. • Other changes made within the application to an instance and that affect its facet. Furthermore, for Supervision data facets only, you have changed any of the following in the Supervision project: <ul style="list-style-type: none"> • The identifier of the cluster or I/O device. • The identifier of the tag container if Protocol is set to <i>OPCUA</i>. • The relationship between the tag container and the I/O device. • The protocol of the tag container. • The application structure, page 206 (in the Application Explorer).
Deleted	You have unselected the corresponding element in the Instance Editor or deleted the instance that references the facet.

Generating a Project for the First Time

Overview

This section describes how to use the **Project Explorer** to create a logical Control or Supervision Participant project or part of a project for the first time.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Generating Entire Projects

Overview

The generation of the entire project for the first time encompasses:

- For a Control project: The FBD sections of the project and the facets that are assigned to these sections.
- For a Supervision project: The various containers of the clusters of the project to which facets are assigned and any associated components of the **Services** node.

The status of facets is updated only after the entire generation process is completed.

NOTE: You can generate a project while the generation of another project is still ongoing.

Aborting Project Generation Tasks

After selecting the **Generate** command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The project is not generated; what the software had already generated at the time you canceled the task is reverted.

Refer to the notification panel for details.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are generating a small project, the task may complete before you are able to click the icon.

Generating an Entire Project

To generate an entire Participant project, proceed as follows.

Step	Action
1	<p>In the project browser or the Containers pane of the Assignment Editor, right-click the project that you want to generate and select Generate.</p> <p>Result:</p> <ul style="list-style-type: none"> • The software creates the corresponding logical Participant project. Generation is complete when Completed is displayed for the process in the notification panel. • The status of the Generation column in the Assignments pane changes to Generated for facets that are assigned to the project. <p>NOTE: If you are generating the project again, the generation status, page 411 of facets that were Unlinked remains unchanged and those that were Inconsistent may remain unchanged.</p>

NOTE: For Control projects, you can configure certain generation settings, page 392.

Generating Sections, Containers, and Nodes

Overview

The generation of parts of a project for the first time encompasses:

- For an FBD section of a Control project: The facets that are assigned to the section and the order of the section.
- For a Supervision project:
 - For the **Containers** node: The containers of all the clusters of the project.
 - For tag containers, pages, client and sever event containers: The facets that are assigned to the respective container.
 - For the **Tags** containers: The facets that are assigned to the containers of the corresponding cluster.
 - For a cluster: The containers of the **Tags**, **ClientEvents**, and **ServerEvents** nodes of the cluster and any components of the **Services** node.
 - For the **Services** node: The components of the node.

NOTE: You can generate several sections or containers of the same project at the same time but the generation process will occur in sequence. If the sections/containers belong to different projects, they are generated in parallel.

Aborting Generation Tasks

After selecting the **Generate** command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The container of the Participant project is not generated; what the software had already generated at the time you canceled the task is reverted.

Refer to the notification panel for details.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are generating a container to which only few facets are assigned, the task may complete before you are able to click the icon.

Generating Part of a Project

To generate part of a project, proceed as follows.

Step	Action
1	<p>In the project browser or in the Containers pane of the Assignment Editor, right-click the item that you want to generate and select Generate.</p> <p>Result:</p> <ul style="list-style-type: none"> • The software generates the contents of the selected item. Generation is complete when Completed is displayed for the process in the notification panel. • The status of the Generation column in the Assignments pane changes to Generated for the facets that are assigned to the selected containers. <p>NOTE: If you are generating this part of the project again, the generation status, page 411 of facets that were Unlinked remains unchanged and those that were Inconsistent may remain unchanged.</p>

NOTE: For Control projects, you can configure certain generation settings, page 392.

Generating Projects After Changes

Overview

This section describes how to use the **Project Explorer** to perform a subsequent generation or to regenerate entire logical Control or Supervision Participant projects, or parts of such projects.

You may be able to import into the deployed Control Participant project, page 738 sections that you have generated or regenerated after modifying them.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Generating Participant Projects After Changes

Overview

You need to generate the logical Participant project that you have created with the first-time generation when you make changes that result in facets having one of the following statuses:

- **Assignment** status:
 - **Unassigned**
 - **Out Of Date**
 - **Deleted**
- **Generation** status:
 - **Non Generated**
 - **Moved**
 - **Inconsistent**

You also need to generate it when you perform the following actions on sections of a Control project in the **Project Explorer**:

- Rename a section.
- Create an FBD section.
- Change the order of sections.

To update the logical Participant project, use one of the following commands:

Generate	Only changes that you made to the application and the project since the last generation are generated incrementally into the logical Participant project. Already generated data remains unchanged. However, certain changes that you made during refinement may be discarded.
Regenerate	<p>Applies to Control projects only, page 421. Re-creates the logical Participant project from the beginning based on the current application and Control project settings. Certain changes that you made during refinement are discarded. You can configure certain generation settings, page 392.</p> <p>The software proceeds with the validation of data, page 385 like it does when generating.</p>

NOTE: To apply to the logical Control project changes that you have made to sections, you can also select the **Refine** command, page 436 at the project level.

Generation Function Description

The table indicates the actions that the software executes on facets when you perform a subsequent generation of a project or parts of it, based on the **Assignment** and **Generation** status of the facet.

Facet assignment status	Facet generation status				
	<i>Non Generated</i>	<i>Generated</i>	<i>Moved</i>	<i>Inconsistent</i>	<i>Unlinked</i>
<i>Assigned</i>	X	O ⁽¹⁾	X	X	O
<i>Out Of Date</i>	–	X	X	X	O
<i>Unassigned</i>	–	Del	–	Del	–

	Facet generation status					
Deleted	-	Del	-	Del	-	
X Creates the constituent of the facet in the logical Participant project						
O No action						
Del Removes the facet and its already generated constituents (for example, variables and FB instances) but not derived types (DFBs and DDTs)						
- Combination not applicable						
(1) Some exceptions apply. Refer to the topic describing the generation of Control projects after refining						

NOTE: When you change the order of a facet inside a section that you had generated a first time and generate it again, the software does not propagate the new order of facets, page 381 to the logical Control Participant project.

Generating Control Projects After Changing the Order of Sections

When you change the order, page 266 of already generated sections and generate again either the entire Control project or a section whose order you have changed, the software propagates the new order to the logical Control Participant project.

NOTE: Generating at the project level may take slightly longer to complete compared to generating at the section level.

Generating Control Projects After Refining

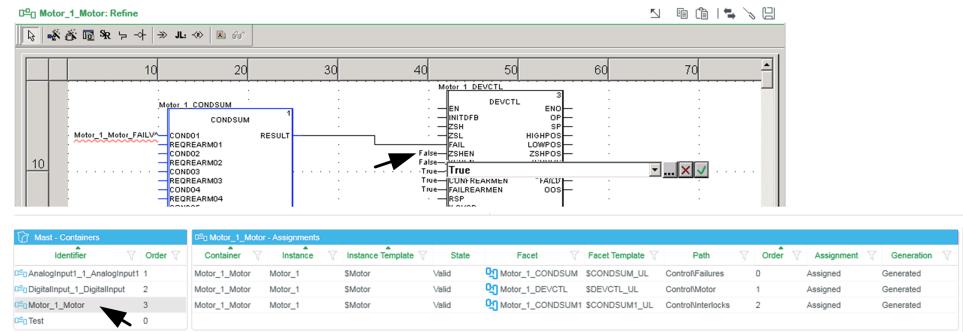
If during refinement you have modified code in a section that was already generated by the software from a facet (for example, you have modified the variable or immediate value on a pin of a DFB), the software reverts your change when you generate the section again if any of the following conditions is true; otherwise your modification is preserved:

- You assign a facet to this section.
- You unassign a facet in this section.
- You move a facet to or from this section.
- You make a modification, which sets the status of any facet assigned to this section to **Out Of Date**.
- You perform a consistency check, which sets the status of any facet assigned to this section to **Inconsistent**.

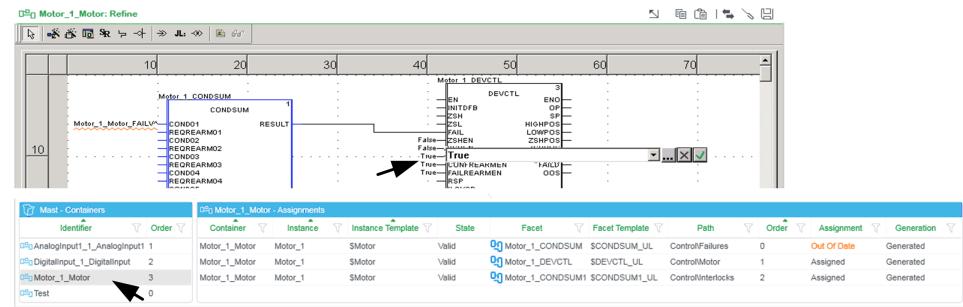
Your modification is reverted to the code that the software had previously generated from the facet.

If your modification consists in replacing a variable or immediate value on the pin of an FFB by a graphical link, such modification is not reverted by a subsequent generation. If the change is detected as an inconsistency, the generation status of the corresponding facet is set to **Inconsistent**.

In this example, the immediate value of the **ZSHEN** input parameter of the **DEVCTL** DFB is changed to *True* during refinement and saved. Section **Motor_1_Motor** that contains the facet, which generates the DFB (**Motor_1_DEVCTL**) contains only facets with the **Assigned** and **Generated** status. No other changes were made in the Control project. Generating section **Motor_1_Motor** does not undo your refinement; the immediate value of the **ZSHEN** input parameter remains *True*.



However, if section **Motor_1_Motor** contains, for example, a facet with status **Out Of Date**, generating the section undoes the refinement change, setting the **ZSHEN** input parameter again to *False*.



Generating Supervision Projects After Refining

Generating a Supervision project overrides any changes that you have made in refinement to elements pertaining to the **Services**, page 296 of the project.

Generating Inconsistent Facets

During generation, the software attempts to restore the generated constituents of facets in the logical Control Participant project that have the **Generation** status **Inconsistent** to be consistent with the related facets of the Control project.

The resulting **Generation** status of an inconsistent facet can be:

- **Generated**: The software completely restored the generated constituents while preserving the refinement that you have made.
- **Inconsistent**: The software was not able to restore the generated constituents completely because it would require altering the refinement that you have made. The software may have partially restored the constituents and the refinement that you have made was preserved.

This is the case, for example, if during refinement of an FBD section, you replace a variable by a graphical link and such change is detected as an inconsistency. Indeed, the software retains your refinement if you generate the section. Your refinement is discarded if you regenerate the section.

Generating Facets with State Invalid

If the **State** of a facet, page 366 has become **Invalid** after you generated it, generating the container or Participant project again does not remove the facet from the container.

This can be the case, for example, when you link instances A and B (where instance A needs to be linked to be valid), assign their facets to a Control project, generate the section, and then delete the link between the two instances. The facet of instance A becomes invalid because the instance is not linked anymore.

You need to either regenerate, page 421 the section to which the invalid facet is assigned or unassign the facet and generate to remove it from the section.

Function Block Layout

When you generate Control projects or sections again, the software positions function blocks that are encapsulated in facets depending on their status:

- Facets that you had already generated and that have the following status:
 - **Out Of Date:** Facets creating a new function block after a change in their properties. The function block is positioned according to the layout defined by the template referencing the facet.
 - **Inconsistent:** Function blocks encapsulated in the facet are recreated and positioned in their previously generated position.
- For both facets, if the position of the function block is already occupied, the block is moved to the end of the section content (Y axis), keeping its horizontal position (X axis) as defined by the layout of the template referencing the facet.
- **Non Generated** facets:
 - Single facets created through an element selection and that you have assigned to the section after generating it: Function blocks encapsulated in the facet are positioned at the end of the section content, on the outer left.
 - Facets referenced by an instance of which no facets were previously assigned: Function blocks encapsulated in the facets are positioned at the end of the section content while respecting the layout as defined by the template that the instance uses.

This requires that the generation of the corresponding facets completes successfully and that the resulting facet status is **Generated**.

The software manages the section size, page 390 as needed.

For more information, refer to Section Layout Management Example (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: To keep the constituents that are generated for an instance grouped, assign all its facets at once.

NOTE: Certain configurable generation settings, page 392 can affect the position of function blocks.

Generating a Project Again

Refer to the procedure describing how to generate:

- The entire project, page 414
- Part of a project, page 415

Regenerating a Control Project

Control Project Regeneration Description

This process recreates the logical Control Participant project from the beginning based on the current settings:

- Section execution order.
- Facet execution order inside the section.
- Properties of assigned facets.

The following changes made during refinement, page 436 are discarded by the regeneration process:

- Any changes to the contents of sections to which facets are assigned.
- Any contents added to such sections.
- If you set the *ReGenerateNonFbdSections* parameter to *false*:
 - Non-FBD sections that you have added during refinement and their contents.
 - Name changes of FBD sections that are created by the generation process.
 - FBD sections that you have added during refinement, their contents and any facets that you have assigned to these sections.

You can regenerate the entire Control project or FBD sections individually.

If the software was not able to regenerate a Control project or FBD sections of it, the status of facets, page 423 can vary.

Types Removed When Regenerating

The following derived types (DFBs and DDTs) are removed from the logical Control project when you regenerate it:

- Types that were generated from facets that you have unassigned from a project section or whose instance has been deleted.
- Types that remained after deleting the section that contained the facet that generated them.

NOTE: In both cases, such types are not removed when you:

- Regenerate only a section of a Control project.
- Generate the Control project.

Actions on Facets

The table indicates the actions that the software performs on facets when you regenerate a Control project or FBD section, based on the **Assignment** and **Generation** status of the facet.

	Facet generation status				
Facet assignment status	Non Generated	Generated	Moved	Inconsistent	Unlinked
Assigned	X	X	X	X	X
Out Of Date	–	X	X	X	X
Unassigned	–	Del	–	Del	–
Deleted	–	Del	–	Del	–
X Creates the constituent of the facet in the Participant project Del Deletes the facet and its already generated constituents – Combination not applicable					

Control Project Section Size

The software manages the section size, page 390.

Execution Order and Function Block Layout

The process to regenerate the section order and section layout is the same as for a first-time generation, page 390, independently of the facet status.

Regenerating Control Projects Containing Non-FBD Sections

When you regenerate a Control project that contains non-FBD sections, a message appears in the notification panel for each one of them informing you that the section was generated because of an order change. This message, which is related to an internal process, appears even if the position of these sections has not changed.

An exception applies to non-FBD sections that have rank 0 (**Order** column) when the **Regenerate** command is used. For these sections, the message does not appear whether their position has changed or not.

Aborting Regeneration of Projects And Sections

After confirming the **ReGenerate** command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The Control project or section is not regenerated; what the software had already regenerated at the moment you canceled the task is reverted.

Refer to the notification panel for details.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are regenerating a small project or section, the task may complete before you are able to click the icon.

Regenerating the Entire Control Project

To recreate the entire logical Control Participant project, proceed as follows.

Step	Action
1	Configure generation settings, page 392 as needed.
2	<p>In the project browser or the Containers pane of the Assignment Editor, right-click the Control project that you want to regenerate and select ReGenerate.</p> <p>Result:</p> <ul style="list-style-type: none"> The software re-creates the logical Control Participant project. Regeneration is complete when Completed is displayed for the process in the notification panel. The status of the Generation column in the Assignments pane changes to Generated for the facets that are assigned to the project.

Regenerating an FBD Section Only

The regeneration at the FBD section level allows you to recreate the Control Participant logic for the facets contained in the selected FBD section only.

To regenerate the logic of an FBD section, proceed as follows.

Step	Action
1	<p>In the Containers pane of the Assignment Editor, right-click the section that you want to regenerate and select ReGenerate.</p> <p>Result:</p> <ul style="list-style-type: none"> The software re-creates the Control Participant logic of the selected section only. Regeneration is complete when Completed is displayed for the process in the notification panel. The status of the Generation column in the Assignments pane changes to Generated for the facets that are assigned to the selected FBD section.

Facet Status After Unsuccessful Regeneration

The table describes the effect on the status of Control facets when the regeneration of a section is not successful.

Regenerated project component	Facet status
Entire Control project	The generation status of facets assigned to sections that are not regenerated, page 389 reverts to the value it was before regenerating.
Control project section	The generation status of facets assigned to this section reverts to the value it was before regenerating.

Generating Equipment in Supervision Projects

Generating Equipment Based on the Application

Overview

By default, generating a Supervision project populates the equipment table of the Supervision project based on the structure of the application, page 154. Each instance of the application creates one equipment instance. This allows you to view alarms during runtime in a structure identical to the hierarchy of the application.

To create equipment names, the software may truncate the identifier of folders and instances to comply with the format of the name field of the equipment database file.

Changes to the structure, page 206 of the application are reflected on generated facets of the project allowing you to update the equipment database file by generating the project again.

NOTE: To view the equipment name that is generated, refine the Supervision project and open the equipment table.

Properties of the Equipment Table

The generation process populates the following properties of the equipment table of the Supervision project.

Field	Description
<i>Name</i>	Instance identifier with its path in the application, satisfying the equipment naming rules for the name field ⁽¹⁾ . The folder and instance identifiers are dot-separated. NOTE: The system root folder identifier is not included in the name.
<i>Cluster</i>	Identifier of the cluster of the Supervision project to which facets of the instance are associated.
<i>Comment</i>	\$Description parameter of the instance.
<i>Area</i>	\$Area parameter of the instance.

(1) The Supervision Participant does not accept the use of certain reserved words for item names (for example, *DO*). While the software allows using these reserved words as identifiers of instances in the application, doing so prevents the Supervision project from generating successfully because the reserved word becomes the equipment name when the instance is created in the system root folder. For details on the reserved words, look up *Reserved Words* in the help of the Supervision Participant, page 93.

NOTE: You can use the **Equipment** element (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) to populate additional properties of equipment instances without impacting the way the equipment name is generated.

Rules Applying to the Creation of Equipment Names

The software may modify and/or truncate the identifier of folders and instances according to the following rules to populate the *Name* property of the equipment table of the Supervision project.

Item	Rule	Comment
Folder levels	7 folders maximum not including the system root folder	–
Instance path	125 characters maximum including the instance identifier and dot separators	–
Folder identifier length	12 characters maximum ⁽¹⁾	–

Item	Rule	Comment
Instance identifier length	18 characters maximum ⁽¹⁾	By default, the identifiers of instances of Schneider Electric templates are limited to 18 characters.
Identifiers of folders	Can only begin with a letter. Can only contain alphanumerical characters and underscores (_). No spaces.	Examples: <i>\$Folder1</i> and <i>Folder\$1</i> are modified to <i>Folder1</i> (or <i>Folder100</i> if <i>Folder1</i> already exists) <i>Folder 2</i> is modified to <i>Folder2</i>

(1) The software may add an indexer consisting of two digits at the end of identifiers so that the name of each folder having the same parent folder and the name of each instance inside a folder remain unique.

For more information on not allowed characters, refer to the topic describing naming rules, page 88.

NOTE: If the software truncates the instance identifier or path to satisfy the equipment name creation rules, it displays a notification once the generation process is completed.

NOTE: The software does not change the identifier of folders and instances in the application.

Example 1 - Folder Identifier Exceeds Length

If in the **Application Explorer**, the following instance exists:

- *Conveyor_Motor_1*
- At path (folder identifiers): *System_1\PlantNorthFace\Unit 20*

The generated equipment name is: *PlantNorthFa.Unit20.Conveyor_Motor_1*

Example 2 - Indexing the Truncated Folder Identifier

If in the **Application Explorer**, the following two instances exist in two distinct folders:

- *Instance_1* at the path *System_1\Folder_12345_1* (the application folder name has 14 characters)
- *Instance_2* at the path *System_1\Folder_12345*

The generated equipment names are:

- *Folder_1234500.Instance_1*
- *Folder_12345.Instance_2*

Example 3 - Number of Folders Exceeds Limit

If in the **Application Explorer**, the following instance exists:

- *Motor_1*
- At path (folder identifiers): *System_1\F1\F2\F3\F4\F5\F6\F7\F8\F9*

The generated equipment name is: *F1.F2.F3.F4.F5.F6.F7.Motor_1*

Generation Diagnostic Messages

Generation Diagnostic Messages

Overview

The following tables describe some of the messages that the software displays when applicable rules are not satisfied during the generation of Control and Supervision projects and propose corrective actions.

In these messages and others, notifications, page 327 that are generated by the Control Participant may be displayed. These notifications provide more detailed information about the reason for which the generation process did not complete successfully.

For Control Projects

Message details	Cause	Corrective action
Section is overcrowded	<p>The section cannot accept as many function blocks as the software is trying to generate inside it, page 390.</p> <p>Also, you may have changed the position of DFBs during refinement so that the software cannot adapt the size of the section to accommodate the function blocks.</p>	<p>Do the following to free up space:</p> <ul style="list-style-type: none"> Move or unassign facets from the FBD section. Change the layout of the function blocks inside the section. Verify whether a comment box is preventing the software from adapting the size of the section, page 390. <p>Then, generate the project/section again</p> <p>NOTE: When you move DFBs inside a section, use either method:</p> <ul style="list-style-type: none"> Position DFBs tightly together to use the full width of the grid effectively, leaving empty rows at the bottom of the section. Position DFBs to use the entire height of the grid, leaving empty columns on the right side of the section. You can change the section size, page 392 by regenerating it.
Variable name conflict	A facet is generating a variable that has the same name as an already existing variable in the project.	<p>You can do either of the following:</p> <ul style="list-style-type: none"> Change the instance <code>\$Name</code> parameter, page 176 so that the facet creates a variable with a different name. Then, generate the project again. If you had already generated the project, refine the project and rename or remove the existing variable. Then, generate the project again. Regenerate the Control project, page 421 in order to remove the existing variable (for example, if you had created the variable during refinement).
Version conflict in the same section	<p>The software is trying to generate, within the same section, a version of a DDT and/or DFB that is different from the version that already exists in the Control project.</p> <p>You may have updated the template used by an instance, page 216.</p>	<p>You can do either of the following:</p> <ul style="list-style-type: none"> Update the template used by instances whose facets are assigned to the section so that only one version of the same type is generated. Remove the type from the project if it is not used anymore. Assign the facet that is generating the new version of the type to another section. <p>Generate the project again.</p> <p>For details, refer to the topic describing the management of type versions in Control facets during generation, page 396.</p>
Invalid characters	The name of a generated constituent contains a character that is invalid for use with the Control Participant.	Edit the properties of the instance referencing the facet that is mentioned in the message details so that resulting variable names and section identifiers satisfy the naming rules, page 89, and generate the project again.

For Supervision Projects

Detected error	Cause	Corrective action
Unique equipment name cannot be generated. Indexer used for uniqueness has reached 99 (maximum limit).	The number of non-unique truncated folder and instance identifiers during generation exceeds 99.	Change the folder structure of the application or instance identifiers to reduce the number of non-unique identifiers that the software creates through truncation, page 425.

Managing Generated Facets

Managing Generated Facets

Overview

Following the generation of the Control or Supervision project, or parts of it, you can execute commands and actions on generated facets from the **Assignments** pane of the **Assignment Editor**.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Generated Facets Actions

The table outlines the commands and actions that you can execute on generated facets.

Command or action	Description
Unassign	<p>You can unassign facets that have the following Assignment status:</p> <ul style="list-style-type: none"> • Assigned • Out Of Date <p>If you unassign a Control facet that has the generation status of Unlinked, the facet is removed from the section but the generated constituents pertaining to the facet are not deleted from the logical Control Participant project. If you assign this facet again to the same project, you may receive a notification about a variable name conflict, page 428.</p> <p>The status UnAssigned is described in Assignment Status of Generated Facets, page 411.</p> <p>Refer to Unassigning Generated Facets, page 432.</p> <p>NOTE: The actions that the software executes when you select the Unassign command for a generated facet or a facet that is not yet generated, page 378 are different.</p>
Unlink	<p>Refer to Unlinking Generated Facets, page 432.</p> <p>NOTE: The status Unlinked is described in Generation Status of Facets, page 411.</p>
Reassign	<p>You can reassign generated facets that have the status UnAssigned.</p> <p>The Reassign command assigns a facet to the container in which it is located, allowing you to:</p> <ul style="list-style-type: none"> • Apply to the facet changes to the instance that references it. • Generate its constituents into the logical Participant project. <p>The software performs a consistency check, page 449 when available.</p> <p>When the consistency check is not available, the generation status of facets changes to Inconsistent. This is the case, for example, for Supervision genie and data facets.</p> <p>Reassigning a facet changes its status to Assigned.</p> <p>Refer to Reassigning Facets, page 432.</p>
Relink	<p>You can relink generated facets that have the status Unlinked.</p> <p>The Relink command links a facet again to its already generated constituents, allowing you to:</p> <ul style="list-style-type: none"> • Apply to the logical Participant project changes of the facet using the Generate command. • Detect inconsistencies by performing a consistency check. <p>When you relink a facet, the software performs a consistency check if the functionality is available for the facet. Depending on the result, it changes the generation status of the facet to either:</p> <ul style="list-style-type: none"> • Generated if it detects no inconsistency. • Inconsistent if it detects an inconsistency. <p>When the consistency check is not available, the generation status of facets changes to Inconsistent. This is the case, for example, for Supervision genie and data facets.</p>

Command or action	Description
	Refer to Relinking Facets, page 433.
Go To Instance	Allows you to go to the instance that references the facet. The navigation feature is not available if the Assignment status of the facet is Deleted . Refer to Navigating From Facets to Instance, page 382.
Move	Refer to Moving Generated Facets, page 433. NOTE: The status Moved is described in Generation Status of Facets, page 411.

NOTE: For a description of the effect of a subsequent generation process on the facets that you have acted upon, refer to Generating Projects After Changes, page 416.

Unassigning Generated Facets

To unassign a generated facet from the container to which it is assigned, proceed as follows.

Step	Action
1	<p>In the Assignments pane, right-click the facet and select Unassign.</p> <p>Result:</p> <ul style="list-style-type: none"> The software unassigns the facet from the selected container. Its already generated constituents (for example, variables and FB instances for Control facets and database entries for Supervision facets) are removed from the logical Participant project (except derived data types, page 421 for Control facets). The status of the facet displayed in the Assignment column changes to UnAssigned. <p>NOTE: You can select several facets in the same container.</p>

unlinking Generated Facets

To unlink a facet, proceed as follows.

Step	Action
1	<p>In the Assignments pane, right-click the facet and select Unlink.</p> <p>Result:</p> <ul style="list-style-type: none"> The software unlinks the facet from its already generated constituents. The status of the facet displayed in the Generation column changes to Unlinked. <p>NOTE: You can select several facets in the same container.</p>

Reassigning Facets

To reassign a facet, proceed as follows.

Step	Action
1	<p>In the Assignments pane, right-click the facet and select Reassign.</p> <p>Result:</p> <ul style="list-style-type: none"> The software reassigns the facet to the same container and performs a consistency check. The status of the facet displayed in the Assignment column changes to Assigned or Out Of Date. The status of the facet displayed in the Generation column changes to Generated or Inconsistent, page 411. <p>NOTE: You can select several facets in the same container.</p>

Relinking Facets

To relink a facet, proceed as follows.

Step	Action
1	<p>In the Assignments pane, right-click the facet and select Relink.</p> <p>Result:</p> <ul style="list-style-type: none"> The software links the facet again to its generated constituents in the logical Participant project. The status of the facet displayed in the Generation column changes to Generated or Inconsistent, page 411. <p>NOTE: You can select several facets in the same container.</p>

Moving Generated Facets

To move a facet from one container to another, proceed as follows.

Step	Action
1	<p>In the Assignments pane, right-click any one of the column headers and select the appropriate command to clear filters and sorting, page 118.</p>
2	<p>In the Assignments pane, drag the facet to another container in the Containers pane.</p> <p>Result:</p> <ul style="list-style-type: none"> The software removes the facet from its original location and assigns it to the target container. The status of the facet displayed in the Generation column changes to Moved. <p>NOTE: You can select several facets in the same container.</p>

NOTE: Control facets that you move are assigned to the new section with the highest order value in the **Order** column of the **Assignments** pane, page 366. If you move several facets at once, the facet that you have selected first gets the lowest order out of the ones that are moved.

NOTE: After you have moved a generated facet from one container to another, generate, page 417 the target container again to update the associated logical Participant project.

Refinement Stage

What's in This Chapter

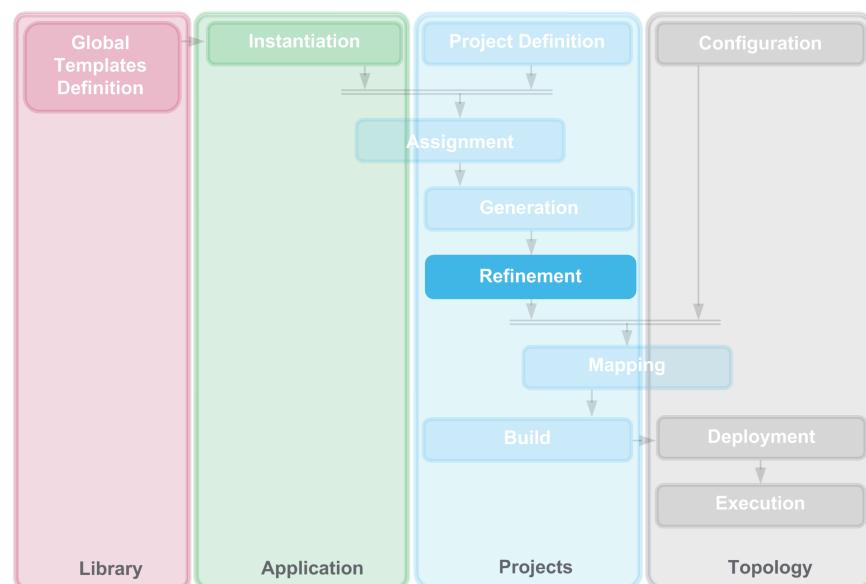
Control Project Refinement Stage.....	435
Managing Peer to Peer Communication	462
Supervision Project Refinement Stage	493

Overview

This chapter describes how to modify the logical Participant projects that the software has created during the generation stage to fulfill the requirements of the system.

The refinement stage is not mandatory to be able to proceed to the next stage.

The following figure shows the position of the **Refinement** stage within the system engineering life cycle.



Refer to Refinement Stage, page 47 for a description of the purpose of this stage.

Control Project Refinement Stage

Refining the Logical Control Project

Overview

You can refine at the project level, which allows you to view and modify the entire logical Control project with the help of the Control Participant.

You can also refine specific sections only by launching the refinement process from a section (except for LL984 sections). The section does not need to be generated to be refined.

You can import LL984 sections (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

FBD and non-FBD sections that you add during refinement are displayed in the Control project (**Assignment Editor**) in the order they appear in the Control Participant. The contents of these sections is not displayed there however.

Once refinement is complete, you can request the software to perform a consistency check, page 449.

At the refinement stage, you can also create and manage network variables, page 452, which are used for peer to peer communication.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Refining Offline With CCOTF Enabled

For M580 and M580 safety Control projects created in the **Project Explorer**, the configuration change on the fly (CCOTF) functionality of the Control Participant is enabled by default. That is, the *Online modification in RUN or STOP CPU* parameter of the default configuration of the Control project is selected.

To complement CCOTF for offline controller configuration, page 505, when you refine a Control project offline, the software informs you ahead of time if a modification that you are about to make will require stopping the controller when you deploy changes, page 682 to the controller later on. This information is available again in the **Impact on Engine** column of the **Deploy Changes / Undo Online Changes** window where modifications that can be deployed are listed.

The working principle is the same as for CCOTF when configuring a controller except for the following:

- If you have performed a CCOTF-incompatible modification, save, and close the Control Participant window, the next time you refine the same project offline, you continue receiving notifications about CCOTF-incompatible modifications. This, even if you have not yet deployed the last CCOTF-incompatible modification.
- Performing a CCOTF-incompatible modification does not disable the **Build** nor the **Deploy Changes / Undo Online Changes** command.

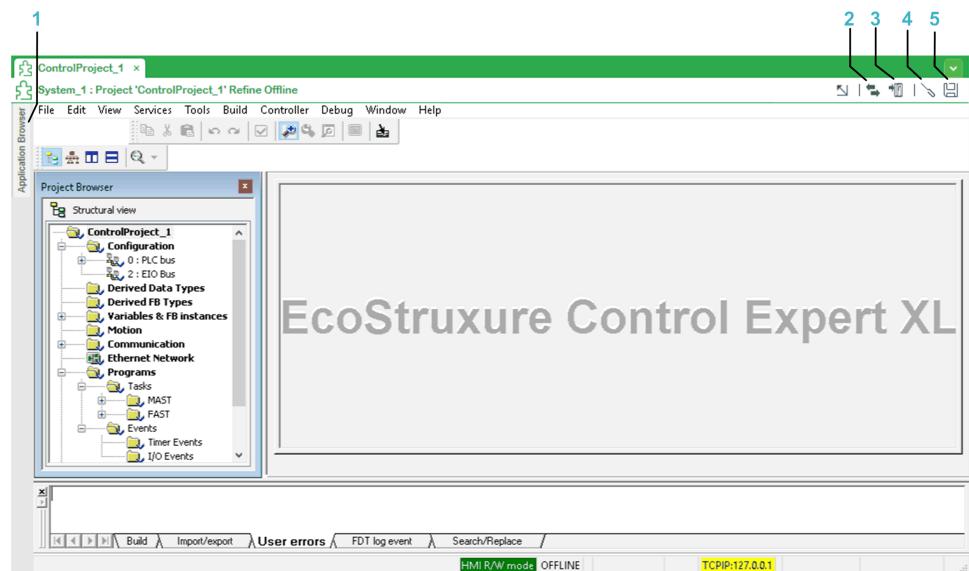
Before performing a CCOTF-compatible modification, refer to the topics describing the CCOTF functionality in the help of the Control Participant for your specific controller platform and follow the recommendations that are given.

NOTE: CCOTF is not supported for Control projects the M340 and Quantum controller families.

NOTE: CCOTF notifications are shown even if you have not yet deployed the Control project.

Project Refinement Window

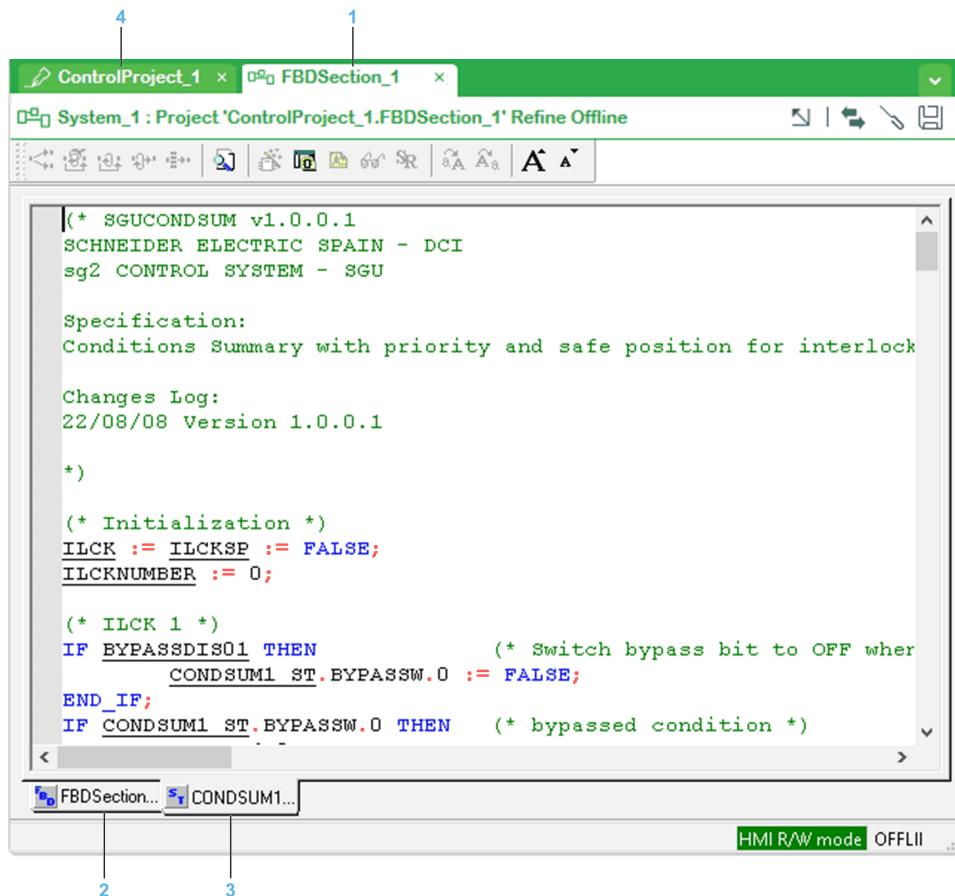
The figure shows the **Refine Offline** window embedding the Control Participant, which opens when you select the **Refine** command at the Control Participant project level. The window is shown restored.



Item	Description
1	Tab of the pane in which you can see the project sections in relation to the structure of the application
2	Manage network variables button
3	Access device DDTs of unmapped hardware button
4	Check consistency button
5	Save button

Section Refinement Window

The figure shows the **Refine Offline** window embedding the Control Participant, which opens when you select the **Refine** command at the section level. The window is shown restored.



Item	Description
1	Tab of the Control Participant window.
2	Tab of the section refinement window.
3	Tab of another refinement window (for example, a DFB refinement window), which you have opened from the section that is being refined.
4	Tab of another window that is open in the Project Explorer .

NOTE:

- To switch between open refinement windows, use the tabs at the bottom of the Participant window.
- To close a refinement window, right-click its tab and select **Close Window**.

Viewing the Application Structure When Refining Control Participant Projects

While you refine a Control Participant project, the **Application Browser** pane displays FBD sections based on the location of instances whose Control facets are assigned to these sections.

It shows the relationship between the folder structure of the application and the sections of the Control Participant project. This is useful, for example, when you want to view the logic related to a specific area or subdivision of the application. You can open sections from the pane.

The pane only shows FBD and non-FBD sections that exist in the logical Control Participant project at the moment you select the **Refine** command.

For a general description of the contents of the pane, refer to the topic describing the application structure view, page 371.

For information on how to use the pane, refer to the topic describing the refinement of Control Participant projects (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

For general information on how to work with panes, refer to the topic describing the engineering client workspace, page 111.

Applying Changes to Generated Control Participant Projects

Selecting the **Refine** command applies to the logical Control Participant project the following changes that you have made in the **Project Explorer** to an already generated Control project:

- Renaming a section.
- Creating an FBD section.
- Changing the order of sections.

If the status of the executable of the Control project is **Built**, it is set to **Out Of Date**.

NOTE: You do not need to click the save button in the Control Participant window to apply the above mentioned changes.

Validity Indicators

When you save the logical Control Participant project during refinement and project analysis is enabled, page 263, the project is analyzed, page 388. The status of the project is indicated by the validity icons, page 324.

Details of the analysis are shown in the **Analyze Project** tab of the Control Participant window while it is still open.

If the logical Control project is not open in the Participant and the validity icon is displayed, you need to refine the Control project and analyze it. This allows you to identify the cause of the invalid status. The Control Participant features tools that help you identify validity issues, which result in an unsuccessful build, page 632.

For information on troubleshooting Control projects and/or analyzing them, refer to the help of the Control Participant.

NOTE: This verification is performed also when you generate a logical Control project, page 388.

Deleting Sections During Project Refinement

The table describes the result when you delete sections during refinement and the action that you need to perform to complete the deletion process depending on the type of section.

Type of section	Result after saving changes and closing the Control Participant window	Required action
Non-generated FBD section that was created in the Project Explorer .	The section still appears in the Containers pane but below other existing sections (higher order value) ⁽¹⁾ .	Right-click the section in the Containers pane and select Delete .
Non-generated FBD section that was created during refinement.		
Any non-FBD section.		
(1) LL984 segments deleted during refinement appear in last position (highest order value)		

Refining Non-Generated Sections

When you refine an FBD section to which facets are assigned but that is not generated yet, you do not see the constituents of these facets in the Control Participant.

If you add logic to this section during refinement, when you generate the section or project, page 388, DFBs that you have added are not moved, and DFBs coming from assigned facets are laid out around them.

Refining at the Control Project Level

To refine the logical Control project, proceed as follows.

Step	Action
1	In the Project Explorer , right-click the Control project you want to refine and select Refine . Result: The Control Participant opens extracted, page 116 and maximized.
2	Proceed with the refinement. For more information on the refinement process, refer to Refining the Logical Control Participant Project (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
3	Click the save button in the toolbar of the Participant window to save the changes in the logical Control Participant project. 
4	Perform a consistency check, page 450 if required.
5	Click the close button in the Control Participant window. NOTE: If you have unsaved changes and click the close button in the Participant window, the Save Refinement dialog box opens where you have the following choices: <ul style="list-style-type: none"> • Yes: Saves changes in the logical Control Participant project and closes the Participant window. • No: Discards changes and closes the Participant window. • Cancel: Does not save changes and leaves the Participant window open.

Refining at the Section Level

To refine a section of the Control project (except for LL984 sections), proceed as follows.

Step	Action
1	In the Containers pane of the Assignment Editor , right-click the section that you want to refine and select Refine . Result: The Control Participant opens extracted, page 116 and maximized and displays only the contents of the selected section.
2	Proceed with the refinement of the section. For more information on the refinement process, refer to Refining the Logical Control Participant Project (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
3	Click the save button in the toolbar of the Participant window to save the changes in the logical Control project. 

Step	Action
4	Perform a consistency check, page 450 if required.
5	Click the close button in the tab of the Control Participant window. NOTE: If you have unsaved changes and click the close button in the Participant window, the Save Refinement dialog box opens where you have the following choices: <ul style="list-style-type: none">• Yes: Saves changes in the logical Control Participant project and closes the Participant window.• No: Discards changes and closes the Participant window.• Cancel: Does not save changes and leaves the Participant window open.

Identifying Code Generated by the Software

Overview

Constituents that are generated from facets assigned to Control projects are locked in the Control Participant project. The locked code is color coded in the DFB editor of the Control Participant.

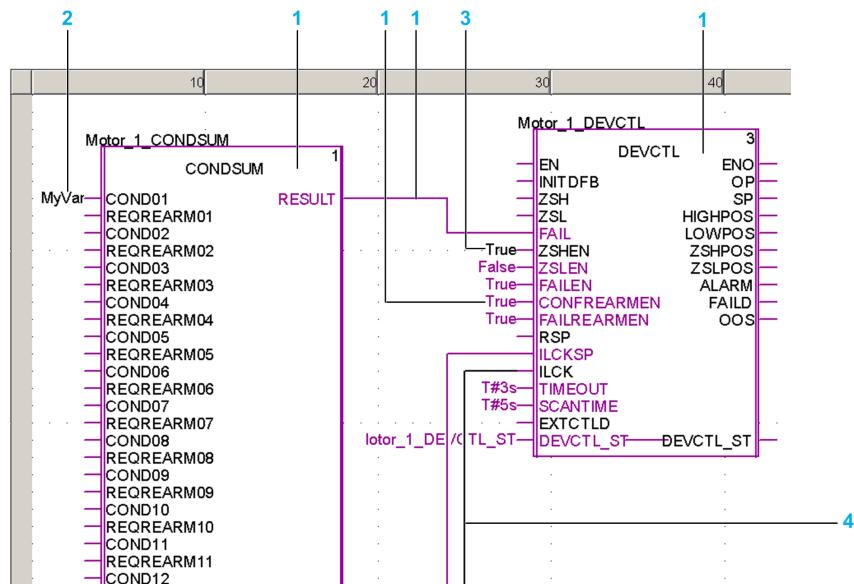
This feature helps you with the following:

- Distinguish code generated by the software from code that you modify and/or add during refinement.
- Avoid modifying code generated by the software inadvertently and maintain consistency. This is achieved by requiring a confirmation before unlocking code and providing the possibility to lock code again.

Locking of code is managed by the software throughout the system engineering life cycle.

Viewing Locked Code

The following figure shows an example of a Control Participant refinement window showing an FBD section, which contains code that was generated by some sample facets assigned to the section.



Item	Description
1	Code that is generated from the facet is locked and shown in purple.
2	Variable that was added by the user during refinement to code generated by the software. The code is not locked and shown in black.
3	Literal, previously shown in purple that was unlocked and changed by the user and now shown in black.
4	Link, previously shown in purple that was unlocked by the user and now shown in black.

Code that Is Locked

The following data is locked when generated from facets assigned to a Control project:

- DFBs and the pins to which an effective parameter is assigned.
- Effective parameters.
- Links (including links created by using an editor of the **Application Explorer**).
- EFB and DFB instances.

NOTE: FBD sections that you have generated are also locked but without visual indication.

Code that Is not Locked

The following data is not locked:

- Non-FBD sections.
- Pins of DFBs to which no effective parameter is assigned or that are not connected.
- Formal parameters of DFBs.
- Variables shown in the **Data Editor** of the Control Participant.
- Any code added by the user during refinement offline and online.
- FBD sections that are generated but to which no facets are assigned.

Modifying Locked Code During Refinement

To modify locked code during refinement offline and online, you need to unlock it first (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

For example, changing the parameter that is assigned to the pin of a generated DFB.

However, adding code that does not modify generated code can be done without unlocking. For example, assigning a parameter to an empty pin of a generated DFB or adding an EFB in a section containing generated code.

NOTE: You cannot delete FBD sections that are locked.

Management of Locked Code in the System Engineering Life Cycle

The fact that code is locked is part of its attributes and managed by the software during the system engineering life cycle.

As such, unlocking code is detected as an inconsistency during a consistency check, page 449 even if the code itself is unchanged. The status of the facet that generated the code is set to **Inconsistent**.

Code that was originally locked and that you have unlocked is:

- Locked again when the facet is generated and either condition applies:
 - The status of the facet is **Inconsistent**.
 - No consistency check was performed but the contents of the section was modified, page 418.
- Detected when you use the following commands:
 - **Deploy Changes / Undo Online Changes**
 - **Update Project**

Considerations when Migrating Databases

When you refine a Control project of a database that was migrated from a previous version of the software, code generated by the software may not be locked.

However, you can enable the code locking feature for such code in a migrated database if you perform a consistency check and generate the Control project.

Code generated from facets that are assigned to Control projects after the migration is locked.

The table describes the impact on code that is generated by the software and contained in a migrated database depending on the action that you perform and the status of facets.

Actions performed after migration	Status of facets assigned to Control projects in the migrated database	Result after generating the Control projects
No consistency check was performed.	Generated or Unlinked.	No change.
	Non Generated, Out Of Date, or Inconsistent.	The code of facets that were successfully generated is locked. FBD sections containing at least one such facet are locked.
A consistency check is performed.	All facets are set to Inconsistent .	The code of facets that were successfully generated is locked. FBD sections containing at least one such facet are locked.

NOTE: If you use the **Regenerate** command on the Control project after migrating the database, code that was generated by the software before the migration is locked even if you have not performed a consistency check.

Creating and Updating Device DDT Variables in Logical Control Projects

Overview

In the system engineering life cycle, the software lets you engineer Control projects independently of the hardware that exists in the topology up to the build stage. As a result, device DDT variables, which are associated to hardware modules and managed by the Control Participant are not readily available when you refine a logical Control project offline.

This functionality lets you create in the logical Control project device DDT variables that are associated to the hardware present in the topology of the system. This makes it possible to use device DDT variables in the program during refinement offline, for example, to manage special modules or access hardware diagnostic data.

Using device DDT variables does not replace hardware mapping. Device DDT variables do not appear in the **Hardware Mapping Editor**.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Device DDT Variables Present in the Control Project by Default

When you create a Control project of the **M580** platform in the **Project Explorer**, it contains a default configuration with only a controller (although it is not to be used from the **Project Explorer** at this stage of the system engineering life cycle). Therefore, by default, the logical Control Participant project already contains the device DDT variable that is mapped to the M580 CPU module (a lock icon appears next to the DDDT variable). You can use it by refining the logical Control project.

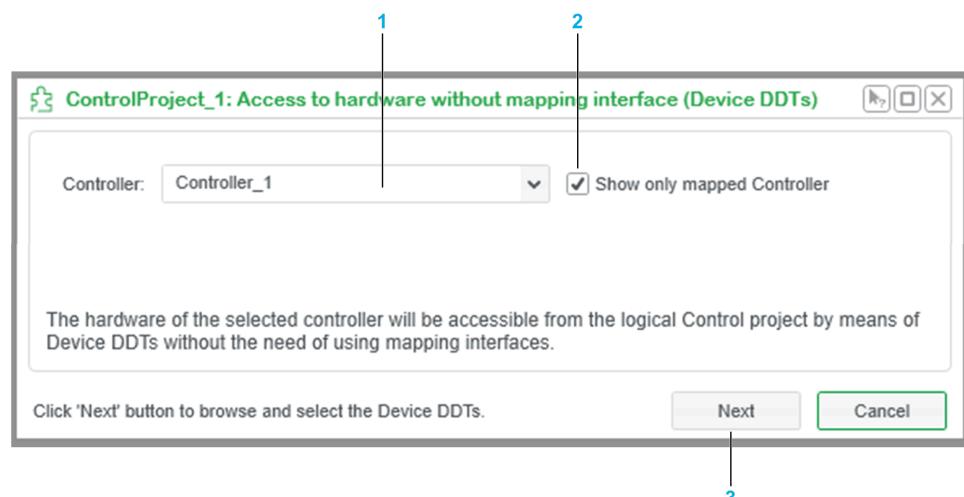
Prerequisites

The topological entities whose device DDT variables you want to create must exist in the topology of the system.

Accessing the Device DDT Variable Wizard

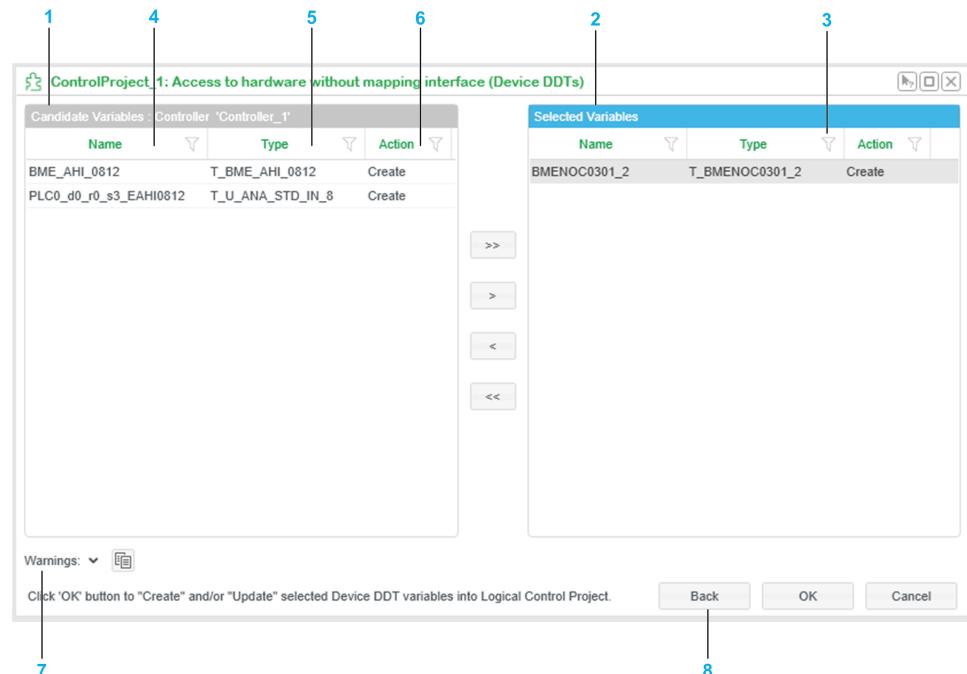
To open the wizard, refine the Control project and click the device DDT wizard button, page 437.

Device DDT Variable Wizard - Controller Selection



Item	Description
1	Box to select the controller whose device DDT variables you want to create in the logical Control project.
2	When the check box is selected, the menu shows only the controllers that are service mapped to an executable of the Control project. Otherwise, it shows the controllers that exist in the topology of the system.
3	Proceeds to the next screen and locks the controller in the Topology Explorer so that no other user can change its configuration or delete it.

Device DDT Variable Wizard - Device DDT Variable Selection



Item	Description
1	Candidate pane that shows device DDT variables of the controller that you have selected and that you can create in the logical Control project. The following can be shown: <ul style="list-style-type: none"> Variables that do not exist in the logical Control project yet. Variables and types that already exist but which have been modified in the controller. NOTE: The following variables do not appear in the candidate pane: <ul style="list-style-type: none"> Device DDT variables of the M580 CPU, page 445. Device DDT variables that already exist in the logical Control project with the same name but with a different type. This can be the case, for example, if after having created the device DDT variable for hardware module 1, you remove this module in the topology, add hardware module 2, which creates a device DDT variable of a different type, and rename this device DDT variable to have the same name as the variable that was associated to hardware module 1.
2	Selected variables panes that shows the device DDT variables that you have selected.
3	Filter icon, page 118.
4	Name of the device DDT variable.
5	Data type of the device DDT variable.
6	Either action appears for each device DDT variable: <ul style="list-style-type: none"> Create: Creates, page 447 the data type and the variable in the logical Control project. Update: Updates, page 447 the data type in the logical Control project.

Item	Description
7	Notification area. Expand the area to view alerts on device DDT variables of the selected controller.
8	Reverts to the previous screen. To be able to select device DDT variables from another controller, you need to complete the process for your current selection first. Any selection that you have not applied is discarded.

Impact of the Create Action on the Logical Control Project

The table describes the impact when you create in the Logical Control Project a device DDT variable that appears in the candidate pane with the **Create** action. The impact varies depending on the scenario.

Scenario	Impact of the Create Action
The variable and the type are not yet present in the logical Control project.	Both are created in the logical Control project.
A type of the same name already exists in the logical Control project but not the variable.	The variable is created. If the structure of the type of the selected variable is different, it overwrites the type that is present in the Control project. This may impact other existing variables of that type.

Impact of the Update Action on the Logical Control Project

The table describes the impact when you update in the Logical Control Project a device DDT variable that appears in the candidate pane with the **Update** action. The impact varies depending on the scenario.

Scenario	Impact of the Update Action
The variable and the type already exist in the logical Control project but the structure of the selected type is different.	The structure of the type is updated in the logical Control project. This may impact other existing variables of that type.
The variable and the type already exist in the logical Control project but the version of the selected type is different.	The version of the type is updated in the logical Control project. This may impact other existing variables of that type.

Creating or Updating Device DDT Variables

To create or update device DDT variables in the logical Control project, proceed as follows.

Step	Action
1	In the device DDT variable wizard, select the controller whose device DDT variables you want to create or update and click Next .
2	Expand the notification area to view alert messages.
3	In the candidate pane, select one or more device DDT variables that you want to create/update and click OK . Result: The selected device DDT variables are processed by applying the corresponding action for each and the wizard is closed.
4	Save changes by clicking the save button, page 437 in the toolbar of the Control Participant window frame.

Modifying Device DDT Variables

Device DDT variables that you create in the logical Control project are not locked and you must not modify their structure or name. If you do, the software cannot

treat them anymore as device DDT variables associated to the device from which they were created originally.

Deleting Device DDT Variables

You cannot remove a device DDT variable and/or its data type from the logical Control project by using the device DDT variable wizard.

To remove the variable, select it in the Control Participant window while refining the logical Control project and select **Delete**. Then, purge the type by using the **Types Library manager**.

Considerations When Removing Modules in the Topology Explorer and Deploying Changes

When the topology of the system contains entities and you had created the corresponding device DDT variables in the logical Control project, built the project, and deployed it, if you delete the source of the device DDT variable from the topology (for example, a module or DTM), verify that the corresponding device DDT variable is removed in the configuration of the **Topology Explorer**. If not (for example, when you delete an analog input module), you need to purge the corresponding device DDT variable.

Otherwise, if you build the logical Control project after removing the source of the device DDT variable from the topology, it remains in the built Control project as unmapped variable (while it is present in the deployed Participant project as mapped variable). If you deploy changes, the software proposes to add the unmapped device DDT variable.

Integrating Device DDT Variables in the System Engineering Life Cycle

To use device DDT variables that you have created logical Control project, follow the stages of the system engineering life cycle described in this manual (build, page 631, build changes, page 637, deployment, deployment of changes, page 687, refinement online, and so on).

Using Device DDT Variables in the Supervision Runtime

Device DDT variables do not have the HMI attribute enabled. To use such a variable in Supervision, you need to connect it to the input pin of a DFB encapsulated in a template that features Supervision services.

Managing the Consistency of Control Logic

Overview

Consistency management is a process that encompasses the following steps:

- Launching the consistency check of a logical Control Participant project or FBD section of such project.
- Indicating if the constituents of facets that the software has generated in such project or FBD section are not consistent with the related facets that are assigned to the Control project or section. In such case, a message informs you of the detected inconsistencies and the **Generation** status of the related facets is changed to **Inconsistent**. Otherwise, the software informs you that it did not detect an inconsistency.
- Attempting to fix the detected inconsistencies by restoring the constituents inside the logical Control Participant project or FBD section to a state that is consistent with the related facets of the Control project or section.

The purpose of consistency management is to provide means for you to know whether changes that you have made during refinement affect the logical Control Participant project that was created during generation and, if desired, to fix inconsistencies when it is possible without modifying the refinements you made.

The software performs a consistency check in the following way when you:

- Generate again an already generated Control project or FBD section:
Depending on the changes you have made, page 418.
- Reassign, page 431 a facet with the status **UnAssigned**: Automatically (if available).
- Relink, page 431 a facet with the status **Unlinked**: Automatically (if available).
- Update the logical Control Participant project, page 748: On request.

NOTE: The software does not perform consistency checks on facets that have the **Unlinked** status, page 411.

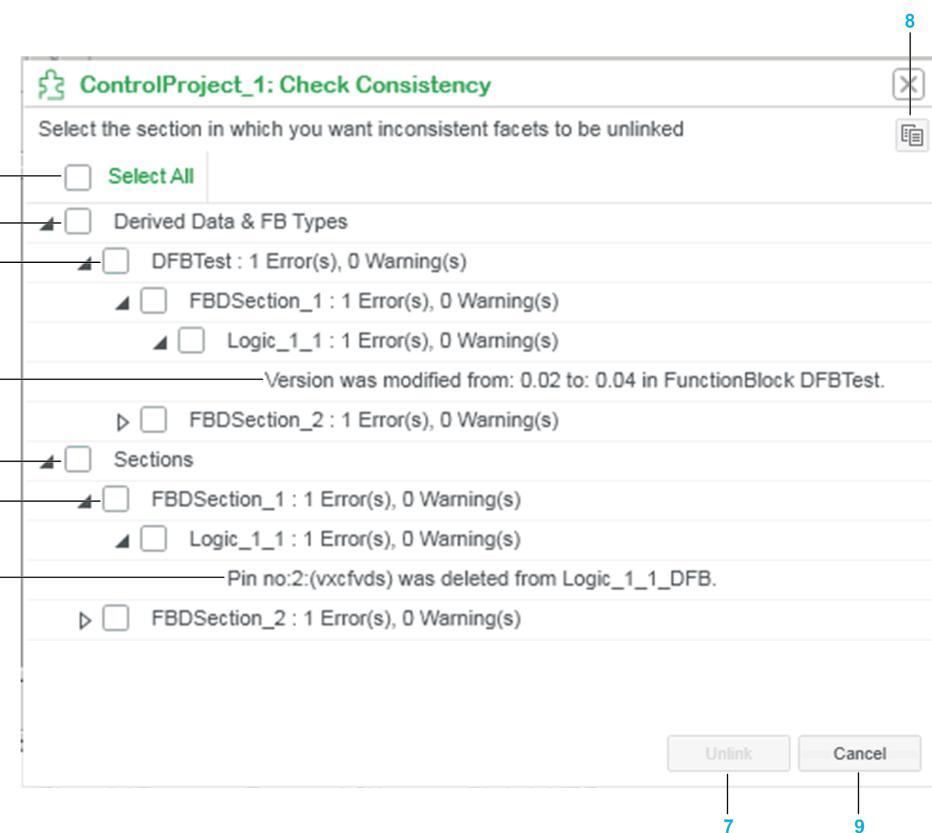
Rules for Consistency Checks

The table indicates in which cases the software detects and reports an inconsistency when performing a consistency check on constituents that the software has generated in a logical Control Participant project. It also indicates the resulting status of the facet encapsulating the constituent.

Type	Change made through refinement	Generation status of the facet
Derived data (DDT) and function block (DFB) types	Any change resulting in a new version of the type ⁽¹⁾ .	Changes to Inconsistent .
	Change causing no version change but resulting in a new type or code signature ⁽¹⁾ .	
Variables	Changing an attribute of a variable.	
	Removing markers created by the software (for example <i>P2P</i>).	
FBD sections	Changing function calls (EFs, EFBs, or DFBs) in an FBD section.	No change.
	Changing the relative execution order of function blocks (EFs, EFBs, or DFBs).	
	Changing links. Includes changing immediate values and links between variables.	Changes to Inconsistent .
–	Unlocking locked data, page 442.	Changes to Inconsistent .
<p>⁽¹⁾ The change may also have been made by encapsulating a modified type in a Control facet and updating the template with this facet. However, starting with Process Expert 4.2 SP2, the software does not allow you anymore to save modifications to the type encapsulated in a Control facet if the version of the type is still the same but its type or code signature different.</p>		

Check Consistency Dialog Box

The figure shows an example of the **Check Consistency** dialog box.



Item	Description
1	Check box to select all FBD sections and facets that appear in the two categories of the dialog box. By default no section or facet is selected.
2	Category listing FBD sections containing facets for which inconsistencies related to their DDT and DFB types have been detected. Select the check box to select all sections and facets that appear in this category.
3	Category listing FBD sections in which inconsistent facets have been detected. Select the check box to select all sections and facets that appear in this category.
4	Name of a type for which an inconsistency has been detected. Expand the node to view the facets generating this type and the sections to which these facets are assigned.
5	Identifier of a section to which facets are assigned, which are inconsistent. Expand the node to view the facets.
6	Description of the detected inconsistency.
7	Unlinks the selected facets so that no consistency check is performed on them.
8	Copies the content of the dialog box to the Clipboard in text format. The status of check boxes is not copied.
9	Cancels your selection and closes the dialog box.

NOTE: When a section or facet appears in both categories, selecting it in one category automatically selects it in the other, and the other way around, clearing its selection in one category automatically clears it in the other.

Performing a Consistency Check

To perform a consistency check on generated constituents, proceed as follows.

Step	Action
1	In the Project Explorer , right-click the Control project or FBD section that you want to check for consistency and select refine . Result: The Control Participant opens in a reduced window.
2	In the toolbar of the window, click the check consistency button  . Result: The software: <ul style="list-style-type: none"> • Performs a consistency check. • Opens the Check Consistency dialog box. • Changes the Generation status of facets referencing inconsistent constituents to Inconsistent. NOTE: The check consistency button is not available when you refine a non-FBD section.
3	In the Check Consistency dialog box, select inconsistent sections and/or facets that you want to unlink.
4	Click Unlink to unlink the selected facets. NOTE: Click Cancel to close the Check Consistency dialog box without unlinking facets. The status of facets that the software has detected as inconsistent remains Inconsistent .

Managing Inconsistencies

Even though a logical Control Participant project contains inconsistent facets, you can proceed further in the system engineering life cycle.

Regenerate the FBD section or project containing inconsistent facets to revert to a consistent state, which is indicated by the generation status **Generated** of the facet but certain changes that you made through refinement are discarded, page 421.

For information on managing inconsistencies without affecting changes made through refinement, refer to Generating Inconsistent Facets, page 419.

About Variables for Peer to Peer Communication

Overview

The software facilitates the creation of variables to exchange scattered data or sets of data between two or more Control Participant projects by using peer to peer communication. Such variables are called *network variables*.

You can create network variables in the following ways:

- By assigning facets of instances to different projects.
- By refining both logical Control Participant projects. Refer to Refining the Logical Control Participant project (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
- By using peer to peer communication templates of the EcoStruxure Process Expert Foundation Library.
- By using a mix of the above.

Depending on the way that you use to create them, network variables have the *candidate* status until they are explicitly created for use as network variables by using the **Manage Network Variables** dialog box, page 459.

You can create network variables before proceeding with the service mapping and the creation of communication channels, page 582.

Variables used for peer to peer communication are allocated during build, page 633 based on the network variable mapping, page 594.

NOTE: For more information on the steps to implement peer to peer communication by using the I/O scanner function, refer to the topic describing peer to peer communication by modbus TCP Ethernet implicit messaging, page 779.

Using the Control Participant Peer to Peer Communication Functionality

EcoStruxure Process Expert also automates the implementation of peer to peer communication by using the specialized *READ_REMOTE* and *WRITE_REMOTE* function blocks of the Control Participant instead of or in addition to using the peer to peer communication templates of the EcoStruxure Process Expert Foundation Library.

For details, refer to the topic the describing the management of peer to peer communication, page 462.

Owner/Consumer Project Principle

To help you understand the way peer to peer communication is implemented in the software, this manual uses a particular naming convention to refer to the Control Participant projects that exchange data.

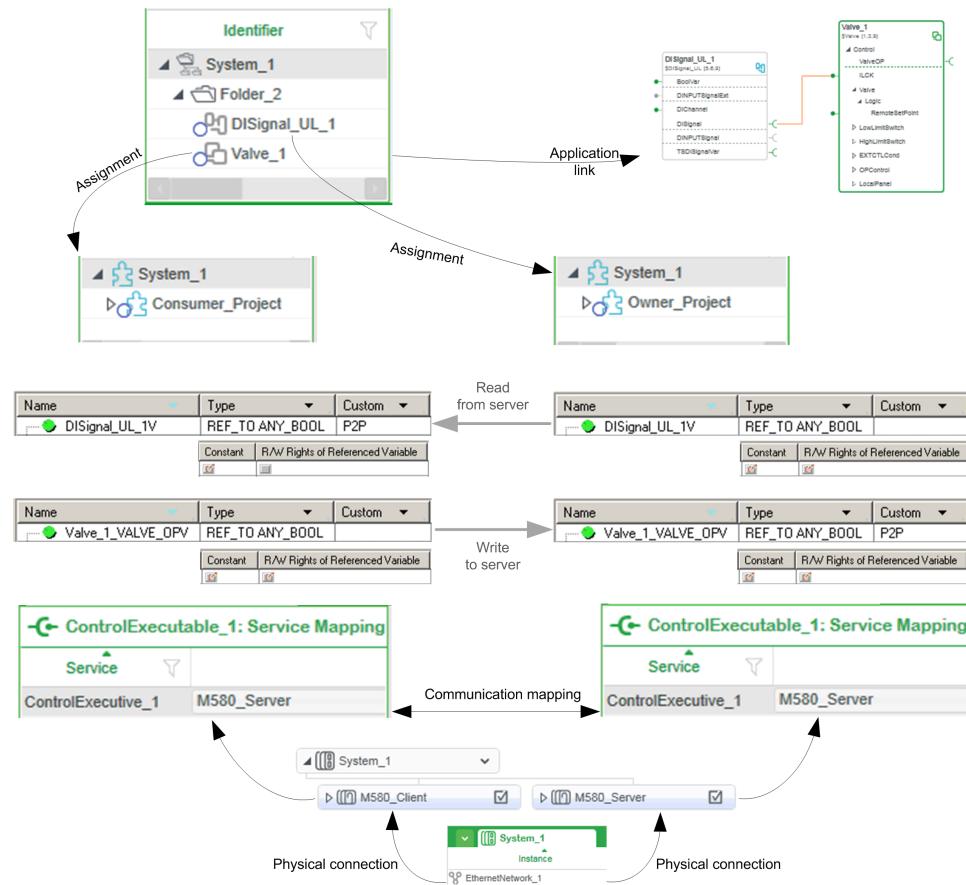
The logical Control Participant project producing the variables to be shared is the *owner* project, and the one receiving the variables is the *consumer* project.

The controller that is mapped to the owner project acts as the server. The one mapped to the consumer project acts as the client.

A network variable allows you to read data from the server by mapping it to a communication channel that exists client-side.

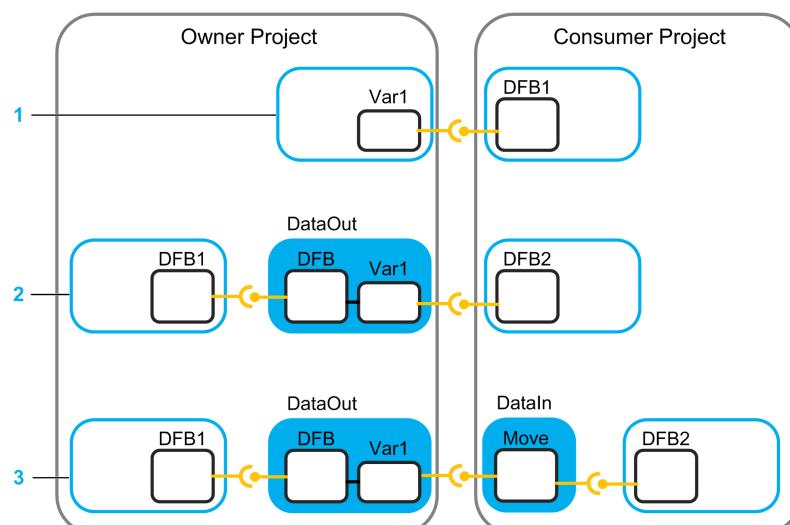
NOTE: You can write data to the server by using a network variable that exists in the owner project and the corresponding variable of same name and type in the consumer project.

The following example illustrates the owner/consumer principle in a system where sample variables *DI Signal_UL_1V* and *Valve_1_VALVE_OPV* are used for peer to peer communication to read from and write to the server respectively. The variables are generated by facets of the *DI Signal_UL_1* and *Valve_1* sample instances respectively. The specific attributes of the variables of reference data type are shown. The two instances are linked and their facets are assigned to the *Owner_Project* and *Consumer_Project* Control Participant projects. The executable of each project is mapped to the corresponding topological entity. The read from server and write to server actions are performed from the consumer (client) project.



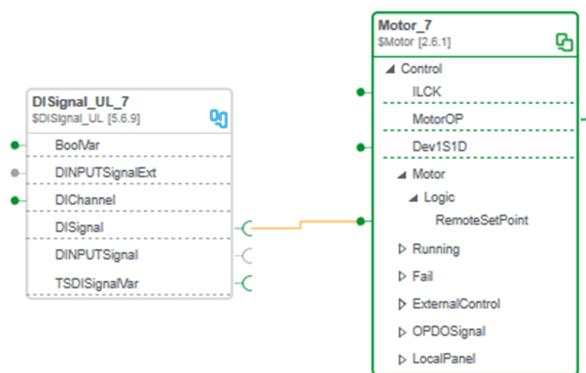
Ways to Create Network Variables to Share Scattered Data One-to-One

The first two scenarios that are shown in the following figure illustrate two common ways to create a network variable to share scattered data between an owner and a consumer project. Other ways to create network variables exist, page 457.

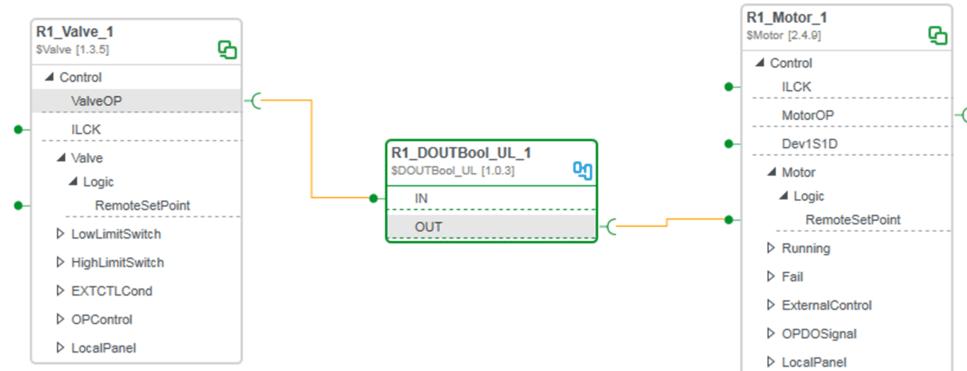


Scenario	Description
1	When the instance in the owner project creates a variable (for example, an instance of a hardware abstraction layer (HAL) template), it can be linked directly to the input pin of a DFB (created, for example, by an instance of an application template) in the consumer project.
2	When the instance in the owner project does not create a variable (for example, an instance of an application template), the output pin needs to be linked first to an instance of a peer to peer communication template (for example, Data Out) in the owner project. The Data Out instance creates a variable, which can be linked directly to the input pin of a DFB (created, for example, by an instance of an application template) in the consumer project.
3	In the first two scenarios, only a network variable candidate is created. You need to create the network variable manually by using the Manage Network Variables command, page 459. This third scenario shows how to create the network variable automatically by using two peer to peer communication templates (Data Out and Data In).

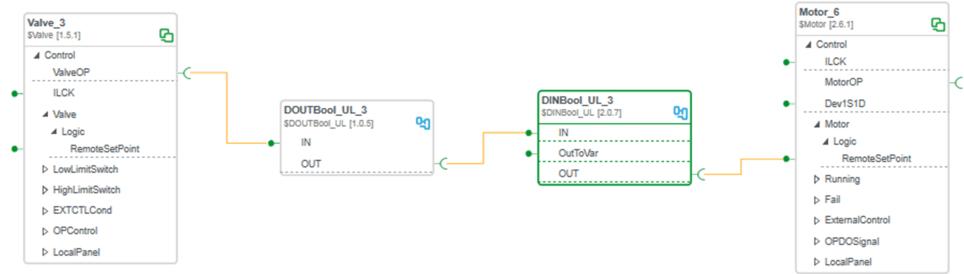
The following example shows how to implement the first scenario by using sample instances and linking them in the **Asset Workspace Editor**. Facets of each instance are then assigned to the owner and consumer projects.



The following example shows how to implement the second scenario by using sample instances and linking them in the **Asset Workspace Editor**. Facets of each instance are then assigned to the owner and consumer projects.

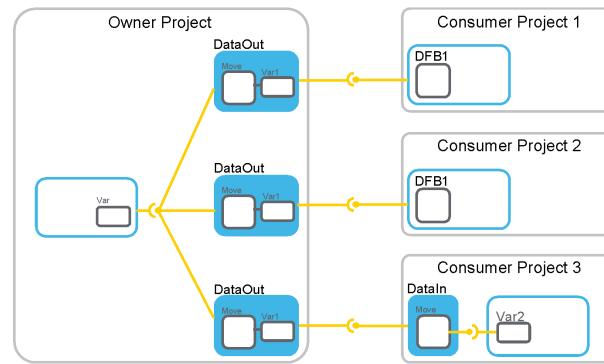


The following example shows how to implement the third scenario by using two peer to peer communication templates to create the network variable automatically. Instances are linked in the **Asset Workspace Editor**.

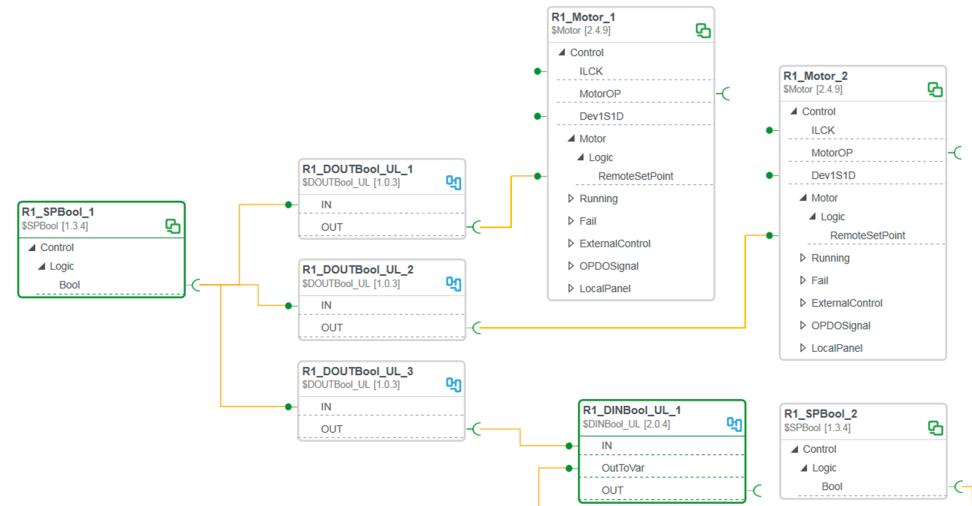


Ways to Create Network Variables to Share Scattered Data One-to-Many

The following figure illustrates the use of Data In and/or Data Out application templates to share one variable exposed by one owner project with three different consumer projects of the same system.



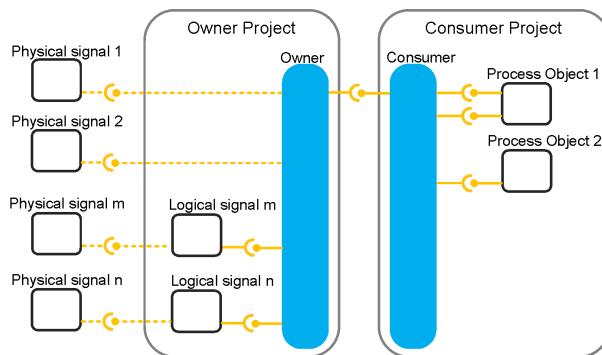
This example illustrates the scenario by using sample instances and linking them in the **Asset Workspace Editor**. Facets of each instance are then assigned to the owner and the three consumer projects.



Ways to Create Network Variables to Share Sets of Data

The following figure illustrates the use of owner and consumer application templates to share physical signals:

- Without signal conditioning.
- With signal conditioning by using a HAL template.



NOTE: In both cases, the physical signals are linked to the owner template and the HAL templates respectively by using mapping interfaces (dotted lines) during the hardware mapping.

For more information on owner/consumer templates and a usage example, refer to the topic describing these templates (see EcoStruxure™ Process Expert , Foundation Application Templates, User Guide).

Properties of Variables for Peer to Peer Communication

Certain attributes of variables that you create through refinement to be mapped to peer to peer communication channels require a specific configuration.

The configuration varies depending on:

- Whether the variable is used to read from the server or write to the server, page 452.
- The platform or the communication module that is used by the topological entity acting as client.

The table describes required variable attribute values in the Control Participant when the topological entity acting as client communicates by using a Quantum CPU module or an NOE communication module (with Quantum or M340 platform).

Usage	Variable attribute	Owner project/Server	Consumer project/Client
Reading data from the server	Designation of the variable in the context of EcoStruxure Process Expert	Variable	Network variable
	Name	Any valid name	Same name
	Type	Any, including the Reference data type (REF_TO). Some restrictions apply, page 459.	Same type
	Custom	Blank	P2P
	Constant⁽¹⁾	True (selected)	True (selected)
	R/W Rights of Referenced Variable⁽¹⁾	True (selected)	False (cleared)
Writing data to the server	Designation of the variable in the context of EcoStruxure Process Expert	Network variable	Variable
	Name	Any valid name	Same name
	Type	Any, including the Reference data type (REF_TO). Some restrictions apply, page 459.	Same type
	Custom	P2P	Blank

Usage	Variable attribute	Owner project/Server	Consumer project/Client
	Constant ⁽¹⁾	True (selected)	True (selected)
	R/W Rights of Referenced Variable ⁽¹⁾	True (selected)	True (selected)
(1) For variables of the Reference data type (REF_TO) only			

The table describes required variable attribute values in the Control Participant when the topological entity acting as client communicates by using an M580 CPU module with or without NOC communication module, or a Quantum controller with NOC communication module.

Usage	Variable attribute	Owner project/Server	Consumer project/Client
Reading data from the server	Designation of the variable in the context of EcoStruxure Process Expert	Variable	Network variable
	Name	Any valid name	Same name
	Type	Reference data type (REF_TO) only	Same type
	Custom	Blank	P2P
	Constant	True (selected)	True (selected)
	R/W Rights of Referenced Variable	True (selected)	False (cleared)
Writing data to the server	Designation of the variable in the context of EcoStruxure Process Expert	Network variable	Variable
	Name	Any valid name	Same name
	Type	Reference data type (REF_TO) only	Same type
	Custom	P2P	Blank
	Constant	True (selected)	True (selected)
	R/W Rights of Referenced Variable	True (selected)	True (selected)

NOTE: For more information on the REF_TO data type, refer to the topic describing the refinement of Control projects (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

Overview of Ways to Create Network Variables

The table provides an overview of the different ways to create network variables in the consumer project to read data from the controller of the owner project, acting as server. It also indicates the status of the variable for each method.

Action in owner project	Action in consumer project	Status of variable
Instantiate a Data Out template corresponding to the data type to be shared (for example, \$DOUTBool_UL to share a boolean variable of the REF_TO ANY_BOOL data type).	-	Once you generate both projects, the Manage Network Variables dialog box detects the variable as a network variable candidate in the consumer project, with the name of the Data Out template instance. Create the network variable, page 459.
	Instantiate a Data In template corresponding to the data type of the Data Out template (for example, \$DINBool_UL).	The generation process creates a network variable in the consumer project with the name of the Data Out template instance. NOTE: The Manage Network Variables dialog box shows the variable in the Existing Network Variables pane.
Instantiate a peer to peer owner template corresponding to the data	Instantiate a peer to peer consumer template matching the peer to peer owner template (for	The generation process creates a network variable in the consumer project with the name of the owner template instance.

Action in owner project	Action in consumer project	Status of variable
type to be shared (for example, \$Bool/16/NO to share up to 16 boolean input signals of the REF_TO data type).	example, \$Bool/16/NC to receive up to 16 boolean input signals).	NOTE: The Manage Network Variables dialog box shows the variable in the Existing Network Variables pane.
Instantiate a template, which creates a variable during the generation process (for example, a variable of type REF_TO ANY_BOOL ⁽¹⁾).	Reference a variable of the same name and data type as the one created by the instance in the owner project without declaring it by using the Refine command.	The Manage Network Variables dialog box detects the variable as a network variable candidate in the consumer project. Create the network variable, page 459.
	Instantiate a template that receives the variable exposed by the instance in the owner project by creating a link between both templates in the Asset Workspace Editor .	
	Create a variable of the same name and data type as the one created by the instance in the owner project, and enter P2P as Custom attribute by using the Refine command.	The Manage Network Variables dialog box shows the variable in the Existing Network Variables pane.
Create a variable ⁽¹⁾ to be shared by using the Refine command.	Reference a variable of the same name and data type without declaring it, using the Refine command.	The Manage Network Variables dialog box detects the variable as a network variable candidate in the consumer project. Create the network variable, page 459.
	Create a variable of the same name and data type, and enter P2P as Custom attribute by using the Refine command.	The Manage Network Variables dialog box shows the variable in the Existing Network Variables pane.

(1) Verify that the data type of the variable that is created meets the requirements to be used for peer to peer communication, page 456.

NOTE:

- The table does not detail each step required to implement peer to peer communication. For more information, refer to Peer to Peer Communication Through Modbus TCP Implicit Messaging, page 787.
- To be detected and/or created as network variables by the **Manage Network Variables** dialog box, variables need to satisfy the network variable management rules, page 459.

Overview of Peer to Peer Templates

The following templates are available from the Global Templates library to create network variables:

- Data IN/Data OUT: Application facet templates that create a network variable in the consumer project. The library contains templates for sharing various variable sizes and data types. Use these templates to share scattered data.

For a detailed description of the templates and how to use them, refer to Data In/Data Out Templates (see EcoStruxure™ Process Expert , Foundation Application Templates, User Guide).

NOTE: To find these templates by using the template browser of the **Application Explorer**, select the **Facets** filter, page 160, and enter either **din** or **dout** in the search field.

- Owner/Consumer: Application composite templates that create a network variable in the consumer project. The library contains templates for sharing up to 16 digital and 8 analog inputs/outputs. Use these templates to share sets of data. Typically, such templates allow you to manage I/O objects exchanged between a Modicon BMX PRA module and a master controller.

For a detailed description of the templates and how to use them, refer to Owner/Consumer Templates (see EcoStruxure™ Process Expert , Foundation Application Templates, User Guide).

NOTE: Network variables created by using peer to peer templates have the **Custom** attribute set to **P2P** by default in the variable properties window of the Control Participant.

Managing Network Variables

Overview

This topic describes how to use the **Manage Network Variables** dialog box to manage network variables, which exist or have the candidate status.

Network variable management includes the creation, modification, and deletion of network variables.

General Network Variable Management Rules

The software applies either of the following rules to identify candidate network variables when you open the **Manage Network Variables** dialog box:

- The variable needs to be referenced from, at least, 1 FBD section but not be declared.
- The variable is created by using a peer to peer template, page 453.

In addition, the following are not considered network variable candidates:

- Variables referencing an element of a data structure (for example, *DigitalInput_DINPUT_ST.STW* where *STW* is a word of the *DINPUT_ST* structure).
- Function block instances (for example, *DigitalInput_DINPUT.PV* where *PV* is a formal parameter of the *DINPUT* DFB).
- Types ANY_X where X represents the type (except for the ANY_BOOL data type).
- Inconsistent types (for example, *a.b* where *a* and *b* are variables and one of them is undefined).
- 32-bit data types.

NOTE: In the **Manage Network Variables** dialog box, it is not taken into account whether the data type of the variable is compatible with the communication module client-side, page 456.

Network Variable Properties

Network variables that the software creates by using the **Manage Network Variables** dialog box have the following properties:

- They are of the same type as shown in the **Type** field of the **Network Variables Candidates** section.
- They have *P2P* as **Custom** attribute in the variable properties window of the Control Participant.

NOTE: The software does not set the **R/W Rights of Referenced Variable** attribute for REF_TO type network variables that it creates. If it needs to be enabled, you need to refine the consumer project and set the attribute manually (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) in the variable properties window of the Control Participant.

Opening the Manage Network Variables Dialog Box

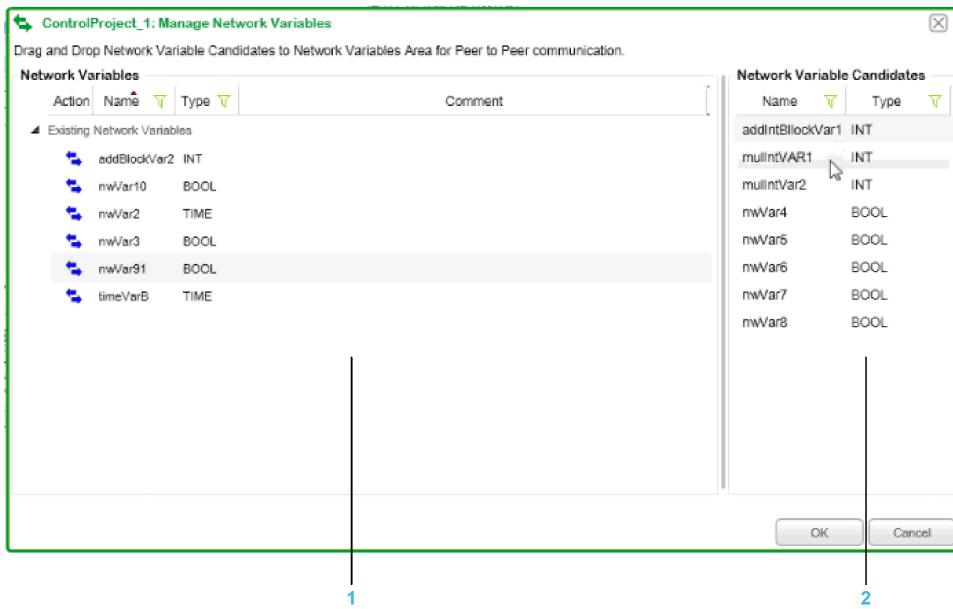
You can open the **Manage Network Variables** dialog box either way:

- From the context menu of the Control Participant project in the **Project Explorer** by selecting **Manage Network Variables**.
- From within the Control Participant by clicking the  button (see EcoStruxure™ Process Expert, Control Participant Services, User Guide). When you refine at the section level, the button is available only for FBD sections.

Both commands open the **Manage Network Variables** dialog box. The difference lies in the way you save changes.

Manage Network Variables Dialog Box

The following figure shows an example of the **Manage Network Variables** dialog box in which several sample variables appear.



Item	Description
1	Network Variables section showing existing network variables.
2	Network Variable Candidates section showing the candidate variables that the software has detected in the selected project. These variables satisfy the applicable rules and you can them use as network variables. Certain restrictions apply depending on the communication module used by the topological entity acting as client.

Fields of the **Network Variables** section.

Header	Description
Action	<ul style="list-style-type: none"> Left/right arrow icon: Existing network variable. Minus icon: The variable will be removed from the project. Plus icon: The variable will be added to the project. Pencil icon: The variable will be modified in the project. Lock icon: The variable is already mapped to a peer to peer communication channel in the communication mapping, page 582.
Name	Identifier of the variable. The field is editable, page 462.
Type	Data type of the variable
Comment	You can enter a comment for the variable using free form text.

Fields of the **Network Variable Candidates** section.

Header	Description
Name	Identifier of the variable. When you use HAL or peer to peer templates, the name of the variable is the identifier of the template instance.
Type	Data type of the variable.

NOTE: You can select multiple variables in either sections by selecting the first variable, pressing and holding **Ctrl**, and selecting the other variables by using the pointer.

Saving Changes in the Manage Network Variables Dialog Box

Depending from where you have opened the **Manage Network Variables** dialog box, the procedure to apply and save your changes varies.

If you have opened the **Manage Network Variables** dialog box from the context menu of the Control project, proceed as follows to apply and save your changes.

Step	Action
1	In the Manage Network Variables dialog box, click OK . Result: A confirmation dialog box opens.
2	Click Yes to save your changes. NOTE: <ul style="list-style-type: none"> Click No to close the Manage Network Variables dialog box without saving changes. Click Cancel to leave the Manage Network Variables dialog box open without saving changes.

If you have opened the **Manage Network Variables** dialog box from within the Control Participant during refinement, proceed as follows to apply and save your changes.

Step	Action
1	In the Manage Network Variables dialog box, click OK to apply the changes to the logical Control project. Result: The Manage Network Variables dialog box closes. NOTE: If you open the Manage Network Variables dialog box again before closing the Control Participant, your changes are preserved but are not saved yet.
2	Click the save button in the toolbar of the Control Participant window to save your changes. NOTE: Closing the Control Participant without saving does not apply your changes.

Creating Network Variables

To create a network variable by using the **Manage Network Variables** command from the context menu of the Control project, proceed as follows.

Step	Action
1	In the Projects Explorer , right-click the logical Control project in which you want the software to identify network variable candidates and select Manage Network Variables . Result: The Manage Network Variables dialog box opens. If the software did not detect any candidate or network variables, the corresponding dialog box fields are empty.
2	In the Network Variable Candidates section, select the candidate variables that you want to create as network variables and drag them to the Network Variables section. Result: The network variables are shown in the New Network Variables section. NOTE: Right-click a network variable to open a context menu, which allows you to make changes, page 462.
3	Save your changes, page 461. Result: The software closes the dialog box, creates the network variables, and displays a summary of the actions in the notification panel. NOTE: You can see the variable by refining the Control Participant project and double-clicking Elementary Variables in the Variables & FB Instances section of the Project Browser .

Modifying Network Variables

NOTE: Depending from where you have opened the **Manage Network Variables** dialog box, there are different ways to save your changes, page 461.

You can perform the following modifications on network variables displayed in the **Network Variables** section:

Action	Description
Adding a comment	Double-click the Comment field of the variable and enter a description with free form text.
Changing the name of the variable	Double-click the Name of the variable and enter a new name with free form text. NOTE: You can only change the name of existing network variables that are not yet mapped to a communication channel.
Undoing changes	Right-click a variable and select Undo to discard the changes that you have made. You can also press Ctrl+Z . NOTE: You cannot undo changes that you have applied by clicking OK if you have opened the Manage Network Variables dialog box from within the Control Participant.
Deleting a network variable	Right-click a variable and select Remove to: <ul style="list-style-type: none"> Display the variable again in the Network Variable Candidates section if it has not been created yet. Mark the network variable for deletion in the Control Participant project. If the variable is mapped to a communication channel, the software removes the mapping, page 600. You can also select a variable and press Del .

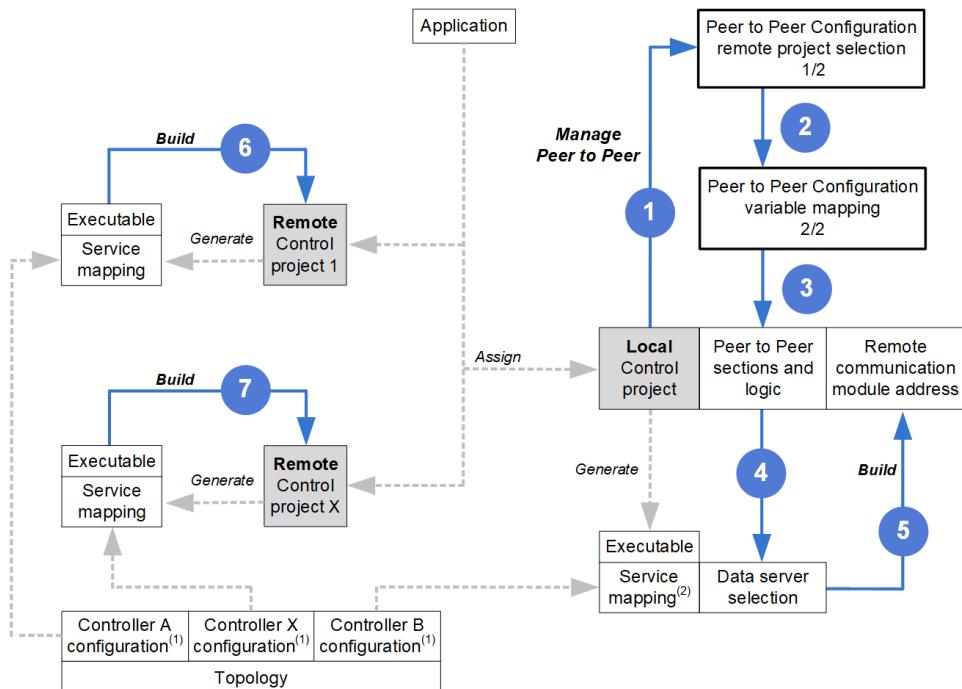
Managing Peer to Peer Communication

Overview

You can implement peer to peer communication by using the specialized *READ_REMOTE* and *WRITE_REMOTE* function blocks of the Control Participant to exchange data between two or more Control projects of the same system. These function blocks as well as the necessary variables and logic are created by EcoStruxure Process Expert in the local Control project.

For more information on the *READ_REMOTE* and *WRITE_REMOTE* function blocks including CPU firmware requirements, refer to the help of the Communication Library of the Control Participant.

The following diagram shows the workflow to configure peer to peer communication between a local Control project (client) and several remote Control projects (servers) in which variables are read and/or written. Refinement of either Control project may be required to perform additional steps. Refer to the next topics for details.



-----	System engineering workflow. Completing these steps is a prerequisite. You can start with the local or the remote Control project.
—	Peer to peer communication configuration steps, page 470.
(1)	Using the Controller simulator, page 516 as topological engine is supported.
(2)	Creating the executable in the local Control project and performing the service mapping can also be done after generating the peer to peer communication mapping (step 3).

NOTE:

You can use this method instead of or in addition to using the peer to peer communication templates, page 452 of the Foundation library.

Peer to Peer Communication Configuration – Remote Project Selection 1/2

Overview

The **Manage Peer to Peer** context menu command of a Control project (**Project Explorer**) opens the **Peer to Peer Communication Configuration** window, which lets you start the peer to peer communication configuration process. The prerequisites, page 470 must be fulfilled to enable the command.

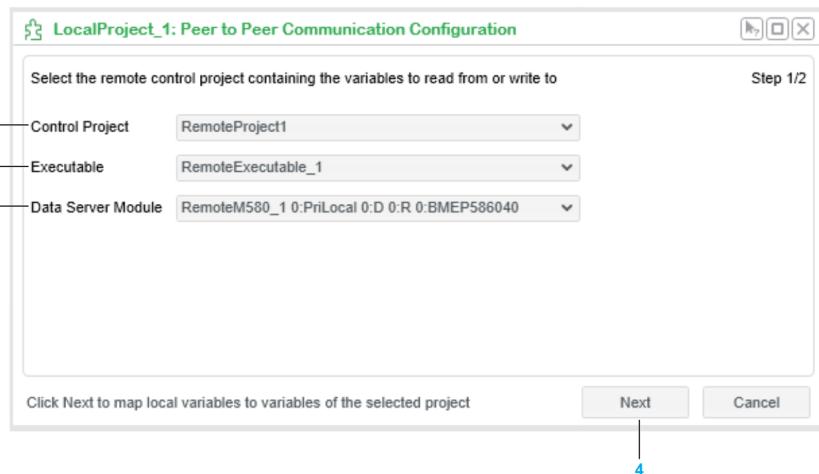
The Control project from which you open this window is considered the *local* project.

Only one user at a time can open this window for the same Control project and while the window is open, the local and the selected remote Control projects are locked.

NOTE: The configuration windows described in this topic do not identify variables that have already been created as [network variables](#), page 452 by using the various methods available. They also do not identify [network variables](#) that have been mapped by using the [Variable Mapping Editor](#), page 594.

Window Description

The following figure shows an example of the window that opens when you select the **Manage Peer to Peer** command for a Control project.



Item	Description
1	<p>Lists the other Control projects of the system, which have at least one executable with existing service mapping.</p> <p>These are the projects (remote) from which you can select variables to read from or write to.</p>
2	Executables of the selected Control project with existing service mapping.
3	<p>Communication modules that exist in the configuration of the controller that is mapped to the selected executable.</p> <p>NOTE: You can also select an executable that is mapped to a station node/Control Expert service instance (Controller simulator). In such case, <i>N/C_X</i> or a CPU module part number appears as data server.</p> <p>NOTE: For peer to peer communication to work, controllers must satisfy the prerequisites, page 470.</p>
4	Opens the variable mapping window (step 2/2).

Configuring Peer to Peer Communication With Controller Simulator Instances

When you select an executable that is mapped to a Control Expert service, page 516 of a station node, the following information appears in the **Data Server Module** text box depending on the port that is used by the Control Expert service:

- Default port 502: *N/C_X* of the station node.
- Other ports: Part number of the CPU module in the configuration of the project file.

The software creates and/or updates the port simulation file IPSimPortAssign.xml in the **Temp** folder at the path C:\Users\<username>\AppData\Local.

Ensure that the same path is configured in the **IP Address <-> Port Simulation File** text box of the **Simulator Panel Options** of the Controller simulator.

The file is created the first time you generate peer to peer communication mappings.

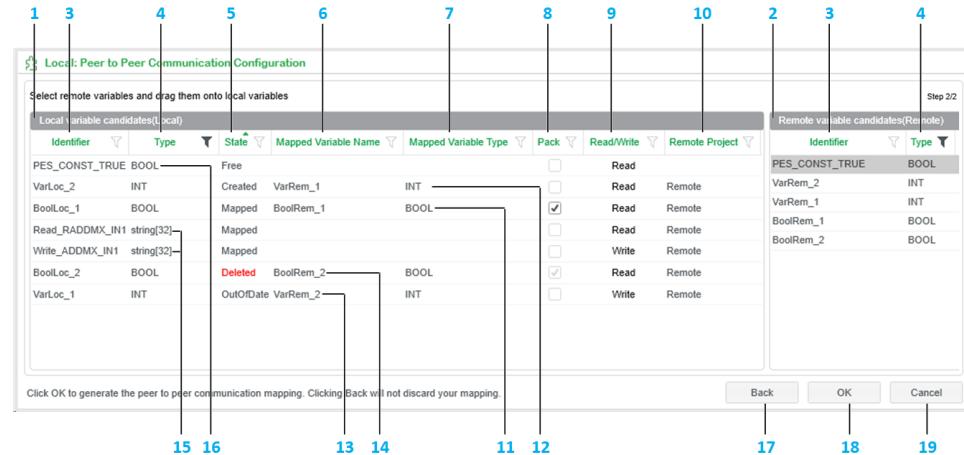
For details on the content and purpose of this file, start the Controller simulator, page 654 and refer to the topic *IP Address and Communication Port Simulation* in the help of the Controller simulator.

NOTE: With Windows Server operating systems, the port simulation file is created in a subfolder of the **Temp** folder. For example, subfolder 2.

Peer to Peer Communication Configuration – Variable Mapping 2/2

Window Description

The following figure shows an example of step 2 of the **Peer to Peer Communication Configuration** window that opens when you click **Next** in step 1. In this example, mappings have been generated a first time then, the window was reopened by using the **Manage Peer to Peer** command and changes were made.



Item	Description
1	<p>Pane showing the variables of the local and the selected remote Control project. It also shows the peer to peer communication mappings that have been generated previously for this pair of Control projects.</p> <p>NOTE:</p> <p>When you reopen the window after generating mappings, the following data is not shown:</p> <ul style="list-style-type: none"> Variables linked to parameters of the <i>READ_REMOTE/WRITE_REMOTE</i> function blocks and that are used for the management of peer to peer communication. Variables of the local Control project that have already been mapped to variables of another remote Control project for peer to peer communication. <p>To view the variables and/or existing mappings, refine the local Control project and open the corresponding sections, page 467.</p>
2	Pane showing the variables of the remote Control project selected in step 1, which you can map to variables of the local project.
3	Name of the variable.
4	<p>Data type of the variable.</p> <p>You can map variables that are of different type but the analysis of the Control project, page 263 during generation of the peer to peer communication mapping or during build of the executable of the local Control project may detect errors.</p> <p>NOTE: By default, a filter is applied to exclude some data types, such as structures. You can clear and/or modify it.</p>
5	<p>State of the local variable with regard to peer to peer communication.</p> <p>Possible values:</p> <ul style="list-style-type: none"> Free: The variable is not yet used for peer to peer communication with any remote project. Created: Variable is already used for peer to peer communication. Mapped: The mapping with the remote variable has been generated in the local Control project. OutOfDate: Applies when a mapping with a remote variable was already generated (state = Mapped) and the following action is performed: <ul style="list-style-type: none"> A mapping with another remote variable, page 472 was performed on the same local variable. The Pack property was changed. <p>For variables assigned to the ADDMX/ADDM function blocks, page 467, this state can also indicate the following in case mappings were already generated:</p> <ul style="list-style-type: none"> The executable of the remote control project has been deleted, page 474.

Item	Description
	<ul style="list-style-type: none"> ◦ The local Control project, which contains peer to peer communication mapping data related to an existing remote Control project was imported. <p>Once you generate changes, the state changes to Mapped again.</p> <ul style="list-style-type: none"> • Deleted: The mapping has been unmapped. You cannot reuse the variable until you generate changes, after which the state changes to Free. <p>This state can also identify the following:</p> <ul style="list-style-type: none"> ◦ Refinement was performed in the local and/or remote Control projects, which results in the deletion of an existing mapping. ◦ The instance or the facet generating a mapped variable has been deleted or unassigned. ◦ The mapping (state = Mapped) pertains to a remote Control project , which has been deleted, page 474. In such case, the identifier of the deleted remote Control project is indicated in the Remote Project column and the mapping will be removed once you generate changes.
6	Name of the variable of the remote Control project that is mapped.
7	Type of the variable of the remote Control project that is mapped.
8	<p>Lets you pack variables, page 469 of boolean data type so that you can group into one variable more than a <i>READ_REMOTE</i> or <i>WRITE_REMOTE</i> function block supports.</p> <p>NOTE: Once you change the Pack property for a mapping that is already generated (state = Mapped), you cannot change it anymore unless you generate changes and reopen the window.</p>
9	<p>Direction of the communication.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Read: The value is read from the remote variable and stored in the local variable. • Write: The value stored in the local variable is written to the remote variable. <p>To change the value, right-click a row in the local variables pane and select a direction. You can set a direction when several mapped variables are selected.</p> <p>NOTE: Once the mapping for a variable has been generated, you cannot change the direction, page 473.</p>
10	Identifier of the remote Control project that the mapped variables belongs to.
11	Variables of one or more remote Control projects that have been mapped to local variables and whose value will be read/written (state is Mapped).
12	New mapping that was created after the variable mapping window was reopened (state is Created). This mapping needs to be generated.
13	Existing mapping that was modified after the variable mapping window was reopened by dragging a different remote variable onto it (state is OutOfDate). This mapping needs to be generated for the change to be applied.
14	Existing mapping that was deleted after the variable mapping window was reopened by using the Unmap context menu command (state is Deleted). This mapping needs to be generated for the change to be applied.
15	<p>Variables of the ADDMX/ADDM function blocks that are linked to a <i>READ_REMOTE</i> or <i>WRITE_REMOTE</i> function block. They hold the topological address and IP address of the selected communication module of the remote Control project.</p> <p>The variables are shown and automatically mapped (and cannot be unmapped) after you generate peer to peer communication. Their value is set after you build the executable of the local Control project.</p>
16	<p><i>PES_CONST_TRUE</i> elementary read-only variable (see EcoStruxure™ Process Expert, Control Participant Services User Guide) that exists in any EcoStruxure Process Expert Control project.</p> <p>NOTE: Mapping onto read-only variables of the local project with direction Read or with read-only variables from the remote project with direction Write is allowed but the analysis of the Control project, page 263 during generation of the peer to peer communication mapping or during build of the executable of the local Control project detects errors.</p>
17	<p>Lets you go back to the Control project selection window where you can select another remote Control project, executable, and/or communication module.</p> <p>Thereafter, when you click Next, you can see already generated mappings and the remote variables that correspond to your new selection.</p> <p>NOTE: Clicking Back does not discard any mappings. However, selecting another remote Control project discards all changes and mappings that have not been generated yet. That is, all mappings with State other than <i>Mapped</i>.</p>

Item	Description
18	Closes the window and generates the peer to peer communication mapping. This creates in the local Control project one or more FBD sections, page 467 containing logic that is used for peer to peer communication.
19	If changes have not been generated yet, opens the Cancel Peer to Peer Communication Configuration dialog box, which lets you perform either action: <ul style="list-style-type: none"> • No, closes the configuration window and discard changes that you have not applied yet. • Yes, applies changes before closing the window. • Cancel, reverts to the variable mapping window. Otherwise, closes the window.

Sections and Logic Created for Peer to Peer Communication

Overview

Generating peer to peer communication mapping for the first time or generating changes creates in the local and/or remote Control projects one or more FBD sections.

The sections appear in the **Assignment Editor** but you cannot assign facets to them. To view and edit their logic, refine the local Control project.

This topic describes the sections and their content.

Refer also to the example scenarios, page 476.

Sections for *READ_REMOTE* and *WRITE_REMOTE* Function Blocks

Section characteristics	Description
Name	<p>Default value:</p> <ul style="list-style-type: none"> • <i>Read_<Remote project identifier>_P2P_X</i> • <i>Write_<Remote project identifier>_P2P_X</i> <p>You can edit the name of the FBD sections in the dialog box that opens when you drag a remote variable onto a local one to create the mapping.</p> <p>Where <i>_X</i> is integer that is incremented if more than one section for a given communication direction is required.</p> <p>The name is discarded and the corresponding section not created if no mapping with this read or write direction has been generated.</p> <p>The next time you drag a remote variable onto a local one, if a section for the <i>READ_REMOTE</i> or <i>WRITE_REMOTE</i> block has not been generated yet in the local Control project, the dialog opens again.</p> <p>For information on applicable naming rules, refer to <i>Variable Names and Section Identifiers</i> in the topic describing naming rules (see EcoStruxure™ Process Expert, User Guide)</p> <p>NOTE: Once a section has been created with the name that you have entered, it appears as read-only in the dialog box if it opens again.</p>
Location	In the local Control project only.
Count	<p>At least one FBD section for each direction (read/write) per remote Control project of which you have mapped variables.</p> <p>If one section cannot hold the required number of function blocks, additional sections are created.</p>
Content	<ul style="list-style-type: none"> • In each section, as many <i>READ_REMOTE</i> or <i>WRITE_REMOTE</i> function blocks as needed to support the number of mapped variables. • One <i>ADDMX</i> or <i>ADDM</i> function block for each <i>READ_REMOTE</i> or <i>WRITE_REMOTE</i> function block. • The necessary variables and logic. <p>NOTE: Sections and variables are locked, page 442.</p>

Section characteristics	Description	
Function block configuration	<p>The variables linked to the following parameters of each <i>READ_REMOTE</i> and <i>WRITE_REMOTE</i> function block are initialized with these values:</p> <ul style="list-style-type: none"> • <i>CTRL</i>: <ul style="list-style-type: none"> ◦ <i>INT[0] (TIMEOUT)</i>: 1000 ◦ <i>INT[1]</i>: 0 • <i>ENABLE</i>: 1 	
Variable properties	Packed variables (independently of the data type)	Refer to the description of variable properties in sections for mapped variables that are packed.
	Variables of <i>REF_TO</i> data type	Refer to the description of variable properties in sections for mapped variables of <i>REF_TO</i> data type.
	Intermediate variables	<ul style="list-style-type: none"> • Name: <ul style="list-style-type: none"> ◦ For variables linked to the <i>READ_REMOTE</i> function block: <i>Read_<First letter of the remote Control project>_RVAR_X</i> where <i>_X</i> is an integer incremented for name uniqueness. ◦ For variables linked to the <i>WRITE_REMOTE</i> function block: <i>Write__RVAR_X</i> where <i>_X</i> is an integer incremented for name uniqueness (the remote Control project is not identified in the name). • Data type: <i>String[128]</i> • Value: Name of the variable of the remote Control project that is mapped or, if the mapped variable is packed, name of the variable that is assigned to the input of the function block unpacking the data, page 469 (for example, <i>Pack_L_UPG_P2P_1</i>).
	Peer to peer management variables	Name: Uses the <i>Read_</i> or <i>Write_</i> prefix and an incremental integer suffix <i>_X</i> , which corresponds to the suffix that is used for the section (for example, <i>Read_ENABLE_1</i> or <i>Write_STATUS_1</i>).
	Variables for topological and IP address of remote data server	<p>Variables linked to the function blocks of type <i>ADDMX/ADDM</i>.</p> <ul style="list-style-type: none"> • Name: <ul style="list-style-type: none"> ◦ For variables in a READ section: <i>Read_<First letter of the remote Control project><Function block type>_INX</i> where <i>_X</i> is an integer incremented for name uniqueness. ◦ For variables in a WRITE section: <i>Write_<Function block type>_INX</i> where <i>_X</i> is an integer incremented for name uniqueness (the remote Control project is not identified in the name). • Data type: <i>String[32]</i> • Value: Topological address and IP address of the selected communication module of the remote Control project. The value is set after you build the executable of the local Control project. Default value: Blank.

Sections for Mapped Variables that are Packed

The following logic is generated when the **Pack** property is selected for at least one boolean variable in step 2 of the **Peer to Peer Communication Configuration** window.

Section characteristics	Description	
Name	In the local Control project	<i>Pack_<Remote project identifier>_P2P_X</i> Where _X is integer that is incremented if more than one section is required.
	In the remote Control project	<i>Pack_<Local project identifier>_P2P_X</i> Where _X is integer that is incremented if more than one section is required.
Count	At least one FBD section per Control project of which you have mapped variables. If one section cannot hold the required number of function blocks, additional sections are created.	
Content	For variables that are read	<p><i>PG_X</i> function blocks of the following types are created:</p> <ul style="list-style-type: none"> • In the remote Control project: <i>BIT_TO_WORD</i> to pack variables • In the local Control project: <i>WORD_TO_BIT</i> to unpack variables • The necessary variables to store the packed data. <p>As many function blocks as needed to support the number of packed variables are created and identified by using the _X integer increment.</p> <p>NOTE: Sections and variables are locked, page 442.</p>
	For variables that are written	<p><i>PG_X</i> function blocks of the following types are created:</p> <ul style="list-style-type: none"> • In the local Control project: <i>BIT_TO_WORD</i> to pack variables • In the remote Control project: <i>WORD_TO_BIT</i> to unpack variables • The necessary variables to store the packed data. <p>As many function blocks as needed to support the number of packed variables are created and identified by using the _X integer increment.</p> <p>NOTE: Sections and variables are locked, page 442.</p>
Variable properties	Variables assigned to the output of the function block packing the data	<p>Name: <i>Pack_<First letter of counterpart project identifier>_PG_P2P_X</i> where _X is an integer incremented for name uniqueness.</p> <p>For additional properties, refine the Control project and open the Data Editor.</p>
	Variables assigned to the input of the function block unpacking the data	<p>Name: <i>Pack_<First letter of counterpart project identifier>_UPG_P2P_X</i> where _X is an integer incremented for name uniqueness.</p> <p>Data type: <i>WORD</i></p> <p>For additional properties, refine the Control project and open the Data Editor.</p>

Sections for Mapped Variables of *REF_TO* Data Type

The following logic is generated when at least one variable of *REF_TO* data type is mapped in step 2 of the **Peer to Peer Communication Configuration** window.

Section characteristics	Description	
Name	In the local Control project	<i>Ref_<Remote project identifier>_P2P_X</i> Where <i>_X</i> is an integer that is incremented if more than one section is required.
	In the remote Control project	<i>Ref_<Local project identifier>_P2P_X</i> Where <i>_X</i> is an integer that is incremented if more than one section is required.
Count	At least one FBD section per communication direction and per Control project of which you have mapped a <i>REF_TO</i> variable. That is, if you have mapped one <i>REF_TO</i> variable to be read and one to be written, two sections are created in the local project and two in the remote project. If one section cannot hold the required number of function blocks, additional sections are created.	
Content	For variables that are read	<p>The following function block instances of type <i>MOVE</i> are created:</p> <ul style="list-style-type: none"> In the remote Control project: Named <i>Ref_<First letters of local project identifier>_P2P_X_Reference_Y</i> to change the variable data type to non <i>REF_TO</i>. In the local Control project: Named <i>Ref_<First letters of remote project identifier>_P2P_X_Reference_Y</i> to revert the variable to <i>REF_TO</i> data type. The necessary variables to store the packed data. <p>As many function blocks as needed to support the number of mapped variables are created and identified by using the <i>_Y</i> integer increment.</p> <p>The <i>_X</i> integer increment corresponds to the section number.</p> <p>NOTE: Sections and variables are locked, page 442.</p>
	For variables that are written	<p>The following function block instances of type <i>MOVE</i> are created:</p> <ul style="list-style-type: none"> In the local Control project: Named <i>Ref_<First letters of remote project identifier>_P2P_X_Reference_Y</i> to change the variable data type to non <i>REF_TO</i>. In the remote Control project: Named <i>Ref_<First letters of local project identifier>_P2P_X_Reference_Y</i> to revert the variable to <i>REF_TO</i> data type. The necessary variables to store the packed data. <p>As many function blocks as needed to support the number of mapped variables are created and identified by using the <i>_Y</i> integer increment.</p> <p>The <i>_X</i> integer increment corresponds to the section number.</p> <p>NOTE: Sections and variables are locked, page 442.</p>
Variable properties	<p>Variables of non <i>REF_TO</i> data type assigned to the input or output of the <i>REF_TO</i> function.</p> <p>Data type: Equivalent to the <i>REF_TO</i> data type (for example, <i>INT</i> for a <i>REF_TO_INT</i> variable).</p> <p>Name: <i>Ref_<First letters of counterpart project identifier>_Ref_P2P_X</i> where <i>_X</i> is an integer incremented for name uniqueness.</p> <p>For additional properties, refine the Control project and open the Data Editor.</p>	

Creating Peer to Peer Communication Mapping

This topic describes the steps to create peer to peer communication mappings and generate the corresponding logic in the respective Control projects.

Prerequisites

Verify that the following prerequisites to generate peer to peer communication mappings are fulfilled:

- Controllers are of the following platform and/or firmware version to support peer to peer communication:
 - M580
Version 3.20 or higher required for the local Control project.
 - M340

Version 3.30 or higher required for the local Control project.

NOTE: BMXPRA0100 CPU modules cannot be used for peer to peer communication because they do not support the data dictionary.

- Quantum (supported only in the remote Control project).

Using the Controller simulator, page 516 as topological engine is supported.

- The data dictionary is enabled for all controllers.
- The variables exist in the local and remote Control Participant projects so that they appear in the peer to peer communication configuration window. That is, the facets creating them are generated and/or variables have been created during refinement.
- The service mapping is done for the executable of the remote Control project so that the **Manage Peer to Peer** command is available.
- The communication modules to be used are configured with an IP address that is unique.

NOTE: For more information on using the *READ_REMOTE* and *WRITE_REMOTE* functions, refer to the help of the Communication Library of the Control Participant.

Creating and Generating the Peer to Peer Communication Mapping

Step	Action
1	<p>In the Project Browser of the Project Explorer, right-click the Control project that is the scanner (local) and select Manage Peer to Peer.</p> <p>Result: The first Peer to Peer Communication Configuration window opens (step 1).</p>
2	<p>Select the Control project (remote) that contains the variables that you want to read from and/or write to, an executable, and a communication module.</p> <p>NOTE: If you have already generated peer to peer mappings and reopen the Peer to Peer Communication Configuration window, a different Control project and executable may be selected by default.</p>
3	<p>Click Next.</p> <p>Result: Step 2 of the Peer to Peer Communication Configuration window opens and after a moment, the local and remote variable panes are populated.</p>
4	<p>In the remote variables pane, select one or more variables and drag them onto a row of the local variables pane.</p> <p>Result: A dialog box opens prompting you to enter an identifier for the sections, page 467 for peer to peer communication that will be created in the local Control project.</p>
5	<p>Enter a meaningful name and click OK.</p> <p>Result: The remote variables are mapped, in the order of selection, to contiguous local variables starting from the row onto which you have dragged them. State of the local variables changes to <i>Created</i>. For boolean variables, Pack, page 469 is selected by default.</p> <p>NOTE:</p> <p>To undo a mapping, page 472, right-click the corresponding row in the local variables pane and select Unmap</p> <p>To modify a mapping, page 472, drag a different remote variable onto the mapped local variable.</p>
6	<p>To change the read/write direction, right-click one or more rows and select a new direction.</p> <p>NOTE: Once the mapping for a variable has been generated (State is <i>Mapped</i>), you cannot change the direction, page 473.</p>
7	<p>Once you are done with the mapping configuration, click OK.</p> <p>Result:</p> <ul style="list-style-type: none"> • The configuration window closes. • The necessary sections and logic, page 467 are created in the local Control project.

Step	Action
	<ul style="list-style-type: none"> The build state of both executables is set to Out Of Date (if it was Built). For the executable of the remote Control project because the <i>HMI</i> attribute of mapped variables is set to true. The outcome of the mapping operation is indicated in the notification panel. <p>NOTE: To map variables of another remote Control project, click Back instead and change your selection in the first window. Any changes and mappings (State other than <i>Mapped</i>) are discarded. To retain your current changes and mappings, generate them first, reopen the Peer to Peer Communication Configuration window, and select the other remote Control project.</p>
8	Open the executable of the local Control project in the Service Mapping Editor , page 580 by selecting Manage from its context menu and select the communication module that you want to use for peer to peer communication in the Data Server for Peer to Peer Communication column.
9	Build the executables of the local and the remote Control projects. Result: The address of the selected communication module of the remote controller is set as value in the address variable of the <i>ADDMX/ADDM</i> function blocks of the local Control project.
10	Refine the local Control project and validate the sections that have been created for peer to peer communication.

Modifying and Deleting Peer to Peer Communication Mapping

Overview

This topic describes various scenarios of peer to peer communication mapping modification and their impact on the information that is shown in the **Peer to Peer Communication Configuration** window.

It also describes a limitation, page 475 related to Control project export/import.

Modifying an Existing Peer to Peer Communication Configuration

To modify an existing peer to peer communication configuration, proceed as follows.

Step	Action
1	Right-click the Control project that is the scanner (local) and select Manage Peer to Peer . Result: The Peer to Peer Communication Configuration window opens (step 1).
2	Select the Control project (remote) whose peer to peer mapping you want to modify, the executable, and the communication module that you had initially selected.
3	Click Next . Result: Step 2 of the Peer to Peer Communication Configuration window opens and after a moment, the local and remote variable panes are populated with the existing mappings.
4	Proceed with your changes and click OK . Result: <ul style="list-style-type: none"> The configuration window closes. The logic, page 467 is created/updated in the local Control project. The build state of both executables is set to OutOfDate (if it was Built). The outcome of the mapping operation is indicated in the notification panel.

Deleting or Changing a Variable Mapping

Depending on the state of the mapping, the table describes how to perform various actions in the **Peer to Peer Communication Configuration** window.

Action	Steps	Additional steps when the mapping has already been generated
Deleting the mapping	Right-click the mapping and select Unmap .	State of the unmapped local variable changes to <i>Deleted</i> . To complete the process and/or map the variable again, generate the change. Open the window again to perform a new mapping.
Moving a mapping to another local variable	1. Right-click the mapping and select Unmap . The mapping is undone. 2. Drag the remote variable onto another local variable.	
Mapping another remote variable to an already mapped local variable	Drag the other remote variable onto the mapped local variable. The existing mapping is undone.	State of the local variable changes to <i>OutOfDate</i> . The existing mapping is updated after you generate the change. NOTE: The new mapping cannot be changed anymore until you generate it.

The following table describes a specific scenario where the local variable is not of *REF_TO* data type and one of the following conditions is satisfied:

- The already mapped remote variable is of *REF_TO* data type and the remote variable that you are dragging to the left pane is not of *REF_TO* data type.
- The already mapped remote variable is not of *REF_TO* data type and the remote variable that you are dragging to the left pane is of *REF_TO* data type.

Action	Steps	Additional steps when the mapping has already been generated
Mapping another remote variable to an already mapped local variable.	1. Right-click the mapping and select Unmap . The mapping is undone. 2. Drag the remote variable onto the local variable.	State of the unmapped local variable changes to <i>Deleted</i> . To complete the unmapping process, generate the change. Open the window again to perform the new mapping.

Changing the Communication Direction of Mapped Variables

Once the mapping of a variable has been generated (**State** is *Mapped* or *OutOfDate*), you cannot change the direction anymore by using the context menu in the **Peer to Peer Communication Configuration** window.

You need to unmap, generate, and redo the mapping.

Deleting the Remote Control Project

When you delete a Control project that is used as *remote* project for peer to peer communication mapping, the corresponding peer to peer communication logic, page 467 is not deleted from the *local* Control project.

By refining the local Control project, you can delete the peer to peer communication function blocks and variables but not the sections.

To delete the corresponding peer to peer communication logic and sections from the local Control project after deleting a remote Control project, proceed as follows.

NOTE: Steps 1 and 2 of this procedure are only required to enable the **Manage Peer to Peer** command from the local Control project.

Step	Action
1	Create a new Control project and an executable. NOTE: Alternatively, you can use any existing Control project other than the local one. If you deleted only the executable of the remote Control project, recreate one.
2	Proceed with the service mapping of the executable to any controller or station node of the topology.
3	Right-click the local Control project and select Manage Peer to Peer . Result: The first Peer to Peer Communication Configuration window opens.
4	Click Next . Result: The second Peer to Peer Communication Configuration window opens and the mappings that exist with variables of the deleted remote Control project are shown with State Deleted . The identifier of the deleted remote Control project is indicated in the Remote Project column.
5	Click OK . Result: The Peer to Peer Communication Configuration window closes and peer to peer communication variables that correspond to mappings with State Deleted are removed from the local Control project. NOTE: Peer to peer management variables (for example, <i>Write_ENABLE_1</i>) are not deleted.
6	Double-click the Containers node of the local Control project in the Control Project Browser .
7	In the Containers pane of the Assignment Editor , right-click the sections that are related to peer to peer communication mapping with the deleted remote Control project and select Delete .
8	If you had created a Control project and/or executable in step 1, delete it.

Deleting the Executable of the Remote Control Project

When you delete the executable of a Control project that is used as *remote* project for peer to peer communication mapping and no other executable exists, you cannot open the **Peer to Peer Communication Configuration** window anymore.

In order to do so, you need to create an executable and perform the service mapping.

Thereafter, when you reopen the **Peer to Peer Communication Configuration** window and select the same remote Control project, existing mappings have the same **State** but the variables assigned to the ADDMX/ADDM function blocks, page 465 have the **State OutOfDate**.

Click **OK** to generate the changes.

Deleting the Local Control Project

When you delete a Control project that is used as *local* project for peer to peer communication mapping, the corresponding peer to peer communication logic, page 467 is not deleted from the *remote* Control project. Logic is present if you have packed variables or mapped variables of *REF_TO* data type.

By refining the remote Control project, you can delete the peer to peer communication function blocks and variables but not the sections.

To delete the corresponding peer to peer communication logic and sections from the remote Control project after deleting the local Control project, proceed as follows.

Step	Action
1	Double-click the Containers node of the remote Control project in the Control Project Browser .
2	In the Containers pane of the Assignment Editor , right-click the sections that are related to peer to peer communication mapping with the deleted local Control project and select Delete .

Refining the Local and Remote Control Projects

NOTE: Changes that you make when refining sections, page 467 that are created for peer to peer communication are not detected when you do a consistency check, page 449. Therefore, it is recommended to create, modify, and delete peer to peer communication mappings only by using the **Peer to Peer Communication Configuration** window.

The following describes refinement scenarios and their impact when you reopen the **Peer to Peer Communication Configuration** window from the same local Control project and select the same remote Control project.

Control project	Action	Impact
Local and/or remote	Deleting an intermediate variable or unlinking a variable from a function block created for peer to peer communication.	State of the corresponding mapping is set to <i>Deleted</i> . Generating changes sets it to <i>Free</i> .
Local	Creating a mapping with a variable from the remote Control project in the section containing the <i>READ_REMOTE</i> or <i>WRITE_REMOTE</i> function blocks.	Variables that you create are shown as local variables in the window. However, the mapping itself is not reflected. Nevertheless, if the mapping is configured properly, it is functional. Refinements that you make in sections created for peer to peer communication are not overwritten when you create or modify mappings in the Peer to Peer Communication Configuration window.

Limitation Related to Project Export/Import

exporting Control projects

When peer to peer communication mappings have already been generated between the local and the remote Control project, exporting either one of these projects or both, reopening the **Peer to Peer Communication Configuration** window for the same projects and creating or modifying mappings and then, importing the project(s) that you have exported may create an inconsistency between the mapping information that is shown in the **Peer to Peer Communication Configuration** window and the mappings that are actually created and visible when you refine the local project.

To help avoid inconsistencies, proceed as follows when exporting/importing Control projects for which peer to peer communication mappings exist:

- Export both the local and the remote Control projects.
- Import both together.
- Before importing the Control projects, delete or rename the corresponding local and remote Control projects in the **Control Project Browser**.

Managing Peer to Peer Communication Example

Overview

By using a simple example, this topic illustrates which sections and logic are generated, page 467 in Control projects whose variables are mapped for peer to peer communication by using the **Peer to Peer Communication Configuration** window.

Various scenarios are described by using one local and two remote Control projects.

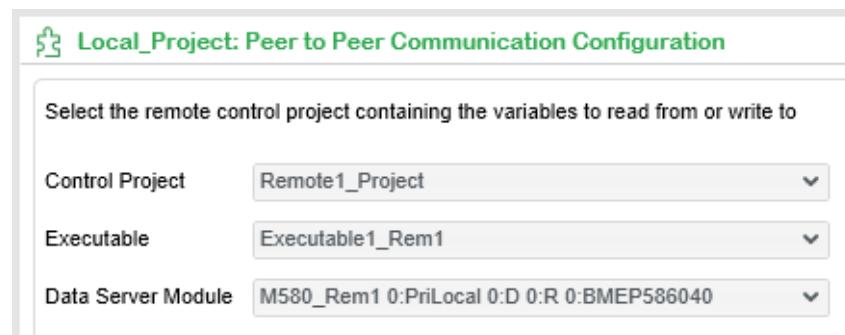
- Mapping of variables of remote Control project 1:
 - Variables of non-*REF_TO* data type, not packed, page 478
 - Packed variables of non-*REF_TO* data type, page 481
 - *REF_TO* variables, not packed, page 483
 - Packed *REF_TO* variables, page 486
- Mapping of variables of remote Control project 2, packed, non-*REF_TO* data type, page 489
- Building the local project, page 492

As a starting point, instances are created in the application, their facets are assigned to the Control projects and generated.

An executable is created for each remote Control project. Each one is mapped to a separate controller.

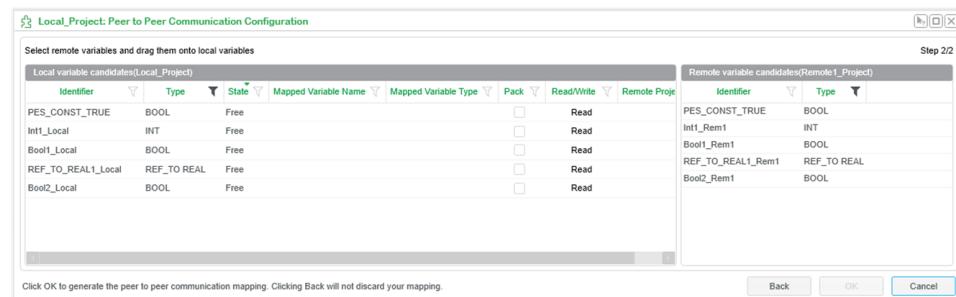
Remote Control Project Selection

The **Peer to Peer Communication Configuration** window is opened from *Local_Project* and *Remote1_Project* is selected as remote project.



Peer to Peer Communication Mapping Window

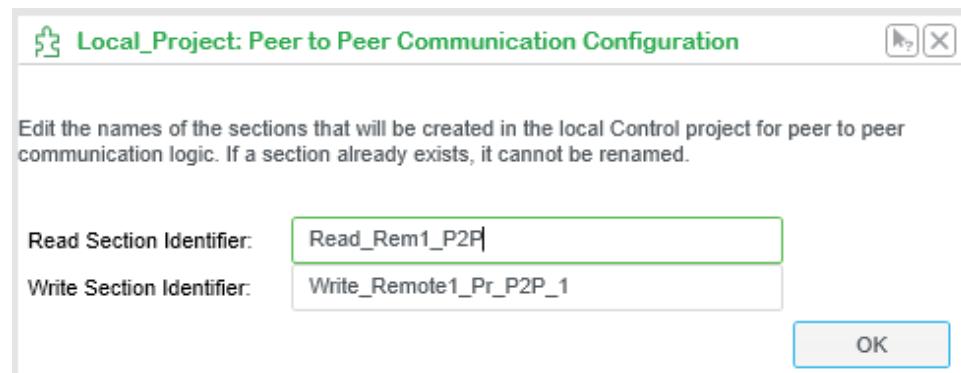
Step 2 of the window shows the variables of *Remote1_Project* (right) that can be mapped to variables of *Local_Project* (left).



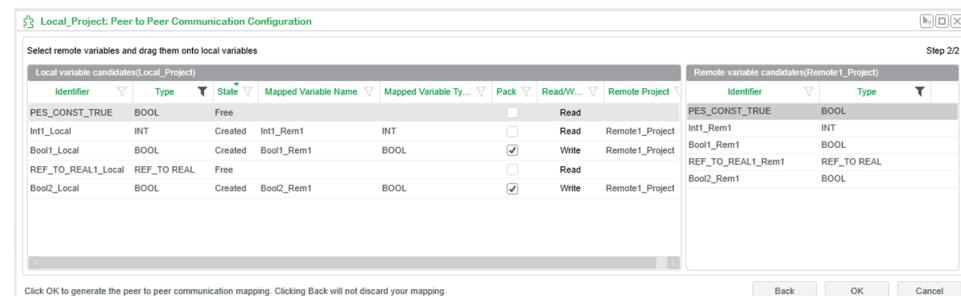
Peer to Peer Communication Mapping Creation

In this example, only the name of the section for the *READ_REMOTE* function block was edited when the first mapping was created by dragging variables of *Remote1_Project* to the left pane.

NOTE: You can edit sections names further by refining the Control projects.



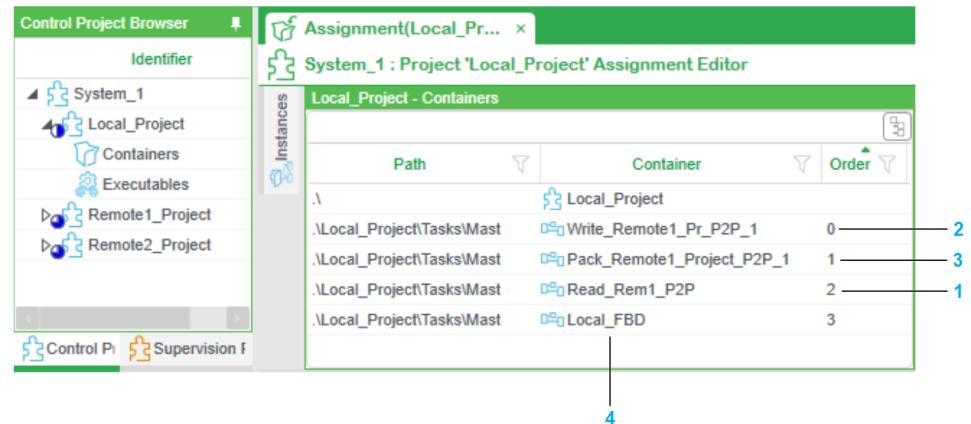
The following figure shows the mappings that are created.



Local variable	Remote variable	Data type	Direction	Packed
<i>Int1_Local</i>	<i>Int1_Rem1</i>	<i>INT</i>	Read	N/A
<i>Bool1_Local</i>	<i>Bool1_Rem1</i>	<i>BOOL</i>	Write	Yes
<i>Bool2_Local</i>	<i>Bool2_Rem1</i>	<i>BOOL</i>	Write	Yes

Peer to Peer Communication Mapping Sections in the Local Project

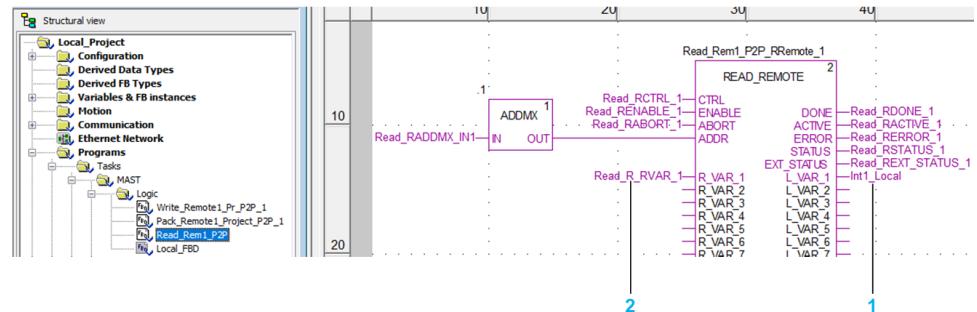
After mappings are generated by clicking **OK** in the communication mapping window, the following FBD sections are shown in *Local_Project*.



Item	Description
1	Peer to peer section containing the <i>READ_REMOTE</i> function block and the necessary variables for the mapping to <i>Int1_Local</i> .
2	Peer to peer section containing the <i>WRITE_REMOTE</i> function block and the necessary variables for the mapping to <i>Bool1_Local</i> and <i>Bool2_Local</i> (packed).
3	Peer to peer section containing the <i>BIT_TO_WORD</i> function block, which packs the values of <i>Bool1_Local</i> and <i>Bool2_Local</i> into one variable (packing).
4	Existing section to which facets of the instances have been assigned.

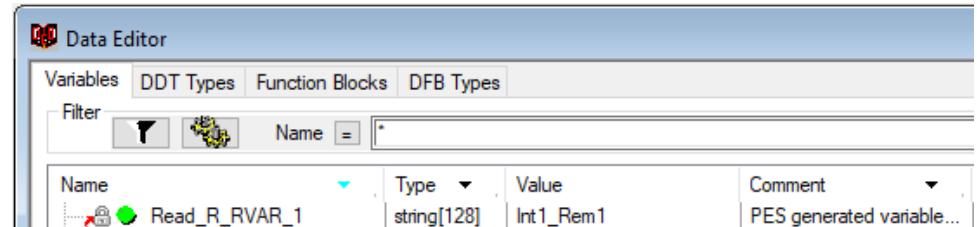
Peer to Peer Communication Mapping Logic in Local Project – Non-Packed Variables

The following refinement window of *Local_Project* shows the logic that is created for the mapping to *Int1_Local*.



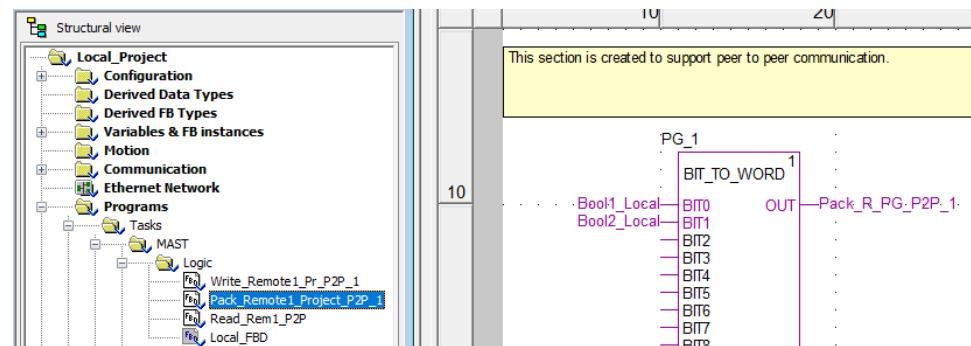
Item	Description
1	Mapped local variable <i>Int1_Local</i> , which receives the value from the intermediate variable.
2	Intermediate variable, which holds the value of the mapped remote variable <i>Int1_Rem1</i> whose value is read.

The following figure shows the values held by the intermediate variable *Read_R_RVAR_1* that is used for peer to peer communication.

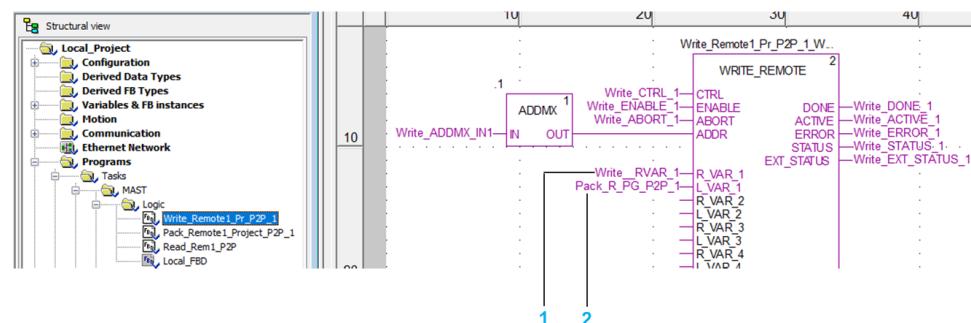


Peer to Peer Communication Mapping Logic in Local Project – Packed Variables

The following refinement window of *Local_Project* shows the logic that is created in a dedicated section to transfer the values of *Bool1_Local* and *Bool2_Local* (**Write** direction) into one variable because their **Pack** check box is selected in the mapping window (packing).

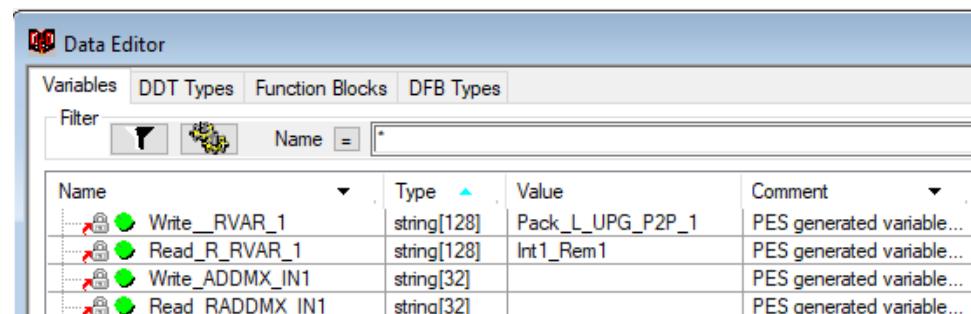


The following refinement window of *Local_Project* shows the logic that is created to transfer the values of packed variables *Bool1_Local* and *Bool2_Local* to one intermediate variable.



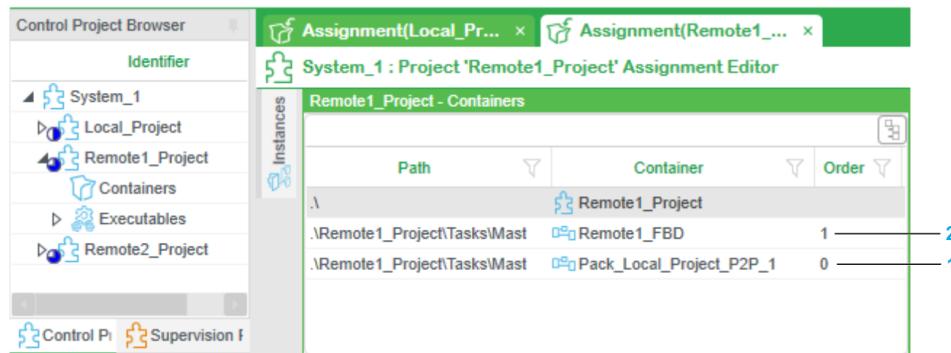
Item	Description
1	Intermediate variable.
2	Variable that holds the values of <i>Bool1_Local</i> and <i>Bool2_Local</i> to be written to the remote variables.

The following figure shows the value held by the intermediate variable *Write_RVAR_1* in *Local_Project*, which is used for peer to peer communication.



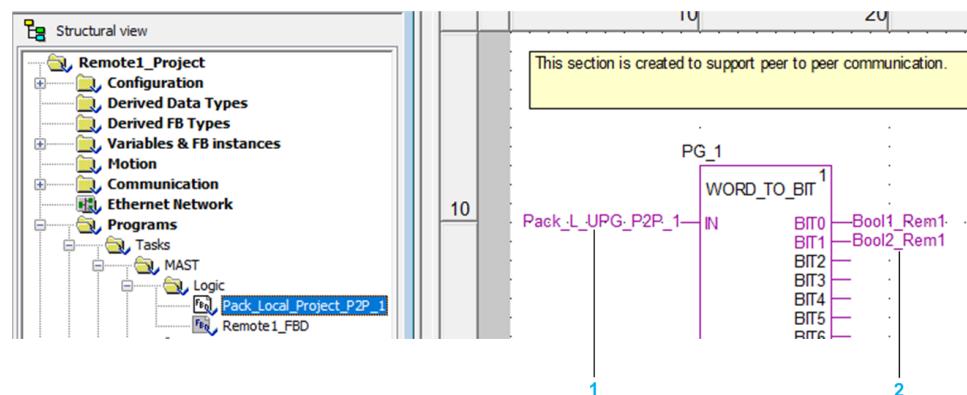
Peer to Peer Communication Mapping Sections in the Remote Project

After generating the peer to peer communication mapping, the following FBD sections are shown in *Remote1_Project*.



Item	Description
1	Peer to peer section containing the <i>WORD_TO_BIT</i> function block, which transfers the values held by one variable to the two remote variables <i>Bool1_Rem1</i> and <i>Bool2_Rem1</i> (unpacking).
2	Existing section to which facets of the instances have been assigned.

The following refinement window of *Remote1_Project* shows the logic that is created in a dedicated section to transfer the value from one variable to the mapped remote variables.



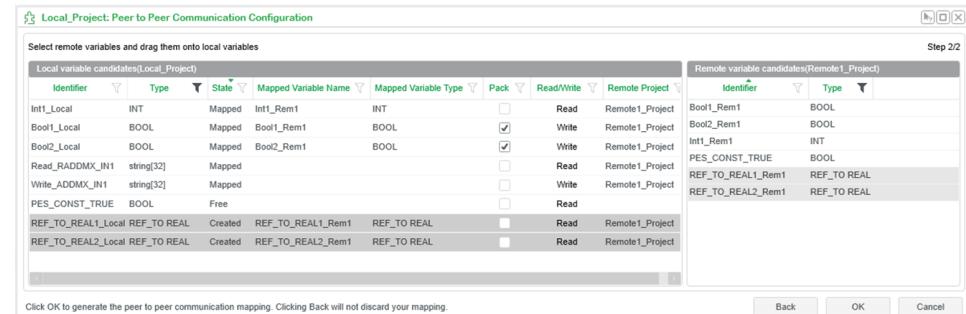
Item	Description
1	Variable holding the values of <i>Bool1_Local</i> and <i>Bool2_Local</i> to be written to the remote variables.
2	Remote variables <i>Bool1_Rem1</i> and <i>Bool2_Rem1</i> .

Peer to Peer Communication Mapping Sections and Logic Local Project – *REF_TO* Variables

Two additional variables, *REF_TO_REAL2_Local* and *REF_TO_REAL2_Rem1*, have been created in the application and their facets assigned and generated in *Local_Project* and *Remote2_Project* respectively.

Then, the **Peer to Peer Communication Configuration** window is reopened from *Local_Project*.

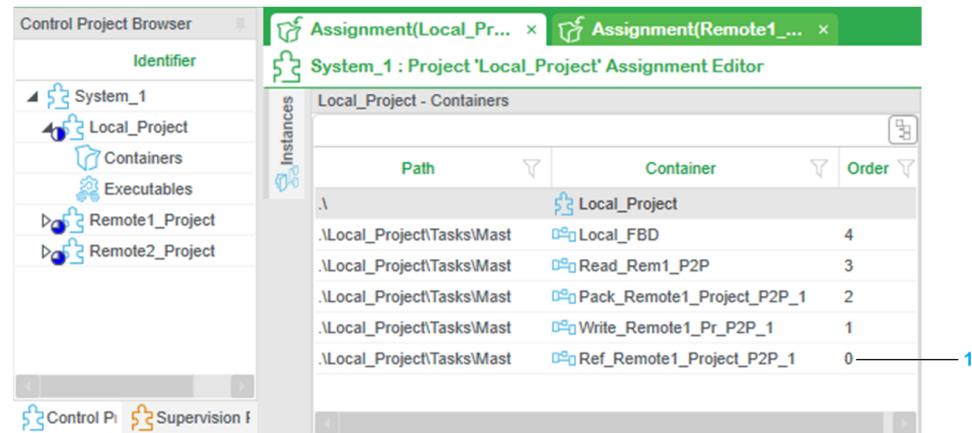
The following figure shows the mappings that have been created.



Local variable	Remote variable	Data type	Direction	Packed
<i>REF_TO_REAL1_Local</i>	<i>REF_TO_REAL1_Rem1</i>	<i>REF_TO_REAL</i>	Read	N/A
<i>REF_TO_REAL2_Local</i>	<i>REF_TO_REAL2_Rem1</i>	<i>REF_TO_REAL</i>	Read	N/A

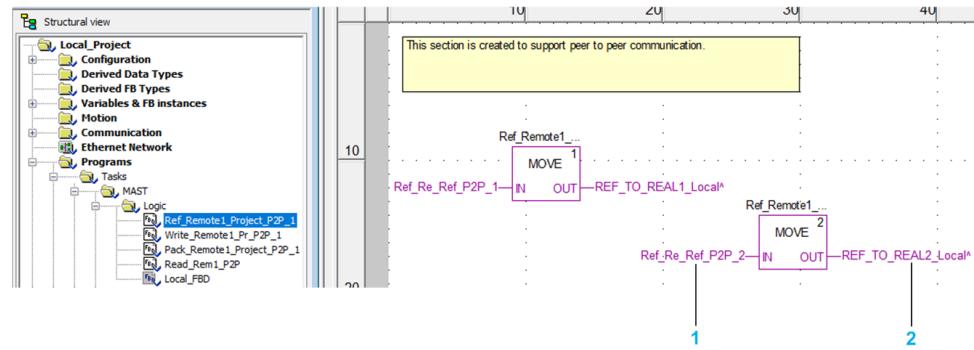
NOTE: In the configuration window, you can see the mappings that have been generated previously and the two variables that hold the topological and IP address of the communication module.

After mappings are generated by clicking **OK** in the communication mapping window, the following FBD sections are shown in *Local_Project*.



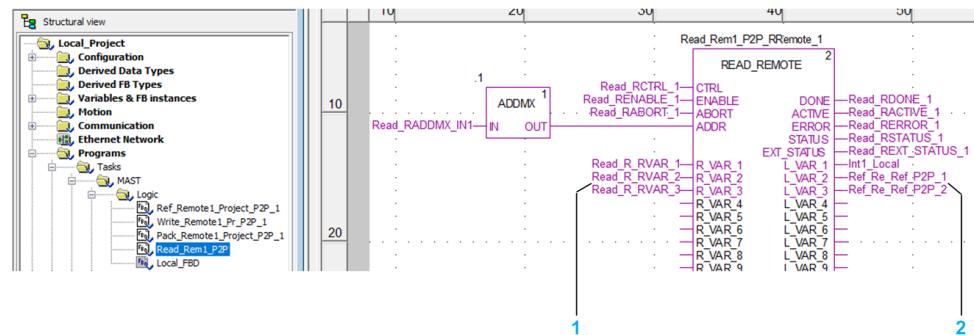
Item	Description
1	<p>New peer to peer section that is created specifically when creating a mapping with local variables of <i>REF_TO</i> data type.</p> <p>NOTE: The section is not created if you map a remote variable of <i>REF_TO</i> data type (for example, <i>REF_TO_REAL</i>) to a local variable of non-<i>REF_TO</i> data type (for example, <i>REAL</i>).</p>

The following refinement window of *Local_Project* shows the logic that is created in a dedicated section to transfer the value from non-*REF_TO* variables to *REF_TO* variables.



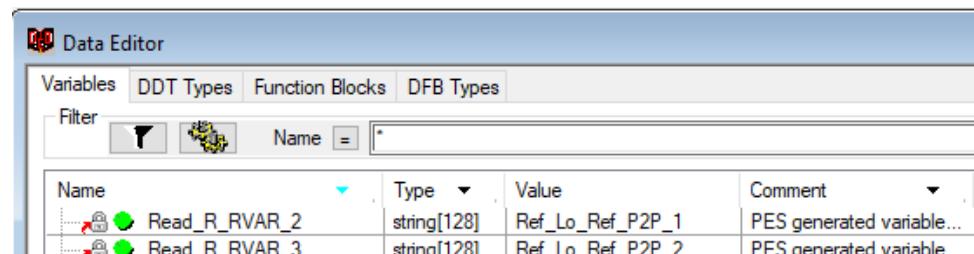
Item	Description
1	Variable that holds the value of the mapped remote variable and which is of compatible data type (<i>REAL</i>).
2	Mapped local variable of <i>REF_TO_REAL</i> data type, which receives the value that is read from the remote variable.

The following refinement window of *Local_Project* shows the logic that is added to the *READ_REMOTE* function block to transfer the values held by the intermediate variables to the non-*REF_TO* variables.



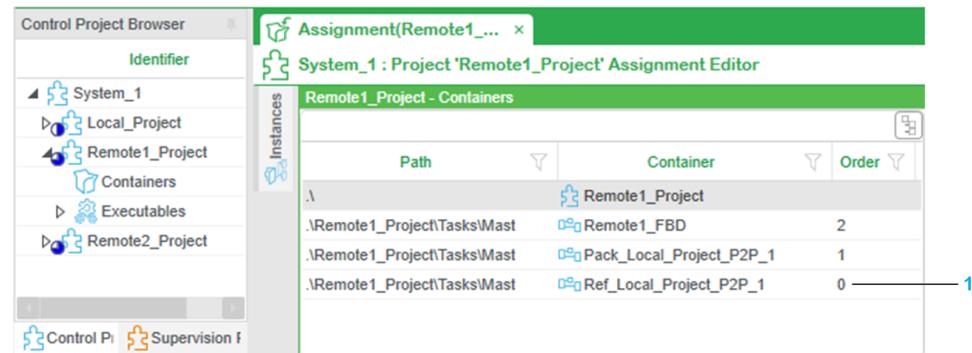
Item	Description
1	Intermediate variables.
2	Variables of non- <i>REF_TO</i> data type holding the value of the remote mapped variables <i>REF_TO_REAL1_Rem1</i> and <i>REF_TO_REAL2_Rem1</i> .

The following figure shows the value held by the intermediate variables *Read_R_RVAR_2* and *Read_R_RVAR_3* in *Local_Project*, which are used for peer to peer communication.



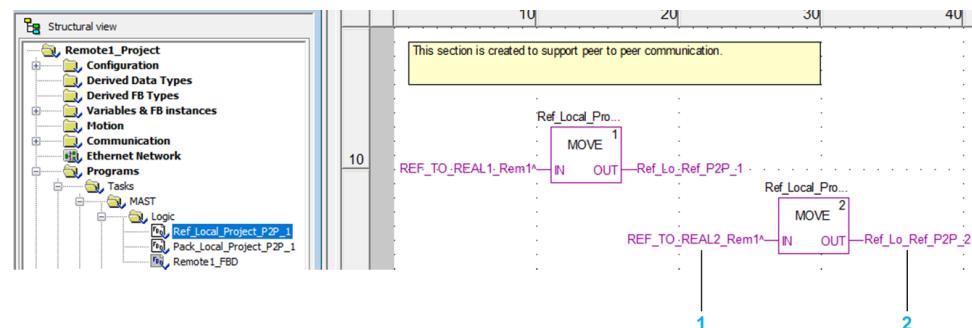
Peer to Peer Communication Mapping Sections and Logic Remote Project – *REF_TO* Variables

After generating the peer to peer communication mapping, the following FBD sections are shown in *Remote1_Project*.



Item	Description
1	<p>New peer to peer section that is created specifically when creating a mapping with remote variables of <i>REF_TO</i> data type.</p> <p>NOTE: The section is not created if you map a remote variable of non-<i>REF_TO</i> data type (for example, <i>REAL</i>) to a local variable of <i>REF_TO</i> data type (for example, <i>REF_TO_REAL</i>).</p>

The following refinement window of *Remote1_Project* shows the logic that is created in a dedicated section to transfer the value from *REF_TO* variables to non-*REF_TO* variables.



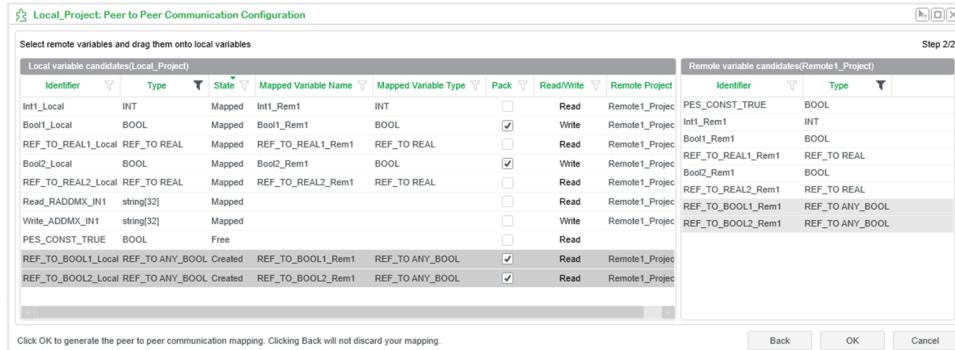
Item	Description
1	Mapped remote variable of <i>REF_TO_REAL</i> data type whose value is read.
2	Variable of compatible non- <i>REF_TO</i> data type (<i>REAL</i>).

Peer to Peer Communication Mapping Sections and Logic – Packed *REF_TO* Variables

New variables of data type *REF_TO_ANY_BOOL* have been created in the application and their facets assigned and generated in *Local_Project* and *Remote1_Project* respectively.

Then, the **Peer to Peer Communication Configuration** window is reopened from *Local_Project*.

The following figure shows the mappings that are created.

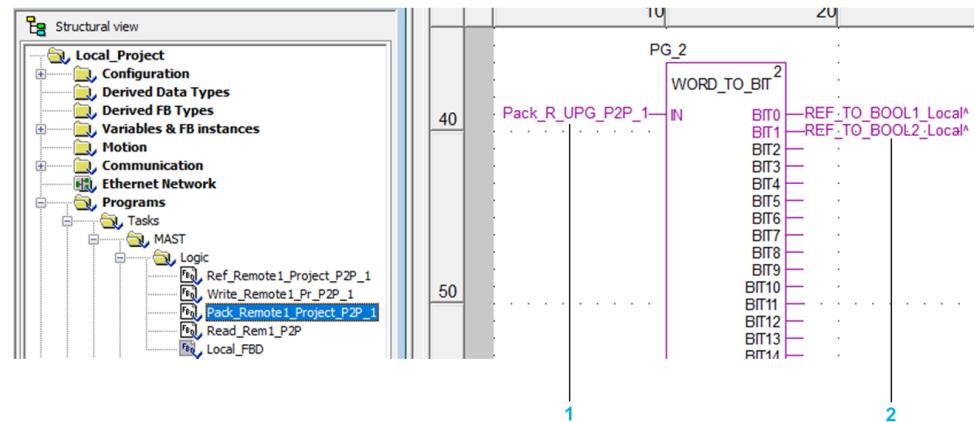


Local variable	Remote variable	Data type	Direction	Packed
<i>REF_TO_BOOL1_Local</i>	<i>REF_TO_BOOL1_Rem1</i>	<i>REF_TO_ANY_BOOL</i>	Read	Yes
<i>REF_TO_BOOL2_Local</i>	<i>REF_TO_BOOL2_Rem1</i>	<i>REF_TO_ANY_BOOL</i>	Read	Yes

After communication mappings are generated by clicking **OK** in the mapping window, no section is added to *Local_Project* nor *Remote1_Project*.

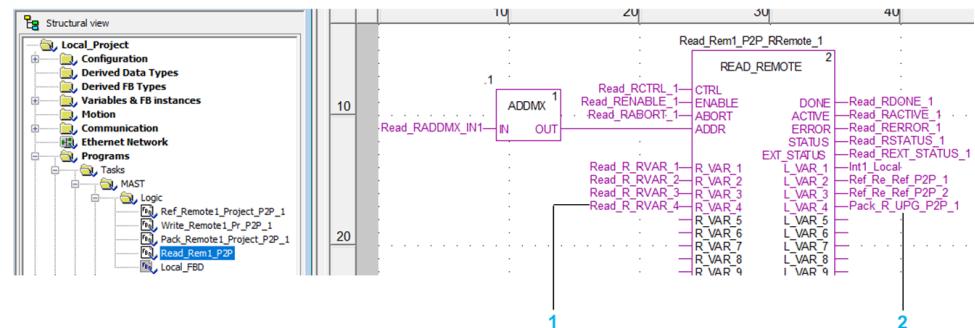
The following refinement window of *Local_Project* shows the logic that is added to the existing *Pack_Remote1_Project_P2P_1* section.

Because the mapped variables are packed, a *WORD_TO_BIT* function block is added to transfer the value from a non-*REF_TO* variable to *REF_TO* variables.



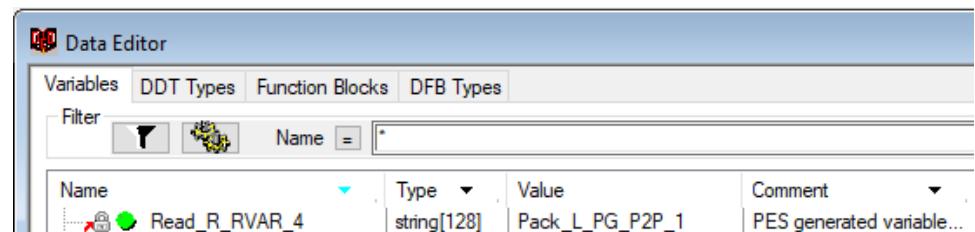
Item	Description
1	Variable of non- <i>REF_TO</i> data type holding the values of the remote mapped variables <i>REF_TO_BOOL1_Rem1</i> and <i>REF_TO_BOOL2_Rem1</i> .
2	Mapped local variables which receive the values that are read from the remote variables.

The following refinement window of *Local_Project* shows the logic that is added to the *READ_REMOTE* function block to transfer the values held by the intermediate variable to one non-*REF_TO* variable.

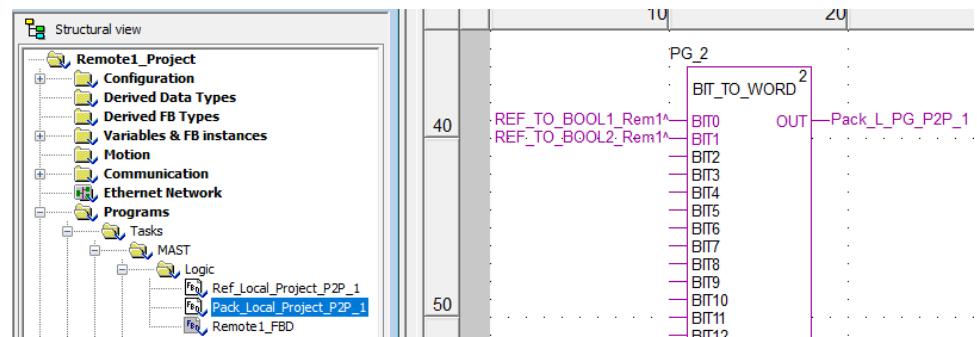


Item	Description
1	Intermediate variable.
	Variable of non- <i>REF_TO</i> data type holding the values of the remote mapped variables <i>REF_TO_BOOL1_Rem1</i> and <i>REF_TO_BOOL2_Rem1</i> .

The following figure shows the value held by the intermediate variable *Read_R_RVAR_4* in *Local_Project*, which is used for peer to peer communication.



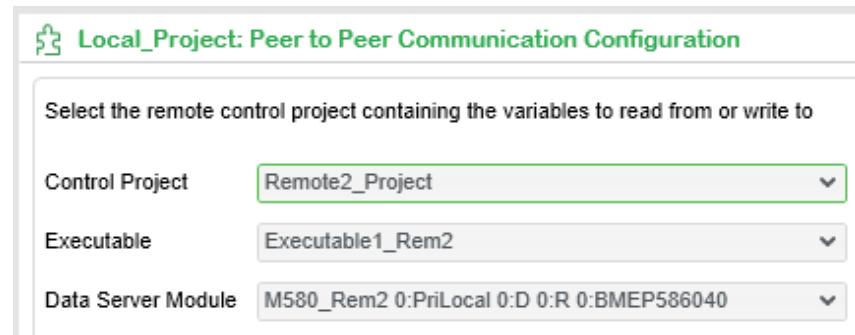
The following refinement window of *Remote1_Project* shows the logic that is added to existing section *Pack_Local_Project_P2P_1* to transfer the value from the mapped remote *REF_TO* variables to the non-*REF_TO* variable.



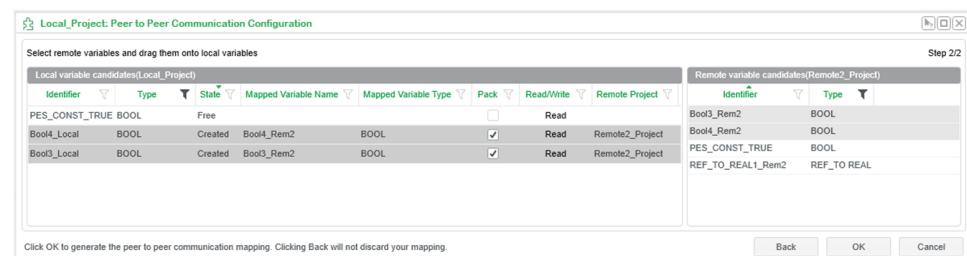
Peer to Peer Communication Mapping With Second Remote Control Project

The **Peer to Peer Communication Configuration** window is reopened from *Local_Project* and *Remote2_Project* is selected.

T



The following figure shows the mappings that are created.

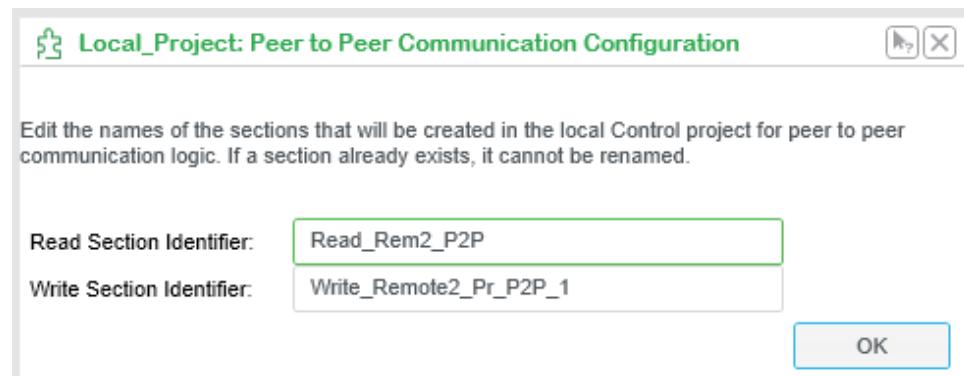


NOTE: In the left pane, you only see the local variables of *Local_Project* that have not been mapped to any remote project yet.

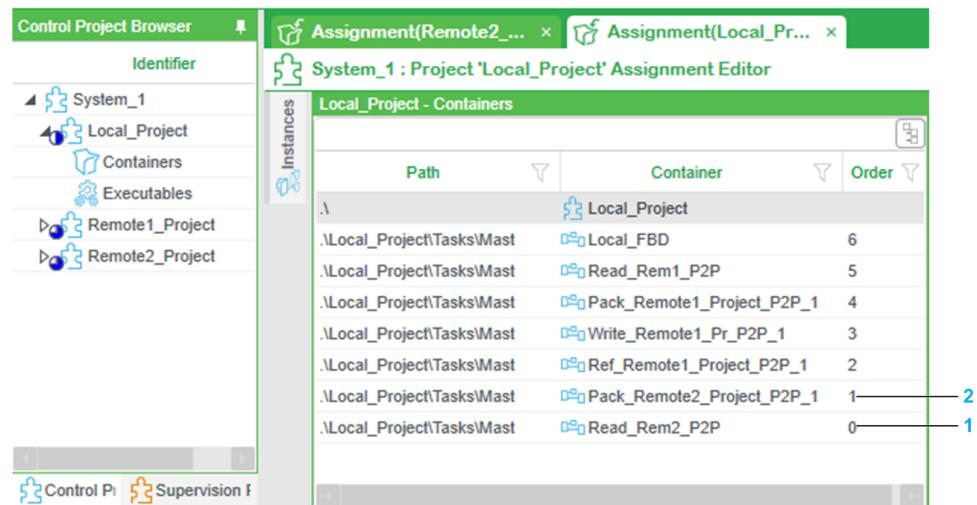
Local variable	Remote variable	Data type	Direction	Packed
BOOL3_Local	BOOL3_Rem2	BOOL	Read	Yes
BOOL4_Local	BOOL4_Rem2	BOOL	Read	Yes

In this example, only the name of the section for the *READ_REMOTE* function block was edited when the first mapping with *Remote2_Project* was created by dragging variables to the left pane. The section for the *WRITE_REMOTE* function block is not created even though its name is shown.

NOTE: You can edit sections names further by refining the Control projects.

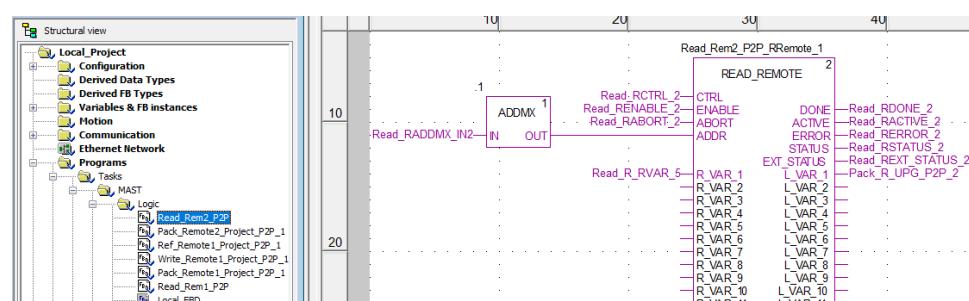


After mappings are generated by clicking **OK** in the communication mapping window, the following FBD sections are shown in *Local_Project*.

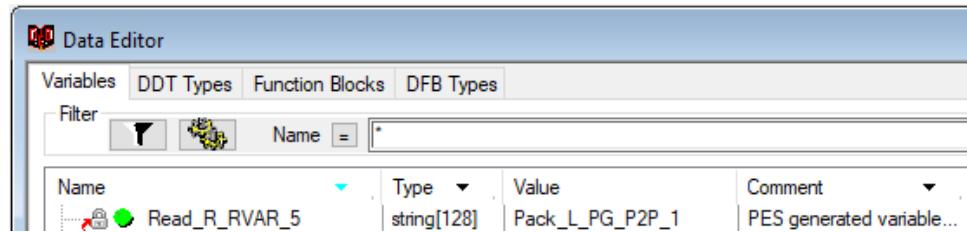


Item	Description
1	New FBD section dedicated to peer to peer communication with variables of <i>Remote2_Project</i> . It contains a <i>READ_REMOTE</i> function block and the necessary variables.
2	New FBD peer to peer section for <i>Remote2_Project</i> containing a <i>WORD_TO_BIT</i> function block, which transfers the values that are held by variable <i>Pack_R_UPG_P2P_2</i> to the two local variables <i>Bool3_Local</i> and <i>Bool4_Local</i> .

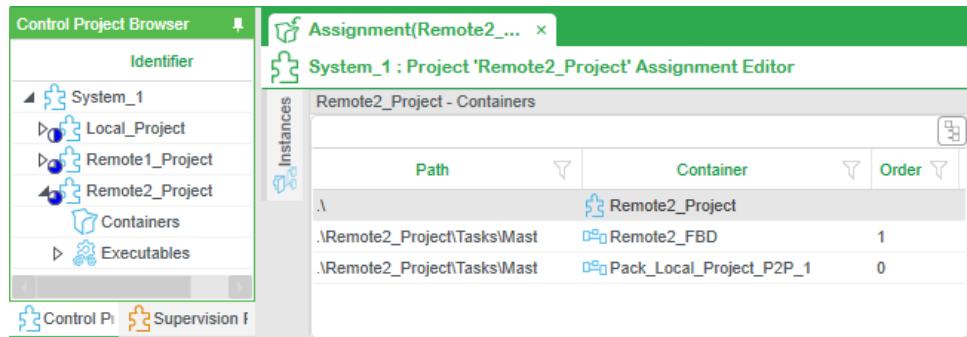
The following refinement window of *Local_Project* shows the logic that is created in the new section to transfer the values held by the intermediate variable to one variable.



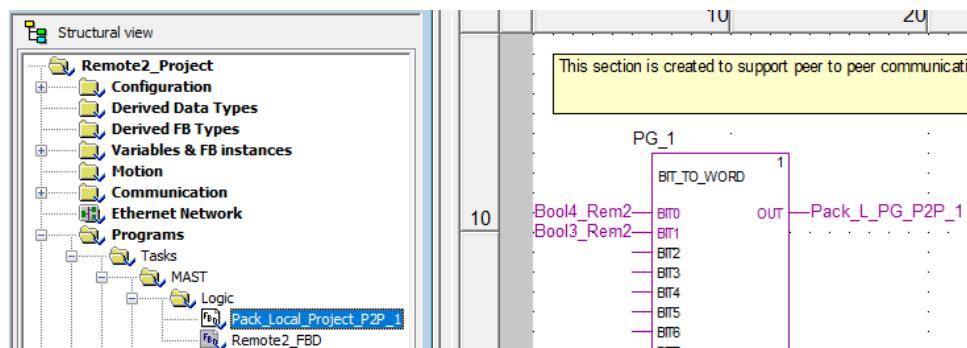
The following figure shows the value held by the intermediate variable *Read_R_RVAR_5* in *Local_Project*, which is used for peer to peer communication.



In *Remote2_Project*, section *Pack_Local_Project_P2P_1* is also added to transfer the values held by the mapped remote variables to one variable.

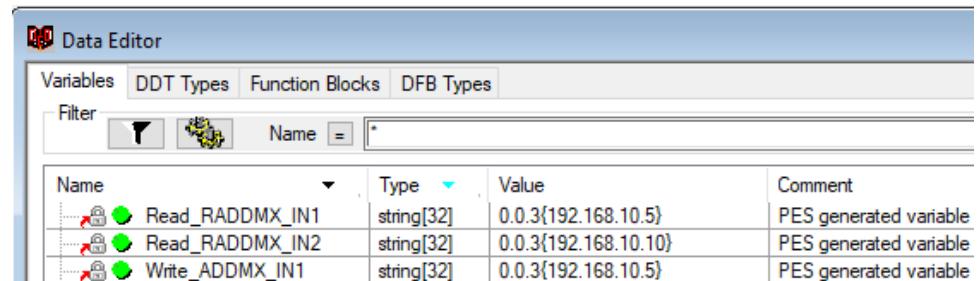


The following refinement window of *Remote2_Project* shows the logic that is added in section *Pack_Local_Project_P2P_1* to transfer the value from the two mapped remote variables *Bool3_Rem2* and *Bool4_Rem2* to one variable.



Building the Executable of *Local_Project*

After building the executable of *Local_Project*, the topological and IP addresses of the communication modules of executables of *Remote1_Project* and *Remote2_Project* appear as value for the corresponding variables.



The screenshot shows the Data Editor window with the 'Variables' tab selected. The table lists three variables:

Name	Type	Value	Comment
Read_RADDMX_IN1	string[32]	0.0.3{192.168.10.5}	PES generated variable
Read_RADDMX_IN2	string[32]	0.0.3{192.168.10.10}	PES generated variable
Write_ADDMX_IN1	string[32]	0.0.3{192.168.10.5}	PES generated variable

Supervision Project Refinement Stage

Refining the Supervision Project

Overview

The refinement process of Supervision Participant projects allows you to:

- View and modify Participant project data by using the **Refine** command. Limitations apply to the modification of project data that is managed by the software.
- Add animated graphics to pages, page 345 of the project, using the **Edit** command.
- Edit Participant project settings by using the **Advanced Settings** command.
- Add included projects and attach user files.

For each Supervision Participant project, you can use only one of the aforementioned commands at a time. However, you can edit several Supervision pages of a project at the same time.

NOTE: While you refine the Supervision Participant project or edit advanced settings, commands such as **Generate** or **Delete** are not available for pages or containers.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Supervision Projects Created in Plant SCADA Standalone

If you have created a Supervision project by using Plant SCADA opened in standalone mode (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide), when you refine any Supervision project on the same computer by using the engineering client, you can see the Supervision project that you have created outside the EcoStruxure Process Expert environment.

The same applies when you edit a Supervision page.

NOTE: Changes to externally-created Supervision projects by using the engineering client are persisted and have no impact on projects managed by EcoStruxure Process Expert. However, Schneider Electric recommends not modifying an externally-created Supervision project by using the engineering client. Work only on the Supervision project managed by EcoStruxure Process Expert, which is the active project.

Refining the Supervision Project

To refine the Supervision project, proceed as follows.

Step	Action
1	Verify that AVEVA Plant SCADA is not open outside EcoStruxure Process Expert.
2	In the Project Explorer , right-click the Supervision project you want to refine and select Refine . Result: The Supervision Participant opens extracted, page 116 and maximized.
3	Proceed with the refinement. For more information on the refinement process and restrictions that apply, refer to the topic describing the refinement of the Supervision project (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

Step	Action
4	<p>Close the Participant window.</p> <p>Result: The Save Refinement dialog box opens.</p>
5	<p>Click the appropriate button in the dialog box.</p> <p>Result:</p> <ul style="list-style-type: none"> • Yes: Closes the Supervision Participant window and keeps only changes that you have saved in the Supervision Participant. • No: Closes the Supervision Participant window and discards all changes, including those that you have saved in the Supervision Participant. • Cancel: Closes the Save Refinement dialog box and keeps the Supervision Participant window open.

Editing Supervision Project Pages

When you add an animated graphic to a page, the software adds the required included project to the Supervision Participant project if it is not yet present. You can download included projects, page 646 as soon as the software has added them.

To edit a Supervision page, proceed as follows.

Step	Action
1	Verify that AVEVA Plant SCADA is not open outside EcoStruxure Process Expert.
2	<p>In the Containers pane of the Assignment Editor, right-click the page, page 345 that you want to edit and select Edit.</p> <p>Result: The Supervision opens extracted, page 116 and maximized with the Instances browser pane next to it.</p>
3	<p>Edit the page as required.</p> <p>For more information on adding animated graphics, refer to Editing Supervision Pages (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).</p>
4	<p>Click either save button in the toolbar of the Participant window:</p>  to save changes only in the active window.  to save changes in all open windows. <p>NOTE: If you have made changes in several pages, clicking the Save All button completes the save operation faster.</p> <p>Result: The facets that encapsulate the animated graphics that you have added to a page appear in the Assignments pane when you select the page. The Assignment and Generation statuses of these facets are Assigned and Generated respectively.</p> <p>If you have deleted an animated graphic that had already been added to a page, the corresponding facet is removed from the Assignments pane.</p>
5	<p>Close the Supervision Participant window.</p> <p>NOTE:</p> <p>If you have unsaved changes and click the close button on the outer frame of the Participant window, the Save Refinement dialog box opens where you have the following choices:</p> <ul style="list-style-type: none"> • Yes: Saves changes you made to the page and closes the Participant window. • No: Discards changes and closes the Participant window. • Cancel: Does not save changes and leaves the Participant window open.

Editing Advanced Supervision Project Settings

To edit advanced project settings, proceed as follows.

Step	Action
1	Verify that AVEVA Plant SCADA is not open outside EcoStruxure Process Expert.
2	In the Project Explorer , right-click the Supervision project and select Advanced Settings . Result: The Supervision Participant opens extracted, page 116 and maximized.
3	Proceed with changes to the advanced project settings. For more information, refer to Editing Advanced Project Settings (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).
4	Click the save button in the toolbar of the Participant window to save changes. 
5	Close the Supervision Participant window. NOTE: If you have unsaved changes and you click the close button in the Participant window, the Save Advanced Settings dialog box opens where you have the following choices: <ul style="list-style-type: none"> • Yes: Saves changes you made to the advanced settings and closes the Participant window. • No: Discards changes and closes the Participant window. • Cancel: Does not save changes and leaves the Participant window open.

NOTE: You need to deploy, page 695 the Supervision project to apply changes to advanced project settings.

Adding Included Projects

When you add an animated graphic, page 495 to a Supervision page, the software automatically adds the required included project to the Supervision Participant project.

You can add additional included projects. However, you cannot add an included project if the software has already added to the Participant project an included project with the same name.

To add additional included projects to a Supervision Participant project, proceed as follows.

Step	Action
1	In the Project Explorer , right-click the Supervision project to which you want to add an included project and select Include Projects . Result: The Include Projects window opens.
2	Click Add . Result: The Open dialog box opens.
3	Browse to the location where the included project file (.ctz) is stored, select it, and click Open . Result: <ul style="list-style-type: none"> • If an included project with the same name already exists in the Participant project, the software displays a notification: <ul style="list-style-type: none"> ◦ Click OK to acknowledge. ◦ Close the Include Projects window. • If an included project with the same name already exists in the Global Constituents folder of the contents repository, the software displays a notification: <ul style="list-style-type: none"> ◦ Click OK to add the included project that exists in the contents repository. ◦ Click Cancel to cancel the process without adding the included project. • If no included project with the same name already exists, the software: <ul style="list-style-type: none"> ◦ Stores the selected file in the contents repository. ◦ Displays the selected file in the window. NOTE: The software lets you deploy, page 693 included project that you add.
4	Close the Include Projects window.

NOTE: To remove an included project from the Supervision project, select the file in the **Include Projects** window and click **Remove**. After you acknowledge the command, the software removes the file and deletes it from the contents repository.

Attaching External Files

To add external files to a Supervision project, proceed as follows.

Step	Action
1	In the Project Explorer , right-click the Supervision project to which you want to add files and select Attach User Files . Result: The Attach User Files window opens.
2	Click Add . Result: The Open dialog box opens.
3	Browse to the location where the file is stored, select it, and click Open . Result: The software: <ul style="list-style-type: none">• Stores the selected file in the contents repository.• Displays the selected file in the window. NOTE: The software includes files that you attach into the Supervision project, which it deploys during deployment.
4	Close the Attach User Files window.

NOTE: To remove a file from the Supervision project, select the file and click **Remove**. The software removes the file and deletes it from the **contents repository**. If user files had already been deployed, they are not removed from the location they were deployed to when you re-deploy the Supervision project, page 696.

Configuration Stage

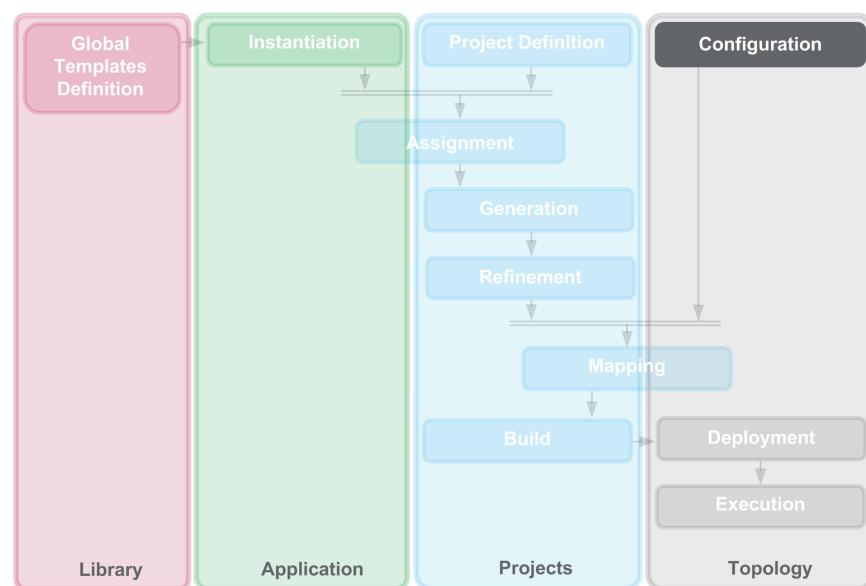
What's in This Chapter

Creating the Topology of the System.....	499
Managing the Topology.....	548
Exporting and Importing Devices to/from CSV Format.....	564

Overview

This chapter provides information to help you model the entire topology of the system, which consists of the hardware and software infrastructure.

The following figure shows the position of the **Configuration** stage within the system engineering life cycle.



Refer to the Configuration stage, page 49 for a description of the purpose of this stage.

Creating the Topology of the System

Overview

This section describes how to use the **Topology Explorer** to create and configure topological entities modeling the topology of the system.

The software includes templates that allow you to model the following infrastructure components:

- Ethernet networks
- Station nodes
- Controllers
- Ethernet STB islands and devices connected through CANopen extension
- Modbus TCP and EtherNet/IP I/O devices
- PROFIBUS remote masters (PRMs) and their decentralized peripherals (DPs)

Controllers and station nodes act as engines to which you can deploy Control or Supervision projects.

For more information on supported architectures, refer to the topic describing integrated architectures (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Organizing the System Topology

Before Starting

The topology of the system is defined through topological entities, page 29, which consist of instances of topological templates, called devices. You can organize these entities in the form of a folder structure, which is displayed in the tree view of the **Topology Explorer**.

Before creating the folder structure for the topology, define the appropriate segmentation and naming convention for the hardware infrastructure of the system.

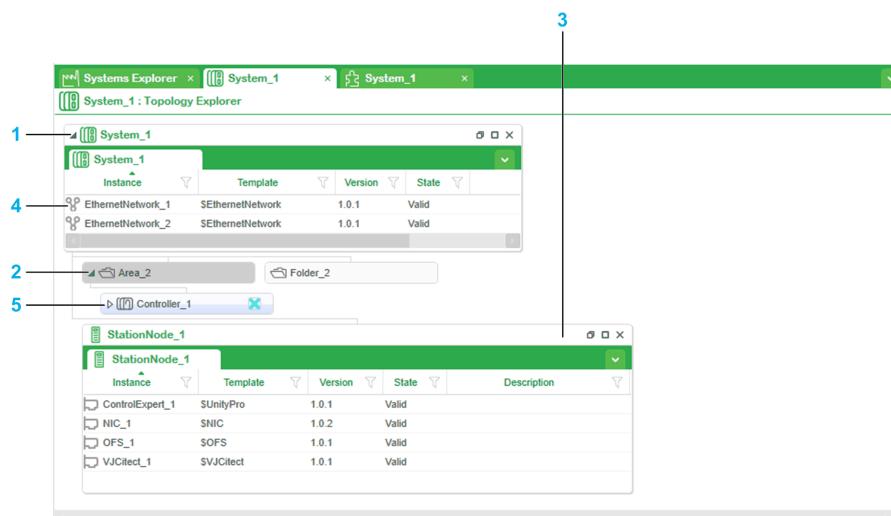
NOTE: Before selecting another command for an entity, wait for the current task to complete. However, the same command can be executed on distinct entities by one or more clients simultaneously.

Opening the Topology Explorer

To open the **Topology Explorer**, right-click the system from the **Systems Explorer** tree view and select **Open Topology**.

Topology Explorer Window

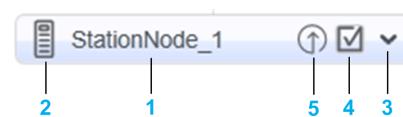
The following figure shows an example of the **Topology Explorer** window.



Item	Description
1	System root folder
2	Topology folder tree view
3	Station node entity (expanded view to display contents)
4	Ethernet networks
5	Controller entity (collapsed view)

Topological Entity Description

The following figure shows an example of topological entity.



Item	Description
1	Identifier of the entity.
2	Icon indicating the type of entity.
3	Show/hide devices and services button (station nodes). For other types of entities, such as controllers, an expand/collapse button appears instead to the left of the icon once the entity has been successfully configured.
4	Mapping indicator. The entity is mapped when a check mark is displayed. It is not mapped when a cross is displayed. Hover with the pointer over the indicator to display in a tooltip the mapping status.
5	Online refinement indicator. The icon indicates that you have made changes online to the Control Participant project that is deployed to the engine. These changes need to be applied to the associated logical Control Participant project and controller configuration to make them consistent with the project that is deployed to the engine. The icon is removed once you successfully update the associated logical Control project, page 748 and the configuration with all the changes that the software has detected in the following sections of the Update Project dialog box: <ul style="list-style-type: none">• Derived Data & FB Types• Variables & FB Instances• Sections• Hardware If you update the project only partially with these categories, the icon remains visible. Hover with the pointer over the indicator to display a tooltip. NOTE: If you have also made changes pertaining to the Animation Tables and/or Operator Screens category, it is not relevant whether you select or not any of these two categories as long as you select all the changes in the aforementioned categories to remove the refinement indicator. If you have made a change online pertaining only to the Animation Tables and/or Operator Screens category, you need to select such change to remove the indicator. NOTE: For a station node in which several Control services exist, which represent controller simulator instances, the icon is displayed as soon as online changes have been made by using either one. A tooltip indicates the identifier of the Control projects that the icon applies to.

Creating Topology Folders

The **Topology Explorer** allows you to organize the system infrastructure in the form of a folder structure.

To create a topology folder, proceed as follows.

Step	Action
1	Right-click the system root folder in the Topology Explorer and select Create Folder . Result: A topology folder is created and displayed in the tree view. NOTE: To create a subfolder, repeat this step at the topology folder level.

Topology Folder Actions

Right-click the system root folder or a topology folder to open a context menu with the following commands.

Command	Description	
Expand	<p>The first time you click Expand, expands the next level in the folder structure or opens the folder/window.</p> <p>The next time you click Expand, displays the folder structure or contents in its last expanded state.</p>	
Collapse	<p>Collapses the folder structure up to the folder or closes the folder/window.</p> <p>The command is available only if the folder/window is already expanded.</p>	
Create Folder	<p>Creates a topology folder or subfolder and displays it in the tree view.</p>	
Open Application	<p>Opens the Application Explorer, page 153, which allows you to start the instantiation stage or access the application of the system.</p> <p>The command is available from the root folder only.</p>	
Open Project	<p>Opens the Projects Explorer, which allows you to start the Project Definition stage, page 254 or access the Control and Supervision projects of an existing system.</p> <p>The command is available from the root folder only.</p>	
Create Ethernet Network, page 538	<p>Creates an Ethernet network entity and displays it in the folder. The entity represents an Ethernet network of the system.</p>	
Create Station Node, page 516	<p>Creates a station node entity and displays it in the tree view.</p> <p>For example, a station node can represent a Supervision I/O server or a PC emulating simulator software.</p>	
Create Controller, page 507	<p>Displays a submenu, which allows you to create a controller entity of the supported platforms (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).</p> <p>The topological entity allows you to configure devices, which represent the CPU, power supply, communication, I/O modules, extended and remote racks, and so on.</p>	
Create STB Island, page 523	<p>Creates an STB island entity and displays it in the tree view.</p> <p>The topological entity represents the segments, communicator, STB island modules, and devices connected through the CANopen bus of the island once you have configured them.</p>	
Create PRM PROFIBUS DP, page 527	<p>Creates a PROFIBUS remote master (PRM) entity and displays it in the tree view.</p> <p>The topological entity allows you to configure the decentralized devices (DPs) connected to the PRM by using device type managers (DTMs) and generic station description (GSD) files.</p>	
Create Device IO, page 531	<p>Opens a submenu containing the following commands:</p> <ul style="list-style-type: none"> • Modbus TCP • EtherNet/IP <p>Selecting either command opens the Select Template dialog box, which lets you select and instantiate a topological template to create an I/O device entity. The topological entity is used to manage communication, page 779 between such device and the controller by using Modbus TCP (I/O scanning and explicit) or EtherNet/IP (explicit) messaging.</p>	
Update Template, page 550	<p>Updates the template that is used by entities inside the folder and any subfolder with the latest version of that template, which is available in the Global Templates library on the system server.</p>	
Export	<p>Opens a submenu with the following commands:</p>	
	<table border="1" data-bbox="635 1763 1438 1954"> <tr> <td data-bbox="635 1763 794 1954" style="vertical-align: top;">Topology, page 552</td><td data-bbox="794 1763 1438 1954"> <p>Opens the Export window, which allows you to select the topological entities that you want to export in the form of a topology export file (.sbk).</p> <p>You can select entities contained in the folder and any subfolders along with their folder hierarchy.</p> <p>The command is not available from empty topological folders.</p> </td></tr> </table>	Topology, page 552
Topology, page 552	<p>Opens the Export window, which allows you to select the topological entities that you want to export in the form of a topology export file (.sbk).</p> <p>You can select entities contained in the folder and any subfolders along with their folder hierarchy.</p> <p>The command is not available from empty topological folders.</p>	
<table border="1" data-bbox="635 1954 1438 2106"> <tr> <td data-bbox="635 1954 794 2106" style="vertical-align: top;">Devices, page 564</td><td data-bbox="794 1954 1438 2106"> <p>Opens the Export window, which allows you to export the devices, Ethernet networks, and physical connections of the topology in the form of a topology export file (.csv).</p> <p>The command is available only from the system root folder.</p> </td></tr> </table>	Devices, page 564	<p>Opens the Export window, which allows you to export the devices, Ethernet networks, and physical connections of the topology in the form of a topology export file (.csv).</p> <p>The command is available only from the system root folder.</p>
Devices, page 564	<p>Opens the Export window, which allows you to export the devices, Ethernet networks, and physical connections of the topology in the form of a topology export file (.csv).</p> <p>The command is available only from the system root folder.</p>	

Command	Description	
Import	Opens a submenu with the following commands:	
	Topology , page 555	Opens the Import window, which allows you to select an export file (.sbk). In a second step, you can select which compatible content of the export file you import in the topology.
	Devices , page 571	Opens the Import window, which allows you to select a device export file (.csv). In a second step, you can select which devices of the export file you import in the topology. The command is available only from the system root folder.
Delete	Deletes the folder and its contents, including subfolders, their contents, and associated configurations. The command is not available from the root folder.	
Rename	Allows you to type a new folder name. The command is not available from the root folder.	
Properties	Opens the folder Properties window, which allows you to view and edit information of the folder. The command is not available from the root folder.	

Topology Folder Properties

To access the **Properties** window of the topology folder, right-click the folder and select **Properties**.

You can edit the following items.

Item		Description
General	Identifier	Designation of the folder. The identifier must be unique for each subfolder of the same parent folder and satisfy naming rules, page 88. Default value: Folder_n where n is an incremental number starting at 1.
	Description	Optional. You can enter a description of the folder with free form text. The description you enter appears in a tooltip when you move the pointer over the folder. Default value: Blank
	Area	Optional. You can assign an access control number to the folder. Value: 1...255 (integer). Default value: Blank

Selecting Topological Templates

Overview

When you create a topological entity, the software displays the **Select Template** dialog box if more than 1 topological template is available to model a hardware module:

- When you first create the entity: For any module that you add in the Control Participant.
- When you modify the entity: For modules that you have added or moved since you last configured the entity.

The **Select Template** dialog box allows you to select the topological template and version that you want to use among the compatible ones that are available.

NOTE: The **Update Templates** command, page 550 is available from the context menu of entities after you create them.

Default Selection

By default, the software selects the topological template in the Global Templates library that has the highest version number with the usability state (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of **Approved** or **Deprecated** among those that have a matching **Hardware Reference** as attribute.

Selection for Redundant Controllers

When you select a topological template for a module of the primary controller of a redundant configuration, the software applies your selection automatically to the counterpart module of the standby controller.

Select Template Dialog Box

The table describes the fields of the **Select Template** dialog box and how to select templates.

Column Header	Description
Hardware References	Module hardware references and network type (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) for which a choice of topological templates is available.
Templates	For each displayed hardware reference, indicates the identifier and version of the default topological template that the software will use to create the device. <ul style="list-style-type: none"> • To select a different template: Click the default template to display the other available templates, select the template that you want to use, and click OK to confirm your selection. • To cancel the configuration process and return to the Control Participant window, click Cancel.

Using Configuration Change on the Fly (CCOTF)

Before Performing CCOTF Modifications

The CCOTF functionality of the Control Participant is enabled by default when you create a controller of the M580 or M580 safety platform in the **Topology Explorer**. That is, the *Online modification in RUN or STOP* parameter of the CPU is selected.

It allows you to deploy compatible hardware configuration and parameter changes to the controller while it is running without impacting other modules or ongoing tasks.

Before performing a CCOTF-compatible modification, refer to the topics describing the CCOTF functionality in the help of the Control Participant, page 93 for your specific controller platform and follow the recommendations that are given.

CCOTF with EcoStruxure Process Expert

In EcoStruxure Process Expert, CCOTF already comes into play when you configure a controller (offline) of the M580 or M580 safety platform in the **Topology Explorer** to which you have already deployed a built Control project.

When CCOTF is enabled, the software informs you if a configuration or parameter change that you are about to make will require stopping the controller when you deploy changes to the controller later on during the deployment stage. That is, it must be built offline.

NOTE: If you configure drops of the S908 architecture or make changes to an existing drop of this architecture during configuration of the controller (offline), you must use the **Build All**, page 636 and **Deploy Built Project** commands thereafter, which requires stopping the controller. To perform CCOTF-compatible configurations of drops of the S908 architecture, use the **Refine Online** command instead.

NOTE: CCOTF is not supported for controllers of the M340 and Quantum platforms.

Enabling and Disabling CCOTF

By configuring the controller, you can disable or enable the *Online modification in RUN or STOP* parameter in the Control Participant **Configuration** section of the CPU.

If you have already deployed a built Control project to the controller, enabling or disabling the parameter will require that you deploy once to the controller by using the **Deploy Built Project** command. This requires stopping the controller.

Configuration Changes not Compatible With CCOTF

The following types of changes can be incompatible with CCOTF and require stopping the controller:

- A single change that is saved during controller configuration.
For example, adding a NOC communication module in the local rack of an M580 controller.
- A series of changes of the same type, which exceeds the number of allowed consecutive changes in a same transaction.
For example, adding five digital input modules to the local rack of an M580 controller (the limit for CCOTF is four modules).

For details on the CCOTF limitations, refer to the Control Participant help on CCOTF for each controller platform.

Working Principle

When you perform a series of CCOTF compatible changes, you can build the executable of the associated Control project in the **Project Explorer** between changes. The changes are still part of the same transaction. A transaction is executed and the CCOTF counter is reset when you use the **Deploy Changes/Undo Online Changes** or **Deploy Built Project** command on the controller.

When the software detects that a configuration change will require stopping the controller, a dialog box opens. It gives you the choice to proceed with the change or cancel it. The following table describes the impact of either choice.

Action selected by the user	Impact on CCOTF	Impact on future configuration changes
Yes	<p>The configuration change is performed. If you save the change when closing the Participant configuration window, the Deploy Changes/Undo Online Changes command is disabled and deploying the change by using the Deploy Built Project command requires stopping the controller.</p> <p>NOTE: Closing the Participant configuration window without saving discards the change and CCOTF is still available.</p> <p>NOTE: Undoing the CCOTF-incompatible configuration change does not revoke the controller stop if you save your changes.</p>	<p>You are not informed anymore if you make CCOTF-incompatible configuration changes until you deploy to the controller.</p> <p>The Build command in the context menu of the executable is disabled. Only Build All can be used.</p>
Cancel	The configuration change is discarded and you can continue with the configuration of the controller.	You continue to be informed when you make a CCOTF-incompatible configuration change.

NOTE: Using the **Deploy Built Project** command also resets the CCOTF counter.

CCOTF in the System Engineering Life Cycle

The following table summarizes the main impact of the CCOTF functionality on the other stages of the system engineering life cycle. For details, refer to the description of each stage.

Stage	Impact of CCOTF
Refinement (offline)	When you perform a change in the Control project, page 436 that is not CCOTF-compatible, the software informs you ahead of time that it will require stopping the controller when you deploy changes later on.
Build	Changes related to the topology that are CCOTF-compatible can be built, page 636 by using the Build command.
Deployment	Changes related to the topology that are CCOTF-compatible can be deployed, page 682 by using the Deploy Changes/Undo Online Changes command without stopping the controller. If you have made a CCOTF-incompatible change, only the Deploy Built Project command is available, which requires stopping the controller.
Execution	It is required to update the project, page 748 with configuration changes that you have made online to be able to use the Deploy Changes/Undo Online Changes command.

Configuring a Controller

Overview

Controller entities act as engines for executing Control projects that are created in the **Projects Explorer**.

You proceed with the configuration of controller entities from within the Control Participant, which then transfers the defined configuration to the controller entity of the **Topology Explorer** by using the appropriate topological templates.

If the software cannot find the corresponding template for a hardware module, it uses the *\$UnknownModule* template. Hardware modules that are modeled with this template do not appear in the **Hardware Mapping** editor.

NOTE: For additional information on configuring M580 safety controllers, refer to the topic describing the engineering of M580 safety systems, page 759.

Device DDTs of Topological Entities

When you add a module to a controller entity, which uses a device type manager (DTM) (for example, an M580 CPU, a NOC communication module, or a HART analog I/O module), the software automatically creates the corresponding device DDT. Such variables exist only in the configuration that you can access from the **Topology Explorer** and are integrated to the Control Participant project during build.

However, you can create certain device DDTs, page 445 in a Control Participant project during refinement.

Redundant Controllers

Typically, for redundant controllers that are configured in the topology, only the entity identifier or the identifier of the primary engine is used at the platform level to identify them (for example, in the **Physical Connections** dialog box when you connect the controller to an Ethernet network). The standby engine is implicitly included. If the standby engine needs to be dealt with explicitly (for example, for deployment tasks), it is identified separately.

Creating a Controller Entity

To create the controller entity, which will contain the controller configuration (devices), proceed as follows.

Step	Action
1	Right-click the system root folder or a topology folder in the Topology Explorer , select Create Controller , and select a controller platform. Result: The controller entity, page 500 is created and displayed in the tree view.

Controller Actions

Right-click the controller entity to open a context menu with the following commands.

Command	Description
Configure , page 511	Opens the Control Participant, which allows you to configure the controller hardware, in-rack modules, drops, and communication services.
Physical Connections , page 542	Opens the Physical Connections window, which allows you to connect or disconnect communication modules of the controller that support Ethernet to existing Ethernet networks to represent physical connections to such networks.

Command	Description
Update Template, page 550	Opens the Update Template dialog box, which allows you to update the templates that are used by controller entities with the highest available version of the Global Templates library on the system server.
Deploy Built Project, page 649	Opens the Deploy Built Project dialog box, which lets you select a built Control Participant project and deploy it entirely to the controller. It can also be used to remove refinements made online.
Deploy Changes / Undo Online Changes, page 649	Deploys to the controller the differences between the deployed Control Participant project and the corresponding built Control Participant project.
Re-Deploy Last Project, page 649	Lets you deploy once again the entire built Control Participant project that you had deployed to the controller last. The command is available only if you had executed the Deploy Built Project command previously on the controller.
Deploy Data, page 677	Opens the Deploy Data dialog box in which you can select a controller data backup file that has been created by using the Back Up Data command. It lets you deploy the data contained in the file to the controller. The command available only if you have deployed a built Control Participant project to the controller.
Start, page 707	Starts the execution of the participant project by the controller.
Stop, page 707	Stops the execution of the participant project by the controller.
Refine Online, page 707	Lets you modify the deployed Control Participant project online.
Sync (Primary→Standby), page 719	Lets you synchronize the standby controller with the primary one. The command is available only for redundant controllers.
Update Project, page 707	Updates the logical Control Participant project that is associated to the deployed Control Participant project with the changes that you have made to the deployed project online by using the Refine Online command. It also requires that you update the controller configuration with hardware changes that you have made during online refinement.
Back Up Data, page 733	Opens the Back Up Data dialog box, which lets you configure the data that you want to back up and create a controller data backup file. The command available only if you have deployed a built Control Participant project to the controller.
Manage Password	Opens the Manage Password dialog box, which lets you set and modify the Controller, page 661 and/or the Safety, page 770 password. Each command is available only if the corresponding property is enabled in the properties of the controller.
Clear Password	Opens the Clear Password dialog box, which lets you remove the Controller password, page 662 of the deployed Control Participant project and/or the Safety, page 770 password. Each command is available only if you have set the corresponding password.
Forgot Password	Opens the Forgot Password dialog box, which contains an authentication code that you need to reset the forgotten Controller, page 663 or Safety, page 770 password.
Delete, page 514	Deletes the controller entity including controller devices and related configurations. NOTE: For M580 safety controllers, deletes also the associated M580 safety Control project, page 761.
Rename	Lets you change the identifier of the controller entity.
Export, page 552	Opens the Export window, which allows you to export the entity and its folder hierarchy in the form of a topology export file (.sbk).
Properties	Opens the Properties window in which you can view and edit properties of the controller entity. NOTE: To view properties of the controller devices representing the hardware modules, refer to Controller Representation, page 512 .

Controller Properties

Right-click the controller entity and select **Properties** to open the **Properties** window in which you can view and edit its properties.

Item		Description
General	Identifier	<p>The identifier must be unique within the system and satisfy naming rules, page 88.</p> <p>The entity identifier becomes part of the name of the devices that the software creates inside in the entity:</p> <ul style="list-style-type: none"> • Folders (identifier field) • Devices (\$Name parameter) <p>The identifier (or part of it) is also used to create the name of the DTM if it is added during build when the controller is acting as server for peer to peer communication, page 631.</p>
	Description	<p>Optional.</p> <p>You can enter a description of the controller with free form text. The description you enter appears in a tooltip when you move the cursor over the entity.</p> <p>Default value: Blank</p>
	Area	<p>Optional.</p> <p>You can assign an access control number to the controller entity.</p> <p>Range: 1...255 (integer)</p> <p>Default value: Blank</p>
Hot Standby <small>(1)</small>	ControllerExist	<ul style="list-style-type: none"> • PrimaryAndStandby: Select this value when both the primary and the standby controllers are physically present and connected by a Hot Standby cable link (default value). • Primary: Select this value when either of the following applies: <ul style="list-style-type: none"> ◦ Only the primary controller is physically present. ◦ Both the primary and the standby controllers are physically present but not connected by a Hot Standby cable link. <p>The value has an impact on the behavior during deployment and execution, page 648.</p>
Controller Family	Value	<p>Indicates the hardware platform that the controller belongs to.</p> <p>Read-only.</p>
CpuVersion	CpuVersion	<p>Indicates the firmware version of the CPU module that is configured.</p> <p>The read-only property is displayed only after you save the controller configuration.</p>

Item	Description
\$Name of communication module ⁽²⁾	<p>ServerMemoryStart</p> <p>Holding register address (%MW) for peer to peer data. (Integer) Default value: Blank You can only enter an odd address, page 633 for this parameter (for example, 1, 5, 11). For more information, refer to Configuring the Controller I/O Scanner Parameters, page 512.</p>
	<p>ServerMemoryLength</p> <p>Holding register size for peer to peer data. (Integer) Default value: Blank. NOTE: The available memory length is <i>ServerMemoryLength</i> - 1 because 1 word is reserved to move the channel address for managing the 32-bit alignment, page 633 for peer to peer communication across Modicon M340, M580, and Quantum platforms. For more information, refer to Configuring the Controller I/O Scanner Parameters, page 512.</p>
\$Name of communication module ⁽²⁾	<p>ClientMemoryStart ⁽³⁾</p> <p>Holding register address (%MW) for I/O scanning. (Integer) Default value: Blank. For more information, refer to Configuring the Controller I/O Scanner Parameters, page 512.</p>
	<p>ClientMemoryLength⁽³⁾</p> <p>Holding register size in words for I/O scanning (Integer) Default value: Blank. For more information, refer to Configuring the Controller I/O Scanner Parameters, page 512.</p>
Password Protection	<p>Controller</p> <p>Lets you manage password protection, page 658 for deployment and execution operations performed on the controller.</p> <ul style="list-style-type: none"> • Enabled (default): Enables password protection. • Disabled: Disables password protection after you confirm the selection. You cannot disable the property when the System Access Password property, page 137 of the system is enabled. <p>By default, the value is set according to the value of the Controller Access Password at Creation property, page 137, which is configured at the system level.</p> <p>NOTE: To disable the property once you have set a password, you need to enter the current password, which also clears it.</p> <p>If you had already deployed a Control Participant project while the property was set to Disabled, you can enable it and set a password for the already deployed Control project.</p>
	<p>Safety</p> <p>The property is available only for M580 safety controllers and lets you manage safe area password protection, page 770.</p> <ul style="list-style-type: none"> • Enabled (default): Makes the use of a password mandatory when performing operations on the safety configuration and/or safety program of the controller. • Disabled: Disables password protection after you confirm the selection. <p>To disable password protection when a password is set, you must enter it. This also clears the password.</p>

(1) The section is displayed only for redundant controller configurations.

- (2) The section is displayed when the software has completed the configuration process. A section is displayed for each communication module that you have configured with an IP address in the Control Participant by using the **Configure** command. The server properties are displayed for CPU modules only.
- (3) The client properties are displayed for communication modules, including the CPU module when the I/O scanner service is enabled. They are not displayed for controllers of the Modicon M580 platform and NOC communication modules.

NOTE: To view properties of the controller devices representing the hardware modules, refer to [Controller Representation](#), page 512.

Configuring the Controller

To configure the controller, proceed as follows.

Step	Action
1	Right-click the controller entity in the Topology Explorer and select Configure . Result: The Control Participant opens extracted, page 116 and maximized.
2	Proceed with the configuration of the controller (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) inside the Control Participant.
3	Click the save button in the toolbar of the Participant window.  Result: The software saves the changes in the configuration file and transfers the configuration to the Topology Explorer . It may display the Select Template , page 504 dialog box. NOTE: If the software detects a configuration issue, it displays a notification, which allows you to change the configuration until you are able to save it.
4	Close the Control Participant window. NOTE: If you change the configuration and you click the close button on the outer frame of the Participant window without saving your changes first, the software displays the Save Configuration dialog box where you can click the following: Yes , to save the changes in the configuration file and close the Participant window. No , to discard your changes and close the Participant window. Cancel , to leave the Participant open without saving your changes.

Changing the Controller Configuration

To change the configuration of a controller, follow the procedure to configure a controller, and proceed with the changes inside the Control Participant.

If the controller is mapped to at least 1 executable, saving changes to communication functions that you made from within the Control Participant may delete the existing communication mapping, page 582 and hardware mapping, page 602 of this executable.

In such case, the software displays a notification asking you to confirm the change. Make either selection:

- **OK:** To save the change to the configuration file of the controller and delete the associated communication and hardware mappings.
- **Cancel:** To keep the Control Participant window open. Unsaved changes are not discarded

For more information, refer to [Configuring the Controller](#) (see [EcoStruxure™ Process Expert, Control Participant Services, User Guide](#)).

Configuring the Controller I/O Scanner Parameters

To use the I/O scanner of the controller and define communication channels between the CPU and/or communication modules and devices connected to the same Ethernet network, configure the read/write zones once the software has completed the configuration process to transfer the controller configuration to the **Topology Explorer**.

The software configures the related parameters of the devices representing the CPU/communication modules and exposes the data in the **Communication Mapping Editor** window during the communication mapping, page 582.

If you configure overlapping client and server memory areas for a controller and map variables to these overlapping addresses, build does not complete successfully and the software displays a message to inform you.

These settings are an example of values entered in the **Properties** window of a controller that result in overlapping memory areas:

- **ServerMemoryStart:** 1
- **ServerMemoryLength:** 100
- **ClientMemoryStart:** 51
- **ClientMemoryLength:** 150

For more information on configuring parameters for I/O scanning, refer to the Control Participant help.

NOTE: If you reduce the value of the size, page 586 that is reserved in the client and/or server memory while peer to peer communication channels exist in the controller and you have already built the associated logical Control Participant project, verify that the new register size meets the requirements of the project. You cannot build the project if the free memory value becomes negative.

NOTE: When you modify a parameter of the client and/or server memory reserved for communication channels and you have already deployed the associated built Control Participant project, you can only apply the change to the engine by using the **Deploy Built Project**, page 686 command, which requires stopping the controller. You cannot use the **Deploy changes** command.

To configure the I/O scanner parameters of the controller, proceed as follows.

Step	Action
1	In the Topology Explorer , right-click the controller folder and select Properties .
2	Enter a value in the appropriate fields of the memory section, page 509 of the CPU and/or communication module that you want to use for I/O scanning.
3	Click the close button to close the Properties window.

NOTE: For controllers of the Quantum platform configured with a NOC communication module and acting as client, configure the memory size of the NOC (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) by using the Control Participant.

Controller Representation

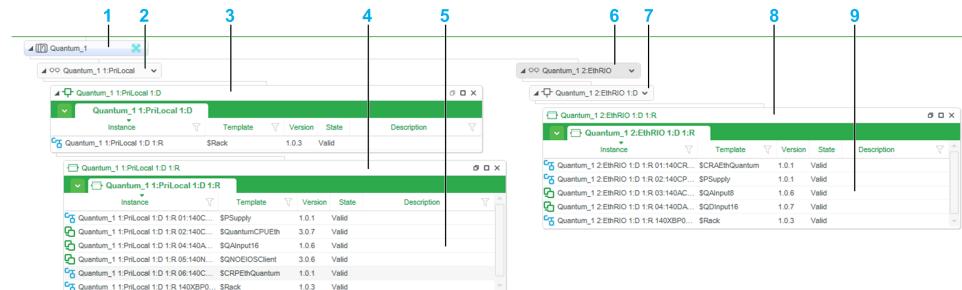
The **Topology Explorer** allows you to view the topological devices representing the following hardware components of the controller entity once you have configured it:

- Racks
- Buses
- Power supplies
- CPUs
- Communication modules

- I/O modules

It lets you also view and configure their properties, page 514.

The following figure shows the representation of a controller entity based on the Modicon M580 platform with a local rack and one Ethernet RIO drop.



Item	Description
1	Controller folder representing the controller entity, page 500
2	Controller local bus folder
3	Controller local rack folder
4	Controller local rack module folder
5	Devices modeling the rack, power supply, CPU, communication modules, and I/O modules of the local rack
6	EIO bus folder
7	EhRIO rack folder
8	EhRIO drop folder
9	Devices modeling the rack, CRA module, and I/O modules of the EhRIO drop

Devices are identified by the \$Name parameter, which uses the format <Entity identifier> <Bus number>:<Designation> <Drop number>:D <Rack number>:R <Position on rack>:<Commercial reference>.

<Designation> is either:

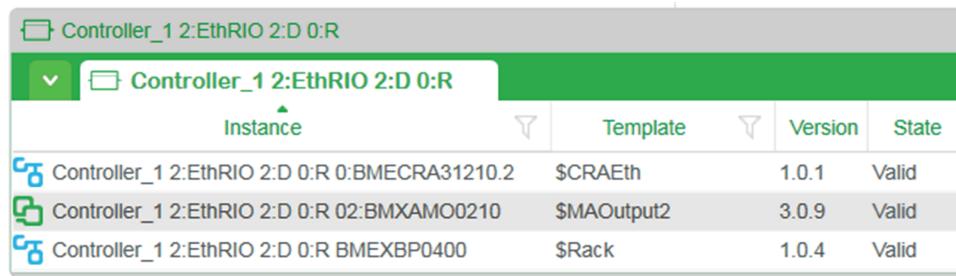
- Local** for the local bus of controller platforms that do not support a redundant configuration.
- PriLocal** or **SecLocal** for the local bus of controller platforms that support a redundant configuration. If a standalone controller is configured, the local bus designation is **PriLocal**.
- EhRIO** for the EIO bus of any controller platform supporting remote drops.

In this example, the highlighted row in the following figure represents a BMXDDI1603 digital input module located in position 3 of the local rack of Controller_1.

The screenshot shows a table of devices in the local rack of Controller_1. The table has columns for Instance, Template, Version, and State. One row is highlighted in grey, representing the BMXDDI1603 module.

Instance	Template	Version	State
Controller_1:0:PriLocal 0:D 0:R 0:BMEP584040	\$M580CPUEthCRP	1.1.7	Valid
Controller_1:0:PriLocal 0:D 0:R 02:BMENOC0301.2	\$MNOCIOSClient	1.0.7	Valid
Controller_1:0:PriLocal 0:D 0:R 03:BMXDDI1603	\$MPDInput16	4.0.3	Valid
Controller_1:0:PriLocal 0:D 0:R BMEXBP0800	\$Rack	1.0.4	Valid
Controller_1:0:PriLocal 0:D 0:R BMXCPS2000	\$PSupply	1.0.1	Valid

In this example, the highlighted row in the following figure represents a BMXAMO0210 analog output module located in position 2 of the second remote drop of Controller_1.



Instance	Template	Version	State
Controller_1 2:EthRIO 2:D 0:R 0:BMECRA31210.2	\$CRAEth	1.0.1	Valid
Controller_1 2:EthRIO 2:D 0:R 02:BMXAMO0210	\$MAOutput2	3.0.9	Valid
Controller_1 2:EthRIO 2:D 0:R BMEXBP0400	\$Rack	1.0.4	Valid

NOTE: For CPUs of the Modicon M580 platform that have an embedded RIO scanner service:

- The parent folder with the name <Entity identifier> <Bus number>:**EthRIO** is created by default. The folder and the device it contains represent the RIO scanner service. The status of the device remains invalid until you configure at least one RIO drop.
- The <Entity identifier> <Bus number>:**EthRIO** device appears in the **Physical Connections** dialog box, page 540 even if you have not yet configured a RIO drop. It is associated to **IP Address A** of the CPU module. In case of a redundant controller, it also implicitly represents the RIO scanner service of the CPU designated as **B** and is associated to **IP Address B**.

Properties of Modules

Expand the topological entity folder, right-click a device, and select **Properties** to open the **Device Editor**, which lets you view some of its parameters without the need to open the Control Participant. For example:

- Its reference
- Its position on the rack
- IP addresses, ports, and other module-specific communication parameters
- Physical interface links with Ethernet networks
- Implicit connections within the entity

Deleting a Controller Entity

When you select the **Delete** command for a controller entity, the software notifies you if it detects any of the following:

- Any mappings.
- Any physical interface link with an Ethernet network.

Once you have confirmed the delete command, the software deletes the following components of the controller entity:

- Physical interface links
- The controller entity, subfolders, and devices contained therein
- The corresponding Control Participant configuration project, including:
 - Any mapping information
 - Peer to peer communication channels, page 591
 - Any deployed Control Participant project related to the engine

To delete the controller entity and its contents, proceed as follows.

Step	Action
1	<p>Right-click the controller entity in the Topology Explorer and select Delete.</p> <p>Result: A dialog box opens.</p>
2	<p>Verify the information that is displayed and click OK to proceed.</p> <p>Result: The software deletes the controller entity and its contents, and displays a confirmation message in the notification panel.</p> <p>NOTE: For M580 safety controllers, deletes also the associated M580 safety Control project, page 761.</p>

Configuring a Station Node

Overview

The station node is a topological entity, which represents the hardware (for example, a network interface cards (NICs)) and software (for example, Supervision servers and clients, or OPC UA Server Expert) that is installed on a computer workstation, which is connected to the Ethernet control network.

Station nodes act as engines for executing:

- Deployed Supervision projects.
- Deployed Control projects running in a Controller simulator, page 654.

For example, station nodes represent operator stations and operation servers.

You create station node entities in the **Topology Explorer** and define their devices from the station node folder representing the entity.

Creating a Station Node Entity

To create a station node entity, proceed as follows.

Step	Action
1	<p>Right-click the system root folder or the topology folder in the Topology Explorer and select Create Station Node.</p> <p>Result: The software creates a station node entity, page 500 and displays it in the tree view.</p> <p>NOTE: By default, the software creates a Network Interface Card (NIC) device when you create a station node entity. To display the NIC device, open the station node entity folder.</p>

NOTE: The software may display the **Select Template**, page 504 dialog box.

Station Node Actions

Right-click the station node folder to open a context menu with the following commands.

Command	Description
Create NIC , page 520	Creates an NIC device and displays it in the station node folder. The NIC provides connectivity to the Ethernet network through the Physical Connections , page 540 dialog box.
Create Supervision Service	Creates a Supervision service instance and displays it in the station node folder. It lets you select the station node in the service mapping, page 622 so that you can deploy a Supervision project to it. Create the service in the stations nodes on which the I/O ⁽¹⁾ , Alarm, Trend, and Report servers as well as the Supervision clients are installed. You can create one Supervision service per station node. Right-click the service and select Properties to access its parameters, page 522. (1) May also require an OFS service.
Create OFS	Creates a protocol-independent OFS service instance and displays it in the station node folder. Create the service in the station node on which the I/O server and OPC Factory Server or OPC UA Server Expert are installed. It is required to map the station node to the IOServer service of the Supervision project in the service mapping, page 622 and to deploy the corresponding configuration files, page 643 to the station node. You can create one OFS service per station node. Right-click the service and select Properties to access its parameters, page 522. NOTE: Requires also a Supervision service. NOTE: The OFS service is not required to perform the service mapping with the IOServer service when the OPCUA Embedded protocol is selected, page 341 for all the tag containers associated to the same I/O server.
Create Control Service	Creates a Control Expert service instance and displays it in the station node folder. It allows you to: <ul style="list-style-type: none">• Deploy, page 649 a built Control Participant project to the Controller simulator that is running on the computer.• Select the station node/ControlExpert service to which you have deployed a Control project in the communication mapping, page 624. It lets you use it as data source (I/O device) for simulating Supervision runtime services. You can create several ControlExpert services per station node. Each one corresponds to a Controller simulator instance that uses a specific port. You need to enter a unique port value when you create a service. Right-click a service and select Properties to access its parameters, page 522.
Physical Connections , page 543	Opens the Physical Connections window, which allows you to connect or disconnect the station node, through its NIC, to existing Ethernet networks to represent physical connections to such networks.
Update Template , page 550	Opens the Update Template dialog box, which allows you to update the templates that are used by station node entities with the highest version that is available in the Global Templates library on the system server.
Control	Opens a submenu containing deployment and execution commands for Control projects. See the next table a description.
Supervision	Opens a submenu that contains deployment and execution commands for Supervision projects. See the next table a description.
Delete	Deletes the station node entity including services and related configurations.

Command	Description
Rename	Lets you change the identifier of the station node entity.
Export , page 552	Opens the Export window, which allows you to export the entity and its folder hierarchy in the form of a topology export file (.sbk).
Properties	Opens the Properties window in which you can view and edit properties of the station node entity.

The following table describes the commands of the **Control** submenu. Commands in this submenu are available only if you have created the necessary service in the station node, mapped an executable to the station node, and built it. Additional availability conditions may apply; refer to the description of each command.

Command	Description
Deploy Built Project	Opens the Deploy Built Project dialog box, which lets you select a built Control Participant project and deploy it to a Controller simulator, page 649.
Deploy Changes / Undo Online Changes , page 649	Lets you deploy to the Controller simulator the differences between the deployed Control Participant project and the corresponding built Control Participant project. It can also be used to remove changes made online.
Re-Deploy Last Project , page 649	Lets you deploy once again the entire built Control Participant project that you had deployed to the Controller simulator last. The command is available only if you had executed the Deploy Built Project command previously on the station node.
Deploy Data , page 677	Opens the Deploy Data dialog box in which you can select a controller data backup file that has been created by using the Back Up Data command. It lets you deploy the data contained in the file to the Controller simulator running in the station node. The command available only if you have deployed a built Control Participant project already.
Start , page 707	Starts the execution of the deployed Control Participant project in the Controller simulator.
Stop , page 707	Stops the execution of the deployed Control Participant project in the Controller simulator.
Refine Online , page 707	Lets you modify the deployed Control Participant project online.
Update Project , page 707	Lets you update the logical Control Participant project that is associated to the deployed Control Participant project with the changes that you have made to the deployed project online by using the Refine Online command.
Back Up Data , page 733	Opens the Back Up Data dialog box, which lets you configure the data that you want to back up and create a controller data backup file. The command available only if you have deployed a built Control Participant project already.
Manage Password	Opens the Manage Password dialog box, which lets you set, page 661 a password for deployment and execution operations performed on a simulator instance and change it. If you have created more than one ControlExpert service, you need to select one from a list. The command is available only if the Simulator password property of the station node is set to Enabled . To use it for a given ControlExpert service, the corresponding simulator instance must be running.
Clear Password	Opens the Clear Password dialog box, which lets you remove, page 662 the application password of the deployed Control Participant project. If you have created more than one ControlExpert service, you need to select one from a list. The command is available only if you have set a password for at least one ControlExpert service. To use it for a given ControlExpert service, the corresponding simulator instance must be running.

The following table describes the commands of the **Supervision** submenu.

Command	Description
Deploy Built Project	<p>Lets you select a built Supervision Participant project and deploy it to the station node, page 698.</p> <p>The command is available only if you have created the necessary service in the station node, mapped an executable to the station node, and built it.</p>
Deploy Changes	<p>Lets you deploy a limited number of changes, page 703 to the station node.</p> <p>The command is available only if you have already deployed the built Supervision Participant project to the station node and if the number and type of changes are supported.</p>
Run Supervision Client	<p>Launches the Plant SCADA runtime locally with the corresponding Supervision project after you confirm the command.</p> <p>The command is available only if the following conditions are satisfied:</p> <ul style="list-style-type: none"> • The NIC of the station node is configured with the IP address of the local computer. • The build state of the Supervision executable that is mapped to the station node is Built or Out Of Date. <p>For the Supervision project to launch in runtime (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide), some actions need to be performed first, such as deploying the executable.</p> <p>If the runtime is already launched, you can restart it by using the command.</p> <p>The runtime is not launched if the Supervision Participant is being used by EcoStruxure Process Expert (for example, if a Supervision project is being refined or its executable built).</p>

Station Node Properties

Right-click the station node folder and select **Properties** to open the **Properties** window in which you can view and edit the station node properties.

Parameter		Description
General	Identifier	The identifier must be unique within the system and satisfy naming rules, page 88.
	Description	Optional. You can enter a description of the station node with free form text. The description you enter appears in a tooltip when you move the cursor over the entity. Default value: Blank
	Area	Optional. You can assign an access control number to the station node entity. Range: 1...255 (integer) Default value: Blank
	Computer Name	Optional. The value is applied to the <i>Computer</i> property in the Network Addresses table of the Supervision Participant project when you build its executable. The value is required, for example, when you are using encryption with Supervision. Default value: Blank
Control Expert	Password Protection	<p>The property is available only once you create a ControlExpert service, page 517.</p> <p>Lets you manage password protection, page 658 for deployment and execution operations performed on simulator instances. The setting applies to all the ControlExpert services of the station node.</p> <ul style="list-style-type: none"> • Enabled (default): Enables password protection. • Disabled: Disables password protection after you confirm the selection. You cannot disable the property when the System Access Password property, page 137 of the system is enabled. <p>By default, the value is set according to the value of the Controller Access Password at Creation property, page 137, which is configured at the system level.</p> <p>NOTE:</p> <p>To disable the property once you have set a password, you need to enter the current password, which also clears it. If several ControlExpert services exist, you need to enter the current password of each one.</p> <p>If you have already deployed a Control Participant project while the property was set to Disabled, you can enable it and set a password for the already deployed Control project.</p>
Supervision	TCP Port	<p>Configure the value to be the same as TCP Listening Port of the OPCUA server that is installed on the station node.</p> <p>The value is used in the I/O device address, page 643 only when Protocol of the associated tag container is set to OPCUA.</p> <p>Valid range: 49152 (default) to 65535.</p> <p>The parameter is available only for station nodes in which an OFS service has been created.</p>

Creating a Network Interface Card (NIC)

To create an NIC device, proceed as follows.

Step	Action
1	<p>Right-click the station node entity in the Topology Explorer and select Create NIC.</p> <p>Result: The software opens the station node folder and displays the NIC device.</p> <p>NOTE: The software may display the Select Template, page 504 dialog box.</p>

NIC Actions

Right-click the NIC device to open a context menu with the following commands.

Command	Description
Update Template , page 550	Opens the Update Template dialog box, which allows you to update the template that is used by NIC devices with the highest version that is available in the Global Templates library on the system server.
Delete	deletes the NIC device including related configurations.
Rename	allows you to change the identifier of the NIC.
Properties	opens the Properties window in which you can view and edit properties of the NIC.

NIC Properties

Right-click the NIC device and select **Properties** to open the **Device Editor**, which lets you view and edit its parameters.

Parameter	Description	
\$System	\$Name	The identifier must be unique within the system and satisfy naming rules, page 88. Default value: NIC_n where n is an incremental number starting at 1.
	\$Description	Optional. You can enter a description of the NIC with free form text. Default value: Blank
	\$Area	Optional. You can assign an access control number to the NIC. Range: 1...255 (integer) Default value: Blank
NIC Parameters	IPAddress	IP address of the NIC on the Ethernet network. The default IP address of the NIC that is created when you create the station node is 127.0.0.1, which corresponds to the local host. For additional NICs that you create manually, the default value is blank. NOTE: For station nodes on which you want to use a Controller simulator, if no system server is installed, enter the IP address, page 656 that is displayed in the simulator panel .
	Subnet Mask	Subnet mask that is used to define IP addresses on the Ethernet network that the NIC belongs to. Default value: Blank
	PhysicalInterface-LinkX ⁽¹⁾	Destination Instance Role Interface

(1) The section is displayed if you have connected, page 543 the station node through its NIC to an Ethernet network of the system. The parameters are not editable. X represents the number of the physical interface link, which is an incremental number starting at 1.

Station Node Service Actions

Right-click an **OFS**, ControlExpert, or Supervision service instance to open a context menu with the following commands.

Command	Description
Update Template , page 550	Opens the Update Template dialog box, which allows you to update the template that is used by the instance with the highest version that is available in the Global Templates library on the system server.
Delete	Deletes the instance including related configurations and mappings.
Rename	Lets you change the identifier of the instance.
Properties	Opens the Properties window in which you can view and edit properties of the service instance.

Station Node Service Properties

Right-click an **OFS**, ControlExpert, or Supervision service instance and select **Properties** to open the **Device Editor**, which lets you view and edit its parameters.

Parameter	Description	
\$System	\$Name	Identifier of the service, which must be unique within the system and satisfy naming rules, page 88.
	\$Description	Optional. You can enter a description of the instance with free form text. Default value: Blank
	\$Area	Optional. You can assign an access control number to the instance. Range: 1...255 (integer) Default value: Blank
Configuration	Port	Port number of the Controller simulator. Verify that the Controller simulator, page 654 to which you want to deploy the Control project uses the same port number. Changing the port value removes service and communication mappings that exist for the ControlExpert service. You cannot change the port value if you have deployed to this instance of the simulator. Default value: 502 This parameter is available only for ControlExpert services.

Configuring an STB Island

Overview

The topological entity represents an STB island that communicates with the controller through I/O scanning. It contains:

- Modules
- CANopen extensions
- HART devices

The STB island acts as a server.

You configure STB island entities from within the Control Participant, which then transfers the defined configuration to the STB island entity of the **Topology Explorer** by using the appropriate topological templates.

If the software cannot find the corresponding template for a hardware module, it uses the *\$UnknownModule* template. Hardware modules that are modeled with this template do not appear in the **Hardware Mapping** editor.

Creating an STB Island Entity

To create the STB island entity, which will contain the STB island configuration (devices), proceed as follows.

Step	Action
1	<p>Right-click the system root folder or a topology folder in the Topology Explorer and select Create STB Island.</p> <p>Result: The software creates an STB island entity and displays it in the tree view.</p>

STB Island Actions

Right-click the STB island entity to open a context menu with the following commands

Command	Description
Configure	Opens the Control Participant, which allows you to configure hardware, such as modules of the STB island, devices connected through the CANopen bus, and communication modules.
Physical Connections , page 545	Opens the Physical Connections window, which allows you to connect or disconnect the STB island to existing Ethernet networks to represent physical connections to such networks.
Update Template , page 550	Opens the Update Template dialog box, which allows you to update the templates that are used by STB island entities with the highest available version of the Global Templates library on the system server.
Export , page 552	Opens the Export window, which allows you to export the entity and its folder hierarchy in the form of a topology export file (.sbk).
Delete , page 526	Deletes the STB island entity including its devices and related configurations.
Rename	Allows you to change the identifier of the STB island entity.
Properties	Opens the Properties window in which you can view and edit properties of the STB island. NOTE: To view the parameters of the devices representing the hardware modules of the STB island, refer to STB Island Representation , page 525.

STB Island Properties

Right-click the STB island entity and select **Properties** to open the **Properties** window in which you can view and edit its properties.

Item		Description
General	Identifier	<p>The identifier must be unique within the system and satisfy naming rules, page 88.</p> <p>The folder identifier becomes part of the name of the items that the software creates inside in the entity:</p> <ul style="list-style-type: none"> Subfolders (identifier field) Devices (\$Name parameter) <p>The identifier (or part of it) is also used to create the name of the DTM if it is added during build, page 630.</p>
	Description	<p>Optional.</p> <p>You can enter a description of the STB island with free form text. The description you enter appears in a tooltip when you move the pointer over the entity.</p> <p>Default value: Blank</p>
	Area	<p>Optional.</p> <p>You can assign an access control number to the STB island entity.</p> <p>Range: 1...255 (integer)</p> <p>Default value: Blank</p>
Identifier <communication module ID>⁽¹⁾	IPAddress	<p>IP address of the communication module of the STB island on the Ethernet network.</p> <p>Default value: Blank</p>
	Subnet Mask	<p>Subnet mask that is used to define IP addresses on the Ethernet network that the communication module of the STB island belongs to.</p> <p>Default value: Blank</p>

(1) The section is displayed once you have closed the Participant window and saved changes.

NOTE: To view the parameters of the devices representing the hardware modules of the STB island, refer to STB Island Representation, page 525.

Changing STB Island Properties

Modifying the IP address or subnet mask of an STB island, may delete the communication mapping, page 582 that exists with this STB island.

In such case, the software displays a notification asking you to confirm the change. Make either selection:

- Yes:** To save the change to the properties of the STB island and delete the associated communication mapping.
- No:** To discard your changes.

Configuring the STB Island

To configure the STB island, proceed as follows.

Step	Action
1	<p>Right-click the STB island folder in the Topology Explorer and select Configure.</p> <p>Result: The Control Participant opens extracted, page 116 and maximized.</p>
2	<p>Proceed with the configuration of the STB island.</p> <p>For more information about configuring the STB island and working inside the Control Participant, refer to the Control Participant Services User Guide (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).</p>
3	<p>Click the close button in the toolbar of the Participant window.</p> 

Step	Action
	<p>Result: The software saves the changes in the configuration file, closes the Control Participant window, and transfers the configuration to the Topology Explorer. It may display the Select Template, page 504 dialog box.</p> <p>NOTE: If the software detects a configuration issue, it displays a notification, which allows you to change the configuration until you are able to save it.</p> <p>NOTE:</p> <p>If you click the close button on the outer frame of the Participant window, the software displays the Configuration dialog box where you can click the following:</p> <ul style="list-style-type: none"> • Yes to save the changes in the configuration file and close the Participant. • No to discard your changes and close the Participant. • Cancel to leave the Participant open without saving your changes.
4	Right-click the STB island entity and select Properties .
5	Enter the IP address and the subnet mask of the STB island.
6	Close the Properties window.

NOTE: To change the configuration of an STB island, follow the procedure to configure the STB island, and make the changes inside the Control Participant.

Configuring Communication Between Controller and STB Island

Refer to [Communicating with STB Islands Through Modbus TCP](#), page 784 for details about implementing communication by using the I/O scanner of the controller.

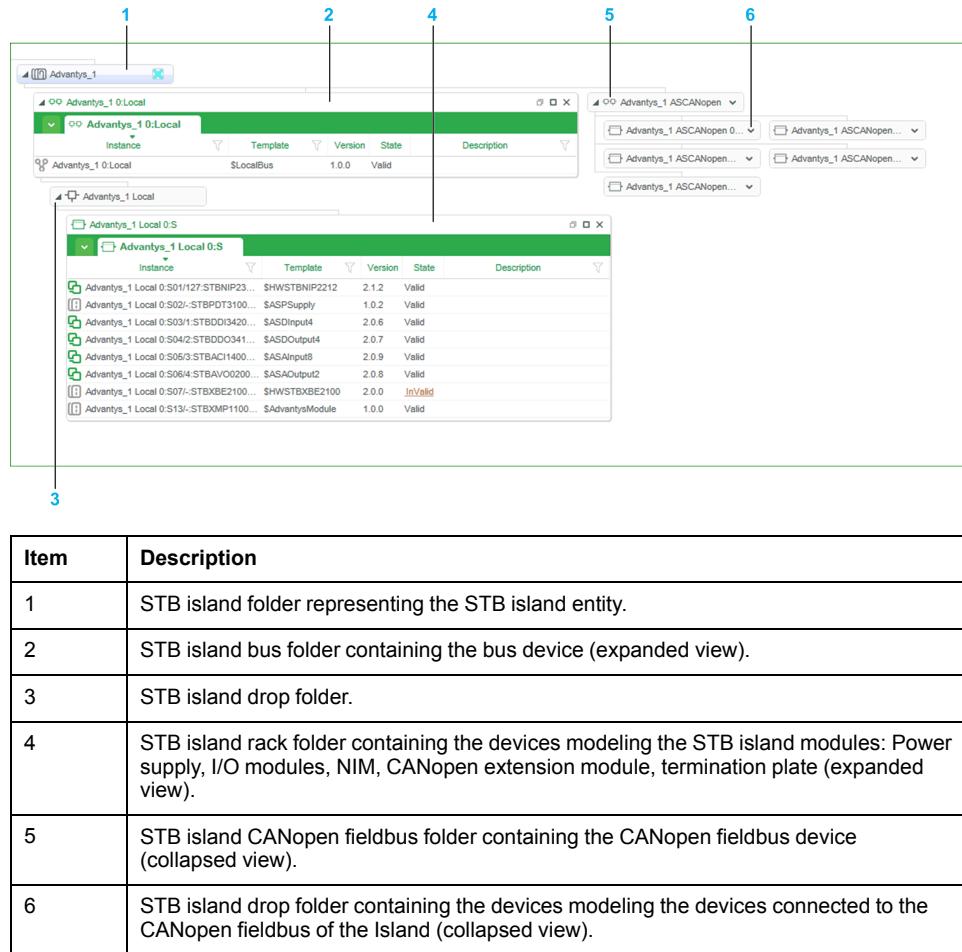
STB Island Representation

The **Topology Explorer** displays the devices representing the following hardware components of the STB island entity once you have configured it:

- Segments
- Bus
- Power supply
- CPU
- Communication modules
- I/O modules
- Devices connected through the CANopen fieldbus

It lets you also view and configure their properties, [page 514](#).

The following figure shows the representation of an STB island entity with a device connected through the CANopen fieldbus:



Deleting an STB Island Entity

When you select the **Delete** command for an STB island entity, the software notifies you if it detects any physical or mapping interface links with any device that is contained in the STB island folder that you want to delete.

Once you have confirmed the delete command, the software executes the following actions in relation to the devices that are being deleted:

- Interface links are deleted.
- The STB island folder, subfolders, and device contained therein are deleted
- The associated STB island profile is deleted.

To delete the STB island entity and its contents, proceed as follows.

Step	Action
1	<p>Right-click the STB island entity in the Topology Explorer and select Delete.</p> <p>Result: A notification dialog box opens.</p>
2	<p>Verify the information that is displayed and click OK to proceed.</p> <p>Result: The software deletes the STB island entity and its contents and displays a confirmation message in the notification panel.</p>

Configuring a PROFIBUS Remote Master and Decentralized Peripherals

Overview

The software allows you to model PROFIBUS remote masters (PRMs) and their decentralized peripherals (DPs), which communicate with supported controller platforms through I/O scanning.

You proceed with the configuration of PRM and DPs from within the Control Participant by using the default device type managers (DTMs) or generic station description (GSD) files that are installed. Thereafter, the software transfers the defined configuration to the entity of the **Topology Explorer** by using the appropriate topological templates.

If the software cannot find the corresponding topological template for a DP, it uses the **\$GenericPBS/ave** template. DPs that are modeled with this template use a generic mapping interface that is available in the **Hardware Mapping** editor, page 602.

NOTE: You can install additional DTMs and GSD files (see EcoStruxure™ Process Expert, Installation and Configuration Guide) in the Control Participant.

Creating a PRM Entity

To create the PRM entity, which will contain the PRM and DP configuration (devices), proceed as follows.

Step	Action
1	Right-click the system root folder or a topology folder in the Topology Explorer , select Create PRM Profibus DP . Result: The software creates a PRM entity and displays it in the tree view.
2	Right-click the PRM entity and select Properties . Result: The Properties window of the PRM entity opens.
3	Select the controller platform that the PRM is communicating with in the menu of the ControllerFamily property, page 528. Result: When you configure the PRM entity, page 528, the software loads the corresponding base configuration.

PRM Actions

Right-click the PRM entity to open a context menu with the following commands.

Command	Description
Configure , page 528	Opens the Control Participant, which allows you to configure the PRM and DPs by using DTMs and/or GSD files.
Physical Connections , page 542	Opens the Physical Connections window, which allows you to connect or disconnect the PRM to existing Ethernet networks to represent physical connections to such networks.
Update Template , page 550	Opens the Update Template dialog box, which allows you to update the templates that are used by PRM entities with the highest available version of the Global Templates library on the system server.
Export , page 552	Opens the Export window, which allows you to export the entity and its folder hierarchy in the form of a topology export file (.sbk).
Delete , page 530	Deletes the PRM entity including DPs and related configurations.
Rename	Allows you to change the identifier of the PRM entity.
Properties	Opens the Properties window in which you can view and edit properties of the PRM entity. NOTE: To view the parameters of the PRM devices representing the hardware modules, refer to PRM Representation , page 529.

PRM Properties

Right-click the PRM entity and select **Properties** to open the **Properties** window in which you can view and edit the properties of the PRM entity.

Item	Description	
General	Identifier	<p>The identifier must be unique within the system and satisfy naming rules, page 88.</p> <p>The entity identifier becomes part of the name of the devices that the software creates inside in the entity:</p> <ul style="list-style-type: none"> • Folders (identifier field) • Devices (\$Name parameter) <p>The identifier (or part of it) is also used to create the name of the DTM if it is added during build, page 630.</p> <p>NOTE: The identifier is not propagated to the PRM DTM (Alias name) in the Control Participant.</p>
	Description	<p>Optional.</p> <p>You can enter a description of the PRM with free form text. The description you enter appears in a tooltip when you move the pointer over the entity.</p> <p>Default value: Blank</p>
	Area	<p>Optional.</p> <p>You can assign an access control number to the PRM entity.</p> <p>Range: 1...255 (integer)</p> <p>Default value: Blank</p>
Controller Family	ControllerFamily	<p>Indicates the hardware platform of the controller that the PRM communicates with through I/O scanning.</p> <p>The value needs to correspond to the family of the controller that you select in the service mapping tab, page 580 of the executable of the Control Participant project.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • <i>M340/M580</i> • <i>Quantum</i>

NOTE: To view the parameters of the PRM devices representing the hardware modules, refer to PRM Representation, page 529.

Configuring the PRM and its DPs

To configure the PRM and its DPs, proceed as follows.

Step	Action
1	<p>Right-click the PRM entity in the Topology Explorer and select Configure.</p> <p>Result: The Control Participant opens extracted, page 116 and maximized.</p>
2	<p>Proceed with the configuration of the PRM and its DPs inside the Control Participant.</p> <p>For more information, refer to the topic describing the configuration of the PRM and DPs (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).</p>

Step	Action
3	<p>Click the save button in the toolbar of the Participant window.</p>  <p>Result: The software saves the changes in the configuration file and transfers the configuration to the Topology Explorer. It may display the Select Template, page 504 dialog box.</p> <p>NOTE: If the software detects a configuration issue, it displays a notification, which allows you to change the configuration until you are able to save it.</p>
4	<p>Close the Control Participant window.</p> <p>NOTE:</p> <p>If you change the configuration and you click the close button on the outer frame of the Participant window without saving your changes first, the Save Configuration dialog box opens where you can click the following:</p> <ul style="list-style-type: none"> • Yes to save the changes in the configuration file and close the Participant window. • No to discard your changes and close the Participant window. • Cancel to leave the Participant open without saving your changes.

Changing the PRM Configuration

To change the configuration of a PRM, follow the procedure to configure a PRM, and proceed with the changes within the Control Participant.

Saving changes to communication functions that you made within the Control Participant, may delete existing communication mappings, page 582 of the PRM.

In such case, a dialog box opens asking you to confirm the change. Make either selection:

- **OK:** To save the change to the configuration file of the PRM and delete the associated communication mapping.
- **Cancel:** To keep the Control Participant window open. Unsaved changes are not discarded.

For more information, refer to Configuring the PRM, page 528.

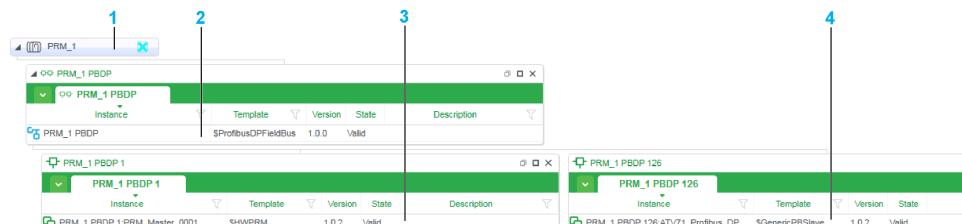
PRM and DP Representation

The **Topology Explorer** allows you to view the topological devices representing the following components of the PRM entity once you have configured it:

- PROFIBUS fieldbus
- PRM master
- PROFIBUS DPs

It lets you also view and configure their properties, page 514.

The following figure shows the representation of a PRM master configuration with one DP communicating with a Quantum controller.



Item	Description
1	PRM folder representing the PRM entity
2	PROFIBUS bus device
3	PRM device
4	DP device

Deleting a PRM Entity

Once you confirm the delete command, the software deletes the following components of the PRM entity:

- Physical interface links
- The PRM entity, subfolders, and devices contained therein
- The corresponding Control Participant configuration project.

To delete the PRM entity and its contents, proceed as follows.

Step	Action
1	Right-click the PRM entity in the Topology Explorer and select Delete . Result: A notification dialog box opens.
2	Verify the information that is displayed and click OK to proceed. Result: The software deletes the PRM entity and its contents, and displays a confirmation message in the notification panel.

Configuring Modbus TCP and EtherNet/IP Devices

Overview

The **Topology Explorer** lets you select and instantiate the topological template of devices that communicate with the controller by using either:

- Modbus TCP implicit (I/O scanning) or explicit messaging.
- EtherNet/IP explicit messaging.

Each template creates an I/O device topological entity.

There are two types of I/O device templates:

- Templates that are specific to a Schneider Electric device (see EcoStruxure™ Process Expert, General Purpose Library Device Templates Reference Manual) (for example, *\$EATV32HW*), which models an Altivar 32 speed drive).
- Generic templates (for example, *\$EIPGenericDeviceHW*), which let you model devices for which no specific template is available. For generic Modbus TCP templates, select one that creates the required number of I/O scanner lines.

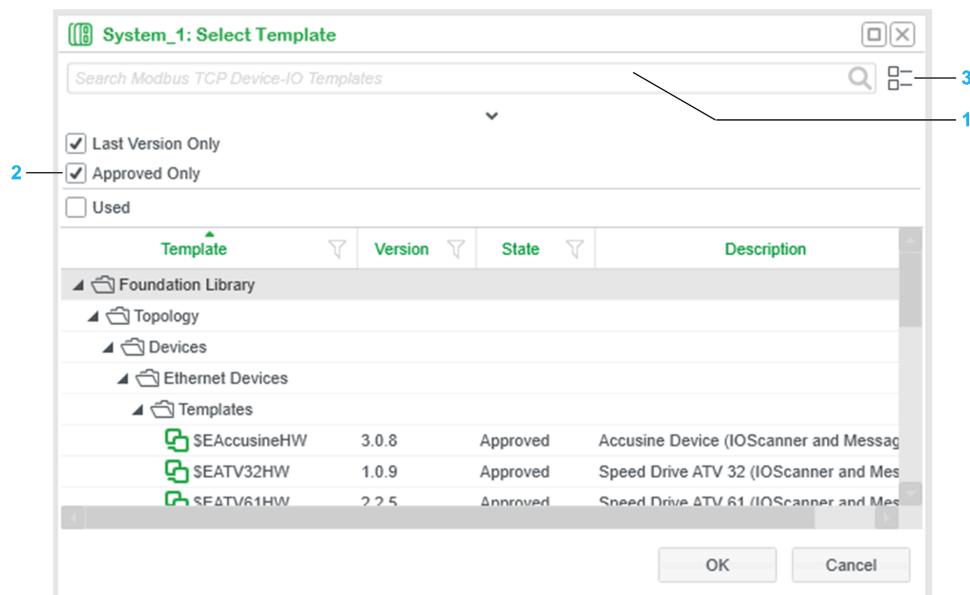
Alternatively, instances of the *\$EGenericDeviceHW* template can act as gateways when you implement monitoring of Modbus serial device data directly from the Supervision infrastructure, page 830.

Selecting Topological I/O Device Templates

The **Select Template** dialog box displays templates of the Global Templates library that meet the following criteria:

- **Type: Topological**
- **SubType: Device-I/O**
- **Usability State** (see EcoStruxure™ Process Expert, Global Templates, Reference Manual): **Approved** or **Deprecated**

The following figure shows an example of the **Select Template** dialog box for Modbus TCP I/O device templates in tree view mode.



Item	Description
1	Search field, page 159 to search for topological I/O device templates.
2	Filters, page 160. The check mark indicates that the filter is applied.
3	Button to toggle between grid view and tree view.

Column header	Description
Template	Identifier of the template. In tree view mode, the path to the template in the Global Templates library is also shown.
Version	Version of the template.
State	Usability state (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of the template.
Description	Description of the template.

Creating an I/O Device Entity

To create an I/O device entity, proceed as follows.

Step	Action
1	In the Topology Explorer , right-click the root or a topological folder and select Create Device IO > Modbus TCP or EtherNet/IP . Result: The Select Template dialog box opens displaying the corresponding templates.
2	Select a template and click OK . Result: The software creates an instance of the selected template in the folder, which represents the I/O device topological entity. NOTE: Click Cancel to close the Select Template dialog box without creating the I/O device entity.

Modbus TCP and EtherNet/IP Device Actions

Right-click the I/O device entity to open a context menu with the following commands.

Command	Description
Physical Connections , page 544	Opens the Physical Connections window, which allows you to connect or disconnect the device to existing Ethernet networks to represent physical connections to such networks.
Update Template , page 550	Opens the Update Template dialog box, which allows you to update the template that is used by Modbus TCP device entities with the highest available version of the Global Templates library on the system server.
Replace Template , page 212	Opens the Replace Device Template dialog box, which allows you to replace the template that is used by the device with another template that meets the criteria, page 531 and that is available in the Global Templates library on the system server.
Delete , page 526	Deletes the device entity including related configurations and mappings.
Rename	Allows you to change the identifier of the Modbus TCP device entity.
Export , page 552	Opens the Export window, which allows you to export the entity and its folder hierarchy in the form of a topology export file (.sbk).
Properties	Opens the Device Editor window in which you can view and edit properties of the device entity.

I/O Device Properties

Right-click the device entity and select **Properties** to open the **Device Editor** window in which you can view and edit its parameters.

You can view and/or edit the following parameters.

Parameter		Description
\$System	\$Name	<p>The identifier must be unique within the system and satisfy naming rules, page 88.</p> <p>The default value: xHW_n where x is the identifier of the corresponding application template of the device and n an incremental number starting at 1.</p> <p>The identifier (or part of it) is also used to create the name of the DTM if it is added during build, page 630.</p> <p>String.</p>
	\$Description	<p>Optional.</p> <p>You can enter a description of the device with free form text (string).</p>
	\$Area	<p>Optional.</p> <p>You can assign an access control number to the device entity.</p> <p>Range: 1...255 (integer)</p>
Communication	IPAddress	<p>IP address of the device on the Ethernet network.</p> <p>Default value: Blank</p>
	Subnet Mask	<p>Subnet mask that is used to define IP addresses on the Ethernet network that the device belongs to.</p> <p>Default value: Blank</p>
	IOScanningLines ⁽¹⁾	<p>You can select the number of I/O scanner lines that are generated during build.</p> <p>Possible values: 1 to <i>n</i>, where <i>n</i>, is the maximum number of lines that a given template can generate.</p> <p>Default value: Varies by template.</p>
	ID ⁽²⁾	<p>Modbus unit ID, unique identifier of the device. The parameter is provided by the <i>DeviceID</i> element of the Modbus TCP server interface (<i>MBTCPServer</i>), which is referenced by all Schneider Electric Modbus TCP I/O device hardware templates.</p> <p>Default value: 255</p> <p>Valid range: 0 to 255</p> <p>(UnsignedInt).</p> <p>For example, for Altivar variable speed drives, the identifier is 255.</p> <p>NOTE: You can configure the parameter also in the Communication Mapping tab, page 625 of Supervision executables.</p>

Parameter	Description
Communication	ReadLength⁽²⁾ Number of words to be read (integer). Enter an appropriate value. The value cannot be null when you implement communication by using I/O scanning with the CPU of an M580 controller or a NOC communication module. Default value: 0
	ReadAddress⁽²⁾ Remote address (%MW) to be read (integer). Enter an appropriate value. Default value: 0
	WriteLength⁽²⁾ Number of words to be written (integer). Enter an appropriate value. The value cannot be null when you implement communication by using I/O scanning with the CPU of an M580 controller or a NOC communication module. Default value: 0
	WriteAddress⁽²⁾ Remote address (%MW) to be written (integer). Enter an appropriate value. Default value: 0
	Description⁽²⁾ Optional. Enter a description of the device with free form text (string).
Configuration	Privilege Parameter for access control purposes. Default value: Blank (String).
	IOSType⁽³⁾ Parameter to select the I/O scanning element to be used: <ul style="list-style-type: none">• <i>TesysTE</i>: To read/write the complete data from/to the device.• <i>TesysTEFast</i>: To read/write a minimum set of data from/to the device. Default value. The selection and the device template that is used in the application of the system needs to match; otherwise you cannot proceed with the hardware mapping, page 602. For example, if you select <i>TesysTEFast</i> for this parameter, instantiate the \$TesysTEFast template in the application. (Enum). NOTE: Changing this parameter value when the hardware mapping is done removes the mapping.
	EnableDHCP When true, the device subscribes to a DHCP server service.
	IdentifyByDevice-Name When true, the DHCP server identifies the device by using DTMDeviceName ; otherwise, it identifies the device by using MACAddress .
	MACAddress MAC address of the device. Use a hyphen as separator.
PhysicalInterfaceLink1⁽⁴⁾	DTMDeviceName Name of the device. For example, TeSys0 64.
	Destination Instance Indicates to which Ethernet network the device is connected.
	Role Indicates the identifier of the role of the interface.
	Interface Indicates the identifier of the interface that the device exposes for connection to the Ethernet network.

- (1) The parameter is displayed only for entities using a generic topological Modbus TCP device template, which allows selecting the number of I/O scanner lines that are generated (for example, `$EGenDev4IOSHW`).
- (2) The parameters are available only for entities using a generic topological Modbus TCP device template (and also for a few Modbus TCP device-specific templates). For generic templates that reference several Modbus TCP server interfaces (for example, `$EGenDev4IOSHW`), the set of parameters appears as many times as there are server interfaces in the template.
- (3) The parameter is available only for certain Modbus TCP entities (for example, those using the `$ETesysTHW` template).
- (4) The section is displayed if you have connected, page 544 the I/O device entity to an Ethernet network of the system. The parameters are not editable.

Changing I/O Device Properties

Modifying the IP address or subnet mask of an I/O device entity, may delete the communication mapping, page 582 that exists with this device.

In such case, the software displays a notification asking you to confirm the change.

Make either selection:

- **Yes:** To save the change to the properties of the I/O device and delete the associated communication mapping.
- **No:** To discard your changes.

Implementing Communication Between Controllers and I/O Devices

For details about how to implement communication between controllers and I/O devices by using Modbus TCP or EtherNet/IP, refer to the topic describing how to manage communication, page 779.

Configuring HART Modules

Overview

The HART communication protocol can be addressed by using either of these Modicon eX80 modules:

- BMEAHI0812 HART analog input module.
- BMEAHO0412 HART analog output module.

You can add these HART analog I/O modules to local racks and/or remote drops of controllers of the Quantum and M580 platform. With the Quantum platform, a NOC communication module is required.

You proceed with the configuration of HART modules from within the Control Participant by using the **Configure** command on a controller entity.

Thereafter, the software transfers the defined configuration to the entity of the **Topology Explorer** by using the appropriate topological templates.

NOTE: Addressing the HART protocol by using a Modicon STB island is also possible with the STBAHI8321 HART interface module, page 523.

Managed Data

The **Primary Variable** (PV) of HART analog I/O modules is the only variable that is available in the **Hardware Mapping Editor** later on. During this stage, you can map facets of analog I/O templates (for example, `$AnalogInput1`) instantiated in the application, which represent PV, to the projection of the I/Os of the HART modules.

The other variables and data of the HART modules that you configure are preserved during build and can be deployed to a controller but are not usable at the system level.

NOTE: You need to use an Asset Management System and perform some additional steps (see EcoStruxure™ Process Expert , Foundation Application Templates, User Guide) manually to take advantage of the advanced functionality provided by HART analog I/O modules.

HART Module Device DDT Variables

When you add a HART analog I/O module to a configuration, the Participant creates a device DDT (DDDT) variable associated to the I/O the module.

When you add the corresponding HART device type manager (DTM) that is preinstalled, it creates a second DDDT variable associated to the DTM to manage the extended data.

Both variable are present only in the configuration that you can access from the **Topology Explorer** and are integrated to the Control Participant project during build, page 630. However, if you want to use the two DDDTs in a Control Participant project during refinement, you can create them, page 445.

Configuring HART Modules

To configure HART analog I/O modules, proceed as follows.

Step	Action
1	Right-click the controller entity in the Topology Explorer and select Configure . Result: The software displays the Control Participant in a reduced window.
2	Maximize the Control Participant.
3	Proceed with the configuration of the HART module inside the Control Participant.

Step	Action
	<p>For more information about configuring HART analog I/O module and working inside the Control Participant, refer to the topic describing the configuration of the controller (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).</p>
4	<p>Click the save button in the toolbar of the Participant window.</p>  <p>Result: The software saves the changes in the configuration file and transfers the configuration to the Topology Explorer. It may display the Select Template, page 504 dialog box.</p> <p>NOTE: If the software detects a configuration issue, it displays a notification, which allows you to change the configuration until you are able to save it.</p>
5	<p>Close the Control Participant window.</p> <p>NOTE:</p> <p>If you change the configuration and you click the close button on the outer frame of the Participant window without saving your changes first, the software displays the Save Configuration dialog box where you can click the following:</p> <ul style="list-style-type: none"> • Yes to save the changes in the configuration file and close the Participant window. • No to discard your changes and close the Participant window. • Cancel to leave the Participant open without saving your changes.

Changing the HART Module Configuration

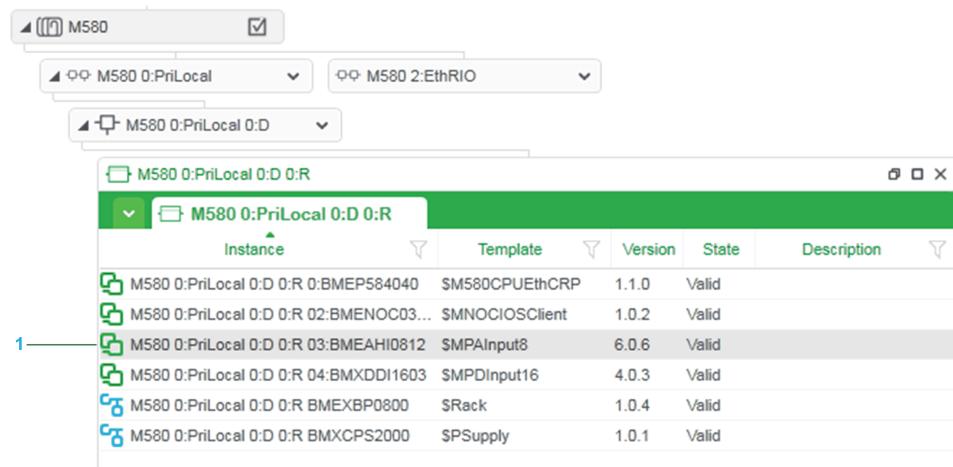
To change the configuration of a HART analog I/O module, follow the procedure to configure a HART module, and proceed with the changes inside the Control Participant.

HART Module Representation

The **Topology Explorer** allows you to view the topological devices representing HART analog I/O modules.

It lets you also view and configure their properties, page 514.

The following figure shows the representation of a BMEAHI0812 HART analog I/O module located in the local rack of an M580 controller entity.



1 Device representing the HART module.

NOTE: The HART module is represented similarly in a topological entity based on the Quantum platform.

Configuring an Ethernet Network

Overview

The Ethernet network entity represents a physical or virtual Ethernet local area network (LAN or VLAN) to which you can physically connect topological entities of the system such as controllers, Modbus TCP devices, EtherNet/IP devices, and station nodes.

You create Ethernet network entities and define their parameters in the **Topology Explorer**.

Creating an Ethernet Network Entity

To create an Ethernet network entity, proceed as follows.

Step	Action
1	<p>Right-click the system root folder or a topological folder in the Topology Explorer and select Create Ethernet Network.</p> <p>Result: The software creates an Ethernet network entity and displays it inside the folder.</p> <p>NOTE: The software may display the Select Template, page 504 dialog box.</p>

Ethernet Network Actions

Right-click the Ethernet network entity to open a context menu with the following commands.

Command	Description
Update Template , page 550	Opens the Update Template dialog box, which allows you to update the template that is used by the Ethernet network entity with the highest available version of the Global Templates library on the system server.
Export , page 552	Opens the Export window, which allows you to export the entity and its folder hierarchy in the form of a topology export file (.sbk).
Delete	Deletes the entity including related configurations.
Rename	Allows you to change the identifier of the Ethernet network entity.
Properties	Opens the Device Editor window in which you can view and edit properties of the Ethernet network.

Ethernet Network Properties

Right-click the Ethernet network entity and select **Properties** to open the **Device Editor** window in which you can view and edit parameters of the Ethernet network.

You can view and/or edit the following parameters.

Parameter	Description	
\$System	\$Name	<p>The identifier must be unique within the system and satisfy naming rules, page 88.</p> <p>Default value: EthernetNetwork_n where <i>n</i> is an incremental number starting at 1.</p>
	\$Description	<p>Optional.</p> <p>You can enter a description of the Ethernet network with free form text.</p> <p>Default value: Blank</p>
	\$Area	Optional.

Parameter	Description
	<p>You can assign an access control number to the Ethernet network entity.</p> <p>Range: 1...255 (integer)</p> <p>Default value: Blank</p>
PhysicalInterfaceLinkX⁽¹⁾	<p>Indicates which topological entities and/or devices you have connected to the Ethernet network.</p> <p>A section containing these parameters is displayed for each of the connected topological entities and devices, including connections to standby controllers of redundant configurations.</p>
Destination Instance	Indicates the \$Name of the connected entity or device, page 512.
Role	Indicates the identifier of the role of the interface.
Interface	Indicates the identifier of the interface that the Ethernet network entity exposes for connecting with the topological entity or device.

(1) The section is displayed if you have connected, page 540 topological entities or devices to the Ethernet network. The parameters are not editable. X represents the number of the physical interface link, which is an incremental number starting at 1.

Connecting to an Ethernet Network

Overview

The **Physical Connections** command allows you to connect topological entities that support Modbus TCP or EtherNet/IP connectivity to an Ethernet network entity of the system.

The physical interface link that is created represents the physical connection of communication modules to a network.

At the platform level, physical connections for entities are required to be able to perform the communication mapping, page 576 in Control and Supervision executables. They are not required to perform service mapping or deployment and execution tasks.

Devices with which you can create a physical connection are identified by using their \$Name parameter, page 512.

To make a physical connection, create an Ethernet network first, page 538.

NOTE: The command also allows you to disconnect an entity from an Ethernet network.

IP Address Uniqueness Requirement

The IP address of a topological entity and/or its communication module on the Ethernet network must be unique to make the physical connection possible. This rule applies to:

- CPU modules (including main IP address and IP address A depending on controller platform)
- Communication modules such as:
 - NOE
 - NOC (for 140NOC78•00 modules, main IP address only, not IP address A)
 - CRA
 - NIC of station nodes
 - NIM of STB islands
 - Embedded RIO scanner services, page 512 of M580 CPUs
- Modbus TCP and EtherNet/IP I/O devices
- PROFIBUS remote masters (PRMs)

NOTE: Refer also to rules for redundant controller entities, page 541.

Viewing Connections to Ethernet Networks

The **PhysicalInterfaceLink** parameter allows you to view, which entities are connected to which Ethernet network. Access the parameter by right-clicking either of the following and selecting **Properties**:

- The Ethernet network entity, page 538.
- The communication module of a topological entity

NOTE: You can also right-click an entity and select **Physical Connections** to view the Ethernet network to which the entity is connected to.

Availability of Ethernet Networks

If the communication or hardware mapping editor is open for an entity, then the Ethernet network to which this entity is connected is not available. Because of this, you cannot use the **Physical Connections** command to connect other entities to this Ethernet network or change existing connections to this network.

Close the communication mapping editor to release the Ethernet network.

Managing Conflicting Connections to Ethernet Networks

If the system contains at least 2 entities, each with a communication module physically connected to an Ethernet network, and during the configuration of 1 of the entities, you enter a conflicting IP address for the module, a dialog box opens when you attempt to save the configuration, requiring confirmation to remove the physical connection of the module.

Click either of the following:

- **OK** to remove the physical connection of the module, disconnecting it from the Ethernet network.
- **Cancel** to cancel the save operation and keep the Control Participant open, which allows you to change the configuration.

For more information, refer to Considerations When Updating an Existing Configuration (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

NOTE: Removing the physical connection to an Ethernet network deletes all or part of the existing communication and hardware mapping of the associated Control Participant project executable.

Connections with Redundant Controllers

For redundant controllers, the rule about uniqueness of the IP address on an Ethernet network to make the connection possible applies also to:

- For CPU modules:
 - (Main) IP address +1 (where IP address is the address of the CPU module of the primary controller)
 - IP address B (M580 platform)
- For BMENOC3•1 communication modules, main IP address +1 only (not IP address B)
- For 140NOC78•00 communication modules, IP address +1 (where IP address is the address of the NOC module of the primary controller)
- For NOE communication modules, IP address +1 (where IP address is the address of the NOE module of the primary controller)

Therefore, when a standalone controller is connected to an Ethernet network and you change the configuration to a redundant one, already connected modules of the controller that has become the primary controller may be disconnected. This affects existing mappings, page 511.

The table describes the various scenarios when you change the configuration of a standalone controller to a redundant one depending on the status of the IP address of counterpart modules in the standby controller. It is assumed that the standalone controller becomes the primary controller of the redundant configuration.

Status of physical connection of module in standalone configuration	IP address ⁽¹⁾ of counterpart module in the standby controller after changing to redundant configuration	Impact on the physical connection of both modules to the Ethernet network
Connected	Unique	Both modules are connected.
Connected	Already exists	Module of the primary controller is disconnected. Module of the standby controller is not connected.
Not connected	irrelevant	Both modules are not connected.

(1) If the module has two IP addresses both are considered unless otherwise mentioned.

For example, if the local rack of an M580 standalone controller contains an NOC module, which is connected to an Ethernet network (main IP address) and the

CPU communication module is also connected (Main IP address and IP address A) if, when you change this configuration to a redundant one, the IP address of the NOC module of the standby controller (main IP address +1) is not unique on the same network, then it is not connected to the network and the NOC module of the primary controller is disconnected from the network. If the IP addresses of the CPU communication module of the standby controller are unique, both primary and standby CPU communication modules are connected to the network.

The other way around, when you change the configuration of a redundant controller to a standalone one, the modules that were connected to the Ethernet network remain connected.

Connecting a Controller to an Ethernet Network

You can connect controller entities that offer physical connectivity to existing Ethernet networks of the system.

To connect a controller to an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the controller entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the identifier of the communication modules of the controller in the Communication Modules column.</p> <p>NOTE:</p> <ul style="list-style-type: none"> • The identifier only appears if you have assigned an IP address to the communication module in the configuration. • When connecting a redundant controller, only communication modules of the primary controller are displayed in the Physical Connections window. • When connecting a controller that has a CRP module or an embedded RIO scanner service (M580 CPUs), the EthRIO bus device that is displayed in the Physical Connections window represents the physical connection with the remote I/O network. It connects the corresponding communication modules such as CRP and CRA modules, and the embedded RIO scanner service.
2	<p>Click the field in the Allowed Network(s) column next to the communication module that you want to connect to the network.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	Select a network.
4	<p>Click OK to save your selection and close the Physical Connections window.</p> <p>Result: The communication module is connected to the selected Ethernet network.</p> <p>NOTE:</p> <ul style="list-style-type: none"> • For redundant controllers, the software automatically connects the counterpart module of the standby controller to the same Ethernet network. • For BMENOC3•1 communication modules, IP addresses A and B are not connected.

Disconnecting a Controller from an Ethernet Network

To disconnect a controller from an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the controller entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the identifier of:</p> <ul style="list-style-type: none"> The communication modules of the controller in the Communication Modules column The Ethernet network to which the controller is connected in the Allowed Network (s) column <p>NOTE:</p> <ul style="list-style-type: none"> When disconnecting a redundant controller, only the communication modules of the primary controller are displayed in the Physical Connections window. When disconnecting a controller that is CRP module or an embedded RIO scanner service (M580 CPUs), the EthRIO bus device that is displayed in the Physical Connections window, represents the physical connection with the remote I/O network. It disconnects the corresponding communication modules, such as CRP and CRA modules, and the embedded RIO scanner service. <p>The other way around, when the software disconnects, page 541, for example, a CRA module because of an IP address conflict, it is the EthRIO bus device that is disconnected.</p>
2	<p>Click the field in the Allowed Network(s) column next to the communication module that you want to disconnect from the network.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	<p>Select Not Assigned in the Allowed Network(s) menu.</p> <p>NOTE: You may also select another Ethernet network from the menu (if available) to create a connection with it instead.</p>
4	<p>Click OK.</p> <p>Result: If a communication mapping exists for the executable to which the controller is mapped, the software displays a notification, asking you to confirm the disconnection.</p> <p>Otherwise, the software disconnects the communication module from the selected Ethernet network, and closes the Physical Connections window.</p> <p>NOTE: For redundant controllers, the software automatically disconnects the counterpart module of the standby controller from the same Ethernet network.</p>
5	<p>Click Yes to confirm.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> Disconnects the communication module from the selected Ethernet network. Deletes the existing mappings, page 511 of the executable to which the controller is mapped. Changes the status of the executable to which the controller is mapped to Out Of Date if it is built. Closes the Physical Connections window. <p>NOTE: Click No to close the Physical Connections window without disconnecting the communication module from the Ethernet network.</p> <p>NOTE: For a redundant controller, the software automatically disconnects the counterpart module of the standby controller from the same Ethernet network.</p>

Connecting a Station Node to an Ethernet Network

You can connect or disconnect station node entities to existing Ethernet networks through their NIC devices.

To connect a station node to an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the station node entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the identifier of the NIC of the station node in the Communication Modules column.</p> <p>NOTE: The NIC identifier only appears if you have assigned an IP address to the NIC.</p>
2	<p>Click the field in the Allowed Network(s) column next to the NIC that you want to connect to the network.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	Select a network.
4	<p>Click OK to save your selection and close the Physical Connections window.</p> <p>Result: The station node is connected to the selected Ethernet network.</p>

Disconnecting a Station Node from an Ethernet Network

To disconnect a station node entity from an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the station node entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the identifier of:</p> <ul style="list-style-type: none"> The NIC of the station node in the Communication Modules column The Ethernet network to which the NIC is connected in the Allowed Network(s) column
2	<p>Click the field in the Allowed Network(s) column next to the NIC that you want to disconnect from the network.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	<p>Select Not Assigned in the Allowed Network(s) menu.</p> <p>NOTE: You may also select another Ethernet network from the menu (if available) to create a connection with it instead.</p>
4	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> Disconnects the station node from the selected Ethernet network. Deletes the existing communication mapping, page 582 of the executable to which the station node is mapped. Changes the status of the executable to which the station node is mapped to Out Of Date if it is built. Closes the Physical Connections window.

Connecting a Modbus TCP or EtherNet/IP Device to an Ethernet Network

You can connect or disconnect Modbus TCP and EtherNet/IP device entities to existing Ethernet networks.

To connect an I/O device entity to an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the device entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the device entity identifier in the Communication Modules column.</p> <p>NOTE: The device identifier only appears if you have assigned an IP address to the device.</p>
2	Click the field in the Allowed Network(s) column next to the device that you want to connect to the network.

Step	Action
	Result: The Ethernet networks to which a connection is possible are displayed.
3	Select a network.
4	Click OK to save your selection and close the Physical Connections window. Result: The device is connected to the selected Ethernet network.

Disconnecting a Modbus TCP or EtherNet/IP Device from an Ethernet Network

To disconnect a Modbus TCP or EtherNet/IP device entity from an Ethernet network, proceed as follows.

Step	Action
1	Right-click the device entity in the Topology Explorer and select Physical Connections . Result: The software opens the Physical Connections window and displays the identifier of: <ul style="list-style-type: none">• The I/O device in the Communication Modules column• The Ethernet network to which the device is connected in the Allowed Network(s) column
2	Click the field in the Allowed Network(s) column next to the device that you want to disconnect from the network. Result: The Ethernet networks to which a connection is possible are displayed.
3	Select Not Assigned in the Allowed Network(s) menu. NOTE: You may also select another Ethernet network from the menu (if available) to create a connection with it instead.
4	Click OK . Result: If a hardware or communication mapping exists for the executable of the client controller communicating with the device, the software displays a notification, asking you to confirm the disconnection. Otherwise, the software disconnects the I/O device from the selected Ethernet network, and closes the Physical Connections window.
5	Click Yes to confirm. Result: The software: <ul style="list-style-type: none">• Disconnects the I/O device from the selected Ethernet network.• Deletes the existing hardware mapping, page 602 and communication mapping, page 582 of the executable of the client controller communicating with the device.• Changes the status of the client controller communicating with the device to Out Of Date if it is built.• Closes the Physical Connections window. NOTE: Click No to close the Physical Connections window without disconnecting the I/O device from the Ethernet network.

Connecting an STB Island to an Ethernet Network

You can connect or disconnect STB islands to existing Ethernet networks through their NIM.

To connect an STB island to an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the STB island entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the identifier of the communication module of the STB island in the Communication Modules column.</p> <p>NOTE: The module identifier only appears if you have assigned an IP address to the NIM.</p>
2	<p>Click the field in the Allowed Network(s) column next to the communication module that you want to connect to the network.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	Select a network.
4	<p>Click OK to save your selection and close the Physical Connections window.</p> <p>Result: The STB island entity is connected to the selected Ethernet network.</p>

Disconnecting an STB Island from an Ethernet Network

To disconnect an STB island entity from an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the STB island entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the identifier of:</p> <ul style="list-style-type: none"> • The NIM of the STB island in the Communication Modules column • The Ethernet network to which the NIM is connected in the Allowed Network(s) column
2	<p>Click the field in the Allowed Network(s) column next to the NIM that you want to disconnect from the network.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	<p>Select Not Assigned in the Allowed Network(s) menu.</p> <p>NOTE: You may also select another Ethernet network from the menu (if available) to create a connection with it instead.</p>
4	<p>Click OK.</p> <p>Result: If a hardware or communication mapping exists for the executable of the client controller communicating with the STB island, the software displays a notification, asking you to confirm the disconnection.</p> <p>Otherwise, the software disconnects the STB island entity from the selected Ethernet network, and closes the Physical Connections window.</p>
5	<p>Click Yes to confirm.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> • Disconnects the STB island entity from the selected Ethernet network. • Deletes the existing hardware mapping, page 602 and communication mapping, page 582 of the executable of the client controller communicating with the STB island. • Changes the status of the client controller communicating with the STB island to Out Of Date if it is built. • Closes the Physical Connections window. <p>NOTE: Click No to close the Physical Connections window without disconnecting the communication module from the Ethernet network.</p>

Connecting a PRM to an Ethernet Network

You can connect or disconnect PRM entities to existing Ethernet networks.

To connect a PRM entity to an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the PRM entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the PRM entity identifier in the Communication Modules column.</p> <p>NOTE: The PRM identifier only appears if you have assigned an IP address, page 528 to the PRM.</p>
2	<p>Click the field in the Allowed Network(s) column next to the PRM.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	Select a network.
4	<p>Click OK to save your selection and close the Physical Connections window.</p> <p>Result: The PRM is connected to the selected Ethernet network.</p>

Disconnecting a PRM from an Ethernet Network

To disconnect a PRM entity from an Ethernet network, proceed as follows.

Step	Action
1	<p>Right-click the PRM entity in the Topology Explorer and select Physical Connections.</p> <p>Result: The software opens the Physical Connections window and displays the identifier of:</p> <ul style="list-style-type: none"> The PRM in the Communication Modules column The Ethernet network to which the PRM is connected in the Allowed Network(s) column
2	<p>Click the field in the Allowed Network(s) column next to the PRM.</p> <p>Result: The Ethernet networks to which a connection is possible are displayed.</p>
3	<p>Select Not Assigned in the Allowed Network(s) menu.</p> <p>NOTE: You may also select another Ethernet network from the menu (if available) to create a connection with it instead.</p>
4	<p>Click OK.</p> <p>Result: If a hardware or communication mapping exists for the executable of the client controller communicating with the PRM, the software displays a notification, asking you to confirm the disconnection.</p> <p>Otherwise, the software disconnects the PRM from the selected Ethernet network, and closes the Physical Connections window.</p>
5	<p>Click Yes to confirm.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> Disconnects the PRM entity from the selected Ethernet network. Deletes the existing hardware mapping, page 602 and communication mapping, page 582 of the executable of the client controller communicating with the PRM. Changes the status of the client controller communicating with the PRM to Out Of Date if it is built. Closes the Physical Connections window. <p>NOTE: Click No to close the Physical Connections window without disconnecting the PRM from the Ethernet network.</p>

Managing the Topology

Overview

This section describes how to use the software to manage the topology of the system.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Moving Topological Folders and Entities

Moving Topological Folders

You can move any topological folder to another folder, including the system root folder, by dragging it.

Updating Templates of Topological Entities

Overview

By using the **Update Template** command in the **Topology Explorer**, you can update the templates that are used by the following topological entities and their devices:

- Controllers
- STB islands
- Station nodes
- Modbus TCP and EtherNet/IP devices
- PROFIBUS Remote Masters and DPs
- Ethernet networks

If the update of one template cannot be completed for an entity, the entire update process is rolled back.

When used at the folder level, the command applies to any entity that is contained in the folder, and in any of its subfolders.

NOTE: The update functionality affects:

- Existing service, communication, and hardware mapping interface links.
- Deployment information related to the executable to which the entity is mapped.

NOTE: No command is available to replace a topological template, such as the *\$UnknownModule* template (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

Accessing the Command

Right-click a folder or an entity, and select **Update Template**.

Description of the Update Functionality

The software proceeds with the update of templates in the following order:

- The templates that are associated to the entity are updated.
- Interface links are processed.

The table describes the effects on the different template items when you update the templates that are used by a topological entity:

Item	Identifier in New Template	Action
Parameter values	Is same as in current template	Explicitly defined parameters of the current template keep their respective value in the new template.
	Is not in current template	New parameters that are not part of the current template keep their default values.
Physical interface links, page 540	Is same as in current template	Physical interface links of the entity are maintained.
	Is not the same as in current template	New physical interface links can be added. Physical interface links that are not part of the new template or that do not satisfy applicable rules are removed.
Mappings, page 577	–	The software maintains existing service, communication, and hardware mapping interface links of the Control and Supervision executables to which the entity is mapped if the corresponding mapping interfaces are available, and satisfy applicable rules.
Deployment information	–	If you had deployed a Control project to a station node or controller entity, the corresponding deployment information is unchanged.

NOTE: When you update the template of a redundant controller entity, the update takes places for both the primary and the standby controller entities.

Default Template Selection

By default, the software selects the topological template with the same identifier that has the highest version number with the **Usability State** of **Approved**.

Updating Templates

To update the templates that are used by a topological entity, proceed as follows:

Step	Action
1	<p>In the tree view of the Topology Explorer, right-click the entity and select Update Template.</p> <p>Result: The Update Device Template dialog box opens. It shows for each device template that can be updated, the new version that will be used.</p>
2	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none">Updates the templates that are used by the entity.May delete existing mapping information of the corresponding executable of the Control project. <p>NOTE: Click Cancel to close the Update Device Template dialog box without updating templates.</p>

NOTE: If you use a filter in the **Name** column of the **Update Device Template** dialog box and as a result, only some of the device templates for which a later version exists are displayed, the update operation is still performed on the entire selection of device templates of the entity for which a later version exists.

Exporting Topological Entities

Overview

The **Export > Topology** command allows you to create a non-editable export file (.sbk), which contains information of selected topological entities, their folder hierarchy, and dependencies.

You can use this data with the import, page 555 functionality of the **Topology Explorer** to do the following:

- Transfer project information to a different system or EcoStruxure Process Expert infrastructure.
- Add and/or modify topological entities.
- Duplicate topological entities.
- Move, remove, replace, and/or add devices of entities.
- Update properties of entities and folders.
- Prepare the topology of a system before importing Control and/or Supervision Participant projects.

NOTE: To export only I/O devices, page 564 to a comma-separated values (CSV) file, use the **Export > Devices** command.

Exporting M580 Safety Controllers

For specific information about exporting M580 safety controllers, refer to the topic describing how to engineer systems with M580 safety controllers, page 772.

Software Version Compatibility

The version of EcoStruxure Process Expert that you use to import the information contained in the export file must be the same as the version that you used to create the export file (a service pack is considered the same version); otherwise you cannot proceed with the import.

Export File Contents

You can select the **Export > Topology** command at different levels to export the following:

- For an entity: Exports the data pertaining to this topological entity and its devices. The parent folder hierarchy up to the system root folder is also included.
- For a topological folder: Exports the entities contained in the folder and in any subfolders, and their devices. The parent folder hierarchy up to the system root folder is also included.
- For the system root folder: Exports the entities of the entire system and their devices. The parent folder hierarchy up to the system root folder for any entity is also included.

By default, in each case, the folder hierarchy of the exported entities is selected for export.

You cannot export empty folders or folders that contain entities that are not selected for export. The rule does not apply for folders that are part of the parent folder hierarchy of an exported entity or folder.

NOTE: If physical connections exist for a selected entity at the time of export, these links are automatically included in the export file. This allows recreating the physical connection at the time of import if applicable rules are satisfied, page 558.

NOTE: Passwords, page 658 are not exported. Yet, the password protection properties, page 509 and their value are exported.

Topological Templates

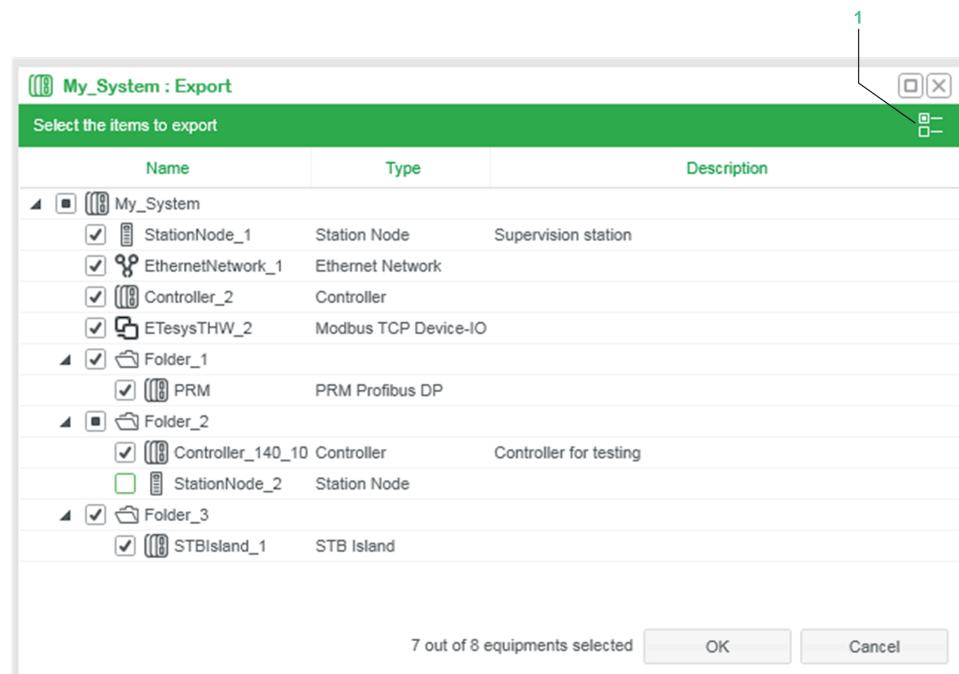
The export file does not contain the topological templates of the entities and devices that you are exporting. To import entities in a different EcoStruxure Process Expert infrastructure, the Global Templates library of the target architecture needs to contain the necessary templates with the same version. Entities for which the template is not present cannot be imported.

For information on how to export templates, refer to the topic that describes managing Global Templates, page 881.

Topological Export Window

The topological **Export** window allows you to select the entities and folders that you export.

The following figure shows an example of the **Export** window in tree view mode.



1 Button to toggle between the grid view and tree view

Header	Description
Name	Shows the identifier of topological entities that you can export. By default, entities are selected for export. NOTE: When check boxes of folders are filled with a black square, at least one entity or folder in a sublevel is not selected for export.
Type	Type of the topological entity (controller, station node, Ethernet network, and so on).
Path	Path to the folder or entity. The column is displayed only in grid view.
Description	Contents of the Description property of the entity.

NOTE: To expand/collapse a node in the **Export** window, select the node and press **Enter**.

Aborting Export Tasks

After confirming the export by clicking **Save** in the **Save** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The export file is not created.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are exporting only a few entities, the task may complete before you are able to click the icon.

Creating the Export File

To create a topology export file, proceed as follows.

Step	Action
1	In the Topology Explorer , right-click the folder or entity that you want to export and select Export . Result: The software opens the Export window and displays the entities that you can export.
2	Select the entities that you want to export.
3	Click OK . Result: The software opens the Save dialog box. NOTE: Click Cancel to close the Export window without creating the export file.
4	Select a location where you want to save the export file, enter a file name, and click Save . Result: The software creates the export file (.sbk), which contains information of the selected topological entities. NOTE: Click Cancel to close the Save dialog box and revert to the Export window.

Importing Topological Entities

Overview

The import functionality allows you to import data of topological folders and entities from a topological export file (.sbk), which was created by using the export, page 267 functionality of the **Topology Explorer**.

You can import this data into an existing system by using the **Import > Topology** command of the **Topology Explorer**.

You have the possibility to select which items you want to import out of those contained in the export file.

Before proceeding with the import, the software verifies if importing the selected instances creates a duplicate entity identifier or an IP address conflict. If required, the import operation includes a second step, which allows you to resolve conflicts by selecting an action to perform during import:

- Skip the entity; it is not imported.
- Update the entity of the target system with the configuration of the export file.
- Create the entity with a different, unique identifier.

The notification panel displays information on both the tasks that completed successfully and those that could not be completed.

Applicable import rules are described in this topic. Examples are given to illustrate various scenarios, page 562.

NOTE: You cannot proceed with the import of an entity if in the target system one of its dependencies is locked by another operation or user. For example, if in the target system the **Hardware Mapping Editor** is open, you cannot import into the topology of this system entities that are linked to the Ethernet network because it is locked by the hardware mapping process. However, you could import a station node that has no physical connection.

Importing M580 Safety Controllers

For specific information about importing M580 safety controllers, refer to the topic describing how to engineer systems with M580 safety controllers, page 772.

Software Version Compatibility

The version of EcoStruxure Process Expert that you use to import the information contained in the export file must be the same as the version that you used to create the export file (a service pack is considered the same version); otherwise you cannot proceed with the import.

Topological Templates

The export file does not contain the topological templates of the exported entities and devices. If you are importing entities in a different EcoStruxure Process Expert infrastructure and the required template for an entity is not present, you cannot import it.

Information on the required templates and template version is indicated in the **Import** window.

Identification of Conflicts

The table indicates the meaning of the color coding that the software uses to indicate the status of entities and folders in the **Import** window, page 559 in the context of the target system.

Identification method	Tooltip text	Description
Tooltip	OK	An entity of the same type with the same name does not exist in the topology of the target system. If you select it for import, the entity is created.
Tooltip	Entity already exists	An entity of the same type with the same name, and with an identical configuration already exists in the topology of the target system. If you select it for import, the entity is created but with a different name to comply with the naming rule of the topology. This rule specifies that identifiers of entities must be unique within a system.
Tooltip	Conflict to be resolved	An entity of the same type with the same name already exists in the topology of the target system but the software detected a change, which can, for example, be related to: <ul style="list-style-type: none"> • Properties. • Devices of entities: Properties, position of modules on the rack, configuration, CPU or module type. • Services of station nodes. • Physical connections. If you want to import the entity anyway, you need to select an action for it. NOTE: Refer to the topic describing import rules for conflicting entities, page 557.
IP with yellow dot	IP conflict	At the time of export, the entity that you want to import was connected to an Ethernet network that exists in the target system and its IP address is already used on this network, page 540. In addition, the tooltip text indicates <conflicting IP address in the target system> - <path to the entity in the target system> - <Ethernet network identifier in the target system>. You can still import the entity but physical connections are impacted. For details, refer to the topic describing import rules for physical connections, page 558. NOTE: If the entity that has the same IP address in the target system also has the same identifier and is of the same type, the software does not indicate an IP address conflict.
T with red dot	Template conflict. Unable to find the target template. <template identifier – version>	The entity that you want to import uses a template that is not present in the Global Templates library. You must import the template before you can import the entity. For information on how to import templates, refer to the topic that describes managing Global Templates, page 881.

General Import Rule for Non-Conflicting Entities and Folders

Each entity that is contained in the export file and that you have selected for import is created in the topology of the target system if the software does not detect a conflict.

Depending on the import path setting that you select, the software may create folders as needed to recreate the original hierarchy.

If a folder with the same identifier already exists in the target system, its properties (for example, its description or area) are automatically updated with the configuration of the export file.

Import Rules for Conflicting Entities

The table describes the actions that you can perform for various types of entities when the software detects a conflict, depending if you select **Update** or **Create**. To be able to perform the changes that are indicated in the table, the export file needs to contain an entity that has the same identifier as an entity of same type in the target system. The imported entity needs to have a configuration that is different from the target entity. Update actions apply to the entity in the target system.

Entity type	Create action	Update action	Comments
Station node	Creates the entity with a new identifier.	<ul style="list-style-type: none"> Services are updated. Properties of the entity and its services are updated. Entity is moved if path in export file differs. 	Removing a service from a station node may affect existing service mappings of related executables.
Controller	Creates the entity with a new identifier.	<ul style="list-style-type: none"> Properties of the entity and its devices are updated. Position of modules are updated. Modules are added, replaced, and/or removed. Entity is moved if path in export file differs. <p>NOTE: Action not allowed if the controller is not of the same hardware platform, page 509.</p>	Moving, adding, replacing, or removing a module of a controller may affect existing service, hardware, and/or communication mappings of the related executables. This includes: <ul style="list-style-type: none"> Replacing the CPU module of a controller. Changing a standalone controller into a redundant one or the other way around.
STB island	Creates the entity with a new identifier.	<ul style="list-style-type: none"> Properties of the entity and its devices are updated. Position of modules are updated. Modules are added and/or removed. Entity is moved if path in export file differs. 	You can overwrite parameters such as: <ul style="list-style-type: none"> IP address. Subnet mask. Explicit parameters that are defined in the corresponding templates.
Modbus TCP and EtherNet/IP I/O device		<ul style="list-style-type: none"> Properties are updated. Entity is moved if path in export file differs. 	Updating an I/O device may affect existing hardware and/or communication mappings of the related executables.
Ethernet network	Creates the entity with a new identifier.	<ul style="list-style-type: none"> Properties are updated. New physical connections are recreated if the entity is also imported and no IP conflict is detected. Entity is moved if path in export file differs. 	Existing physical connections in the target system are not maintained.

NOTE: Refer also to the topic describing import rules for physical connections, page 558.

Import Rules for Physical Connections

When you import an entity for which a physical connection, page 540 to an Ethernet network existed at the time of export and the target system contains no Ethernet network with the same identifier, the software recreates the physical connection in the target system if the following conditions are true:

- The Ethernet network is included in the export file.
- Both the entity and the Ethernet network are imported successfully with their original identifier (if an entity is created with a new identifier, the physical connection is not recreated).

On the opposite, if an Ethernet network with the same identifier already exists in the target system, the Ethernet network does not need to be contained in the export file and you do not need to select it if it is contained. The physical connection to this Ethernet network is automatically maintained if the IP address of the imported entity is still available on the Ethernet network of the target system.

NOTE: As a rule, physical connections that exist in the target system prevail.

When you import an entity that is connected to an Ethernet network, if the Ethernet network in the target system has the same name and the IP address of the entity is already used, page 556 on this Ethernet network, after import, the entity is not connected to the Ethernet network anymore. The disconnection from the Ethernet network happens whether you create an entity or update an existing one. The IP address of the entity after import is the one contained in the export file.

Disconnecting an entity from the Ethernet network may affect existing hardware and/or communication mappings of the related executables.

NOTE: More information on physical connections of redundant controllers is available in the topic describing connections to Ethernet networks, page 541.

Password Management After Importing Topological Entities

After you import into a system a controller and/or a station node for which password protection, page 509 has been enabled, you need to do the following to be able to deploy a Control Participant project:

- If no password was set, you need to set one.
- If a password was set, you need to enter the password when prompted.

For more information, refer to the topic describing how to manage passwords for Control Participant projects, page 658.

NOTE: The above applies even if you are importing these topological entities into the same system from which they were exported or if you have not logged out of the engineering client since the export was performed.

Opening the Topology Import Window

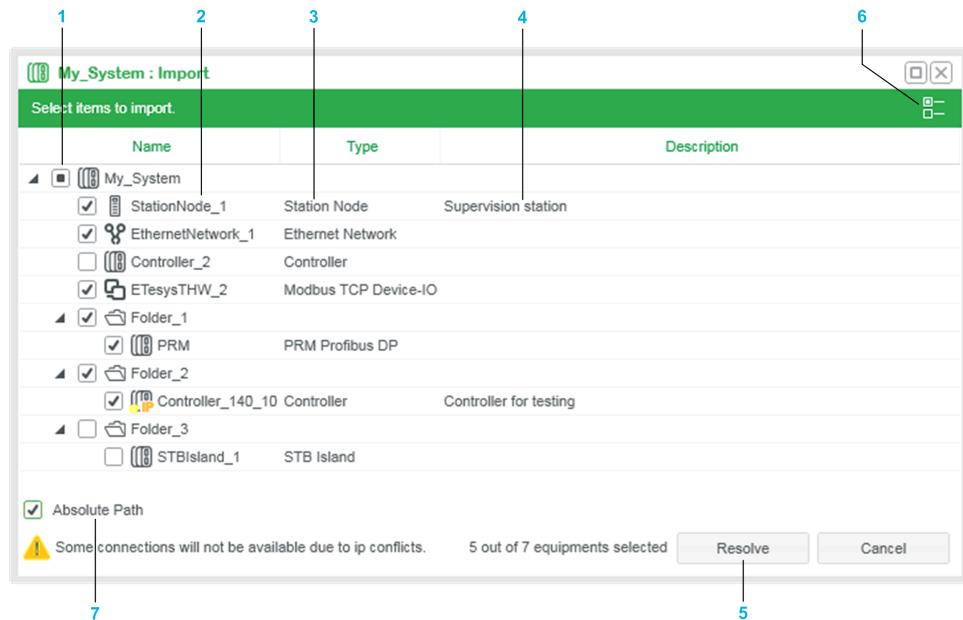
To open the topology **Import** window, proceed as follows.

Step	Action
1	From the Systems Explorer , right-click the system into which you want to import topological entities and select Open Topology . Result: The Topology Explorer opens.
2	Right-click the system root folder and select Import . Result: The Import dialog box opens.
3	Browse to the topology export file (.sbk) that contains information on the topological entities that you want to import and click Open . Result: The Import window opens and displays the contents of the file that you have selected.

Topology Import Window Description

The topology **Import** window allows you to select the entities that you want to import in the topology of a system and to take appropriate actions in case of detected conflicts, page 556.

The following figure shows an example of the **Import** window displaying the contents of the selected topological export file.



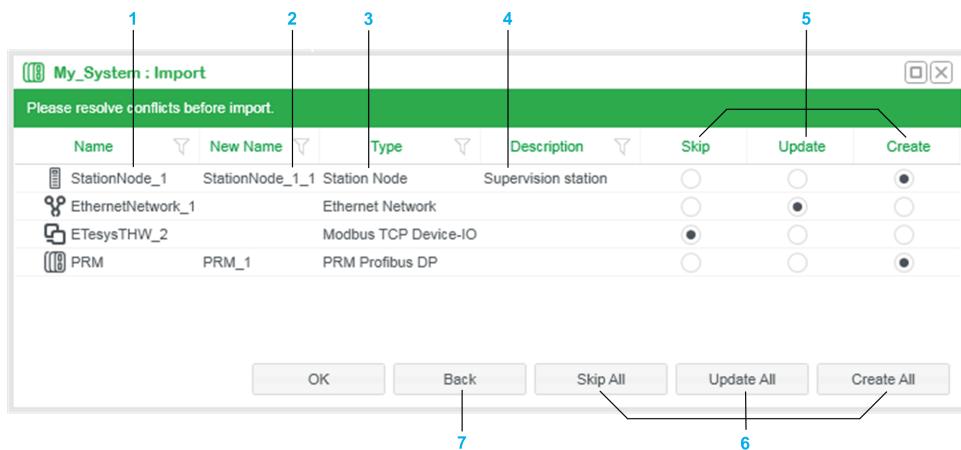
Item	Description
1	<p>The check boxes allow you to select the folders and/or entities that you want to import. Selecting a folder selects any item it contains and the other way around. By default, items contained in the export file are selected.</p> <p>NOTE: A check box filled with a black square indicates that at least one item in a sublevel is not selected.</p>
2	<p>Identifier of the folders and entities contained in the export file (Name column). Entities for which the software has detected a conflict are identified, page 556. In tree view mode, the column also reflects the hierarchy of the items at the time of export. The hierarchy may be different depending on the import path setting.</p>
3	Type of the entity.
-	<p>Path column, which indicates the path to the folder or entity. The column is displayed only in grid view. Refer to item (6).</p>
4	Description that comes from the properties of the entity or folder contained in the export file.
5	<p>OK or Resolve button.</p> <p>Either button is displayed depending on the status of the instances selected for import in the context of the target system:</p> <ul style="list-style-type: none"> • OK: The software did not detect duplicate identifier conflicts and selected instances are created in the target system upon clicking the button. • Resolve: The software detected at least one duplicate identifier conflict for an instance selected for import. Upon clicking the button, a new import window opens, page 560, which displays only entities for which a conflict was detected. It allows you to resolve identifier conflicts. <p>The other entities for which no conflict was detected and that you have selected remain selected for import.</p> <p>NOTE: If Resolve is displayed and you clear the check box of instances for which a conflict has been detected, the button changes to OK, and the other way around.</p>

Item	Description
6	Button to toggle between grid view and tree view.
7	Import path setting check box: <ul style="list-style-type: none"> Relative Path: Default setting. The folder in which the import is performed becomes the parent folder of the imported folder structure. The hierarchy of the imported folder structure remains unchanged. Absolute Path: When selected, the software maintains the hierarchy that existed at the time of export. The software creates folders as needed to recreate the original hierarchy. <p>NOTE: The check box is not displayed when you import in the system root folder because in such case, relative and absolute paths are the same.</p>

NOTE: To expand/collapse a node in the **Import** window, select the node and press **Enter**.

Resolving Detected Conflicts

The following figure shows an example of the **Import** window that is displayed when the software has detected conflicts and you click **Resolve** in the first import window.



The table describes the second topology **Import** window, which opens when you click **Resolve**.

Item	Description
1	Identifier of entities that you have selected for import and for which the software has detected an identifier and/or IP address conflict (Name column). The type of conflict is identified, page 556. NOTE: Only grid view mode is available in this window.
2	New name that will be used to create the entity when you select the Create action for the entity (New Name column). The software automatically changes the identifier of the entity by adding the <i>_n</i> suffix where <i>n</i> is an incremental number starting at 1. This is for not having two entities with the same identifier in the topology after import. For example, if the original identifier of the instance is <i>Controller_1</i> , the new identifier is <i>Controller_1_1</i> .
3	Type of the entity.
4	Description that comes from the properties of the entity or folder contained in the export file.
5	Buttons that allow you to select an action for individual instances: <ul style="list-style-type: none"> Skip: The instance is not imported. Update: Default action. The instance that has the same identifier in the target system is updated, page 557 with the configuration contained in the export file. Depending on the type of entity, this update can pertain to descriptions, services, IP addresses, I/O modules, and so on. Create: The instance is created in the target system with the new identifier indicated in the New Name column.

Item	Description
6	Buttons that allow you to select an action for the conflicting instances as a whole. The actions are the same as when you select the action for instances individually. NOTE: Clicking on one of these buttons overrides the selection you have made for individual instances.
7	Back. Click to revert to the first import window. NOTE: If you modify the default selection in this window and click Back to revert to the first import window, when you click Resolve again in the first import window, the changes you had made to the selection are retained. Instances that you may have unselected in the first import window do not appear anymore, and the other way around.

Aborting Import Tasks

After confirming the import by clicking **OK** in the **Import** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. No entities are imported.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are importing only a few entities, the task may complete before you are able to click the icon.

Importing Topological Entities

To import topological entities into an existing system, proceed as follows.

Step	Action
1	In the Import window, page 558, select the entities that you want to import.
2	If you are importing in a folder other than the system root folder, select Absolute Path if needed, page 559.
3	If the software has detected an identifier conflict, click Resolve , otherwise proceed to step 5. Result: A new import window opens, which displays entities for which the software has detected a conflict. The other entities for which no conflict was detected and that you have selected remain selected for import. NOTE: Click Cancel to close the Import window without importing entities.
4	Select the action that you want to perform for each conflicting entity. NOTE: Click Back to revert to the first import window. Your selection is retained when you return to this window by clicking Resolve again.
5	Click OK . Result: The software: <ul style="list-style-type: none">• Applies the import rules based on your selection and updates the topology of the target system.• Displays an import summary in the notification panel.

Examples

This topic gives examples of various import scenarios of an entity into a system, which contains an Ethernet network with the same name as the network in the source system.

The table gives an example of an imported entity that creates an identifier conflict. The **Update** action is selected in the second step of the import.

Element	What is in the target system	What is Imported	Result
Entity identifier	StationNode_1	StationNode_1	StationNode_1
Services	ControlExpert	Supervision	Supervision
Properties	Description: Station1	Description: Station2	Description: Station2
Folder properties	None	Description: Control Room	Description: Control Room

The table gives two examples of an imported entity that does not create an identifier conflict. In the second example, the software detects an IP address conflict. In both cases, you can import Entity_2 in one step.

Element	What is in the target system	What is Imported	Result	
Entity identifier	Entity_1	Entity_2	Entity_1	Entity_2
IP address	IP_1	IP 1	IP 1	IP 1
Physical connection to network_1	Not connected	Connected at time of export	Not connected	Connected
<hr/>				
Entity identifier	Entity_1	Entity_2	Entity_1	Entity_2
IP address	IP 1	IP 1	IP 1	IP 1
Physical connection to network_1	Connected	Connected at time of export	Connected	Not connected

The table gives two examples of an imported entity that creates an identifier conflict but without an IP address conflict. The action that is selected in the second step of the import is indicated.

Element	What is in the target system	What is Imported	Result
Update action is selected			
Entity identifier	Entity_1	Entity_1	Entity_1
IP address	IP 1	IP 1 or any IP	IP address of export file
Physical connection to network_1	Connected	Not connected at time of export	Not connected
Entity identifier	Entity_1	Entity_1	Entity_1
IP address	IP 1 (not used on network_1)	IP_1 or any IP address not used on network_1 of target system	IP address of export file
Physical connection to network_1	Not connected	Connected at time of export	Connected
Create action is selected			
Entity identifier	Entity_1	Entity_1	Entity_1
IP address	IP 1	Any IP address not used on network_1 of target system	IP 1
Physical connection to network_1	Connected	Connected at time of export	Connected
			Not connected

The table gives two examples of an imported entity that creates an IP address conflict. In each example, a different action is selected in the second step of the import.

Element	What is in the target system	What is Imported	Result
Update action is selected			
Entity identifier	Entity_1	Entity_1	Entity_1
IP address	IP 1	IP address already used on network_1 of target system	IP address of export file
Physical connection to network_1	Connected	Connected at time of export	Not connected
Create action is selected			
Entity identifier	Entity_1	Entity_1	Entity_1
IP address	IP 1	IP 1	IP 1
Physical connection to network_1	Connected	Connected at time of export	Connected
			Not connected

Exporting and Importing Devices to/from CSV Format

Overview

The device export and import functionality of the **Topology Explorer** lets you create comma-separated value (CSV) files containing device data and import them into the topology of the same or another system.

This allows you to do the following:

- Move or copy the I/O devices, page 531, Ethernet networks, page 538, and physical connections, page 540 that exist in the topology, or a selection of these, to another system.
- Perform bulk processes for creating and/or modifying devices, Ethernet networks, and physical connections by editing the export file.

This topic describes how to export objects, edit the export file to perform the required changes, and import it.

NOTE: The functionality does not apply to other topological entities, such as controllers and station nodes. To export these, use the **Topology** command, page 552 in the **Export** context menu.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Exporting Devices to CSV Format

Overview

You can export I/O devices by using the **Export > Devices** command of the **Topology Explorer**.

The device entities and their respective configuration parameters are saved to an export file in comma-separated, page 568 values (CSV).

For example, you can use the export file to import, page 571 the devices into another system or to modify some of their parameters, such as IP addresses.

NOTE: You also need to export the topological templates, page 886 that are used by the device entities that you are exporting if the same version of template is not present in the Global Templates library of the target system at the time of import. Devices whose template is not present cannot be imported.

Good Practices

If you want to create devices by importing a CSV file, use the **Export > Devices** command from the topology of any system even if it does not contain devices. This creates an editable, page 567 CSV file, which contains the columns for the data that you need to enter.

Invariant Culture

The software uses the invariant culture, page 568 to format parameter values independently of the language setting that is used on the computer on which you create the export file.

For example, a decimal value that appears as 5,5 (comma separator) in the **Device Editor** because of your regional settings appears as 5.5 (dot separator) in the export file.

Object Selection

When you use the **Export > Devices** command, the following objects are exported:

- Only the devices that exist in the topology without the folder structure.
- Physical connections of the devices.

NOTE: If a device is locked when you select the command (for example, because its properties are being edited), it is still exported but with the configuration that was saved last.

Aborting Export Tasks

After confirming the export by clicking **Save** in the **Export** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The export file is not created.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are exporting only a few devices, the task may complete before you are able to click the icon.

Creating the Device Export File

To export the devices of the topology, proceed as follows.

Step	Action
1	In the Topology Explorer , right-click the system root folder and select Export Devices . Result: The Export dialog box opens.
2	Select the location and file name for the export file.
3	Click Save . Result: <ul style="list-style-type: none">• The export file (.CSV) is created.• A summary of the exported objects is displayed in the notification pane.

Editing the CSV Device Export File

Overview

By editing the device export file (.csv), you can create and/or update devices, page 531 and their physical connections in the topology of a system.

Various rules and requirements apply when you open and edit CSV export files. They are described in this topic.

The software applies the changes when you import the edited CSV file, page 571 into the topology of the system. At that time, you have the possibility to review and modify the list of imported devices.

Good Practices

- Before editing a device export file, create a copy of it. If needed, it lets you restore the original device topology.
- Edit only an up-to-date export file to help avoid data conflicts when importing.
- Carefully remove from the export file devices that are not required, such as devices that you are not creating or updating.

Opening the CSV File by Using a Spreadsheet Editor

When you open a CSV export file by using a spreadsheet editor, if a device parameter value is of type *Duration* with millisecond values, contains special characters or strings starting with - (hyphen)), it may get modified by the editor.

So that such values remain unchanged, you first need to import the data into a blank workbook in *Unicode (UTF-8)* format.

Follow this procedure. Microsoft® Excel® is used as an example. Menu items and dialog box names may vary depending on the version of the application.

Step	Action																																																														
1	Open a blank workbook.																																																														
2	In the Data menu, select to import from text or CSV.																																																														
3	In the Import Text File dialog box, select the exported device CSV file.																																																														
4	Select the following attribute values for the import of data: <ul style="list-style-type: none"> • File origin: <i>65001 : Unicode (UTF-8)</i> • List separator (delimiter): <i>Comma</i> • Format for all columns (Data Type): <i>Text⁽¹⁾</i> 																																																														
5	Load the values to the blank workbook. <p>NOTE: After loading is completed, the first row of the device CSV file must start with <i>Action</i> (first cell).</p> <p>The following example shows an extract of a device CSV export file after loading values to the blank workbook.</p> <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> <th>G</th> <th>H</th> </tr> <tr> <th>1</th> <th>Action</th> <th>DeviceName</th> <th>Description</th> <th>Template</th> <th>IPAddress</th> <th>SubnetMask</th> <th>ID</th> <th>EthernetNetwork</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Create</td> <td>ATV1</td> <td>My ATV1</td> <td>\$ETesysTHW</td> <td>192.168.10.20</td> <td>255.255.255.0</td> <td></td> <td>EthernetNetwork_1</td> </tr> <tr> <td>3</td> <td>Create</td> <td>ATV2</td> <td>My ATV2</td> <td>\$ETesysTHW</td> <td>192.168.10.21</td> <td>255.255.255.0</td> <td></td> <td>EthernetNetwork_1</td> </tr> <tr> <td>4</td> <td>Create</td> <td>ATV3</td> <td>My ATV3</td> <td>\$ETesysTHW</td> <td>192.168.10.22</td> <td>255.255.255.0</td> <td></td> <td>EthernetNetwork_1</td> </tr> <tr> <td>5</td> <td>Create</td> <td>ATV4</td> <td>My ATV4</td> <td>\$EATV320HW</td> <td>192.168.10.23</td> <td>255.255.255.0</td> <td></td> <td>EthernetNetwork_1</td> </tr> <tr> <td>6</td> <td>Create</td> <td>ATV5</td> <td>My ATV5</td> <td>\$EATV320HW</td> <td>192.168.10.24</td> <td>255.255.255.0</td> <td></td> <td>EthernetNetwork_1</td> </tr> </tbody> </table>	A	B	C	D	E	F	G	H	1	Action	DeviceName	Description	Template	IPAddress	SubnetMask	ID	EthernetNetwork	2	Create	ATV1	My ATV1	\$ETesysTHW	192.168.10.20	255.255.255.0		EthernetNetwork_1	3	Create	ATV2	My ATV2	\$ETesysTHW	192.168.10.21	255.255.255.0		EthernetNetwork_1	4	Create	ATV3	My ATV3	\$ETesysTHW	192.168.10.22	255.255.255.0		EthernetNetwork_1	5	Create	ATV4	My ATV4	\$EATV320HW	192.168.10.23	255.255.255.0		EthernetNetwork_1	6	Create	ATV5	My ATV5	\$EATV320HW	192.168.10.24	255.255.255.0		EthernetNetwork_1
A	B	C	D	E	F	G	H																																																								
1	Action	DeviceName	Description	Template	IPAddress	SubnetMask	ID	EthernetNetwork																																																							
2	Create	ATV1	My ATV1	\$ETesysTHW	192.168.10.20	255.255.255.0		EthernetNetwork_1																																																							
3	Create	ATV2	My ATV2	\$ETesysTHW	192.168.10.21	255.255.255.0		EthernetNetwork_1																																																							
4	Create	ATV3	My ATV3	\$ETesysTHW	192.168.10.22	255.255.255.0		EthernetNetwork_1																																																							
5	Create	ATV4	My ATV4	\$EATV320HW	192.168.10.23	255.255.255.0		EthernetNetwork_1																																																							
6	Create	ATV5	My ATV5	\$EATV320HW	192.168.10.24	255.255.255.0		EthernetNetwork_1																																																							
6	Save the workbook in CSV <i>UTF-8 (comma delimited)</i> format (keep saving in this format when you edit the file).																																																														
(1) You may need to edit the file during the import process to access the attribute (<i>Transform Data</i>).																																																															

Setting the List Separator Format

CSV export files use the comma as list separator (delimiter) independently of the setting that is in effect on the computer on which the file is created.

If the format of the list separator on the computer is different from comma, data will not be displayed correctly when you open the export file by using a spreadsheet editor.

To verify the setting and change it to comma if necessary, open the Windows® Control Panel and click **Clock and Region > Region**. In the **Formats** tab, click **Additional settings....**

NOTE: For information on how to use commas in values of string data types in the CSV file, refer to the topic describing editing rules, page 570.

Using the Invariant Culture

Use the invariant culture to modify or add parameter values in CSV export files.

If required, during import the software converts the format according to the language setting that is used on the target computer.

The table describes the formats of the invariant culture to be used in a CSV export file.

Type	Format	Example
Decimal separator	x.y	2.5
Time	<i>hh:mm:ss</i>	14:30:00
Date	<i>mm/dd/yyyy</i>	11/20/2014
Date Time	<i>mm/dd/yyyy hh:mm:ss</i>	11/20/2014 14:30:00
Duration	<i>DD.hh:mm:ss.ms</i>	10.12:30:10.15 translating into 10 days 12 hours 30 min 10 sec 15 msec

NOTE: Parameters of the types described in the table may not necessarily appear in device CSV export files.

Device Export File Content Requirements

This topic describes the columns that appear in the device CSV export file, the requirements, and the impact for each one. Use this information when you edit existing data and/or add columns. The conflicts that can be detected, page 573 are described in the topic documenting the import of devices.

Header	Description	Requirement and impact
<i>Action</i>	Action that is performed on the device during import.	A column with this header and a valid value in each row are required. Possible values (not case-sensitive): <ul style="list-style-type: none"> • <i>Create</i> (default) • <i>Update</i>
<i>DeviceName</i>	Device name, page 88 (<i>\$Name</i>). The value is used to identify the device or calculated by the software in case of device creation.	A column with this header and a unique, valid value in each row are required. To update a device, the value must correspond to the name of a device that exists in the topology of the target system. NOTE: If the column contains several entries with the same value, only one device will be created at the most. It is the one in the topmost row.
<i>Description</i>	String that appears in the description of the device (<i>\$Description</i>).	A column with this header is required. The value is optional. If a row contains data, it must be valid.
<i>Template</i>	Topological template, page 504 that is used by the device.	A column with this header and a valid value in each row are required. The topological template must exist in the Global Templates library of the target system. NOTE: The software automatically uses the latest template version that is available independently of the version that the device was using at the time of export.
<i>IPAddress</i>	IPv4 address of the device.	A column with this header is required. The value is optional. If a row contains data, it must be valid. The value must be in the proper Pv4 format. NOTE: To create a physical connection for a device, you need to enter an IP address that is available on the Ethernet network that is configured for the device.
<i>SubnetMask</i>	Subnet mask that is used to define IP addresses on the Ethernet network that the device belongs to.	A column with this header is required. The value is optional. If a row contains data, it must be valid.
<i>ID</i>	<i>Modbus</i> Unit ID of the device.	A column with this header is required. The value is optional. If a row contains data, it must be valid.
<i>EthernetNetwork</i>	Name of the Ethernet network to which the device is connected by a physical connection.	A column with this header is required. The value is optional. Possible values: <ul style="list-style-type: none"> • Name of an existing Ethernet network in the topology of the target system: A physical connection is created between this network and the device. • Name of a new Ethernet network: The network is created in the topology of the target system and a physical connection is created between this network and the device. • Blank: No physical connection is created. In case of a device update, an existing physical link in the target topology is removed. NOTE: To create a physical connection for a device, it must have an IP address that is available on the Ethernet network.
<i>Privilege</i>	Runtime privilege required to access the runtime service.	A column with this header is required. The value is optional. If a row contains data, it must be valid.
<i>IOSType</i>	Parameter to select the I/O scanning element to be used. For <i>TeSys T</i> devices only.	A column with this header is required. The value is optional. If a row contains data, it must be valid. Possible values: <ul style="list-style-type: none"> • <i>TesysTEFast</i> • <i>TesysTE</i>

Header	Description	Requirement and impact
<i>EnableDHCP</i>	Indicates whether the device subscribes to a DHCP server service.	A column with this header is required. The value is optional. If a row contains data, it must be valid. Possible values: <ul style="list-style-type: none">• <i>TRUE</i>• <i>FALSE</i>
<i>IdentifyByDevice-Name</i>	Indicates whether the DHCP server identifies the device by using <i>DTMDeviceName</i> or <i>MACAddress</i> .	A column with this header is required. The value is optional. If a row contains data, it must be valid. Possible values: <ul style="list-style-type: none">• <i>TRUE</i>: Requires entering a value for <i>DTMDeviceName</i>.• <i>FALSE</i>: Requires entering a value for <i>MACAddress</i>. The value is ignored and not exported/imported if <i>Enable DHCP</i> = <i>FALSE</i> .
<i>MACAddress</i>	MAC address of the device.	A column with this header is required. The value is optional. If a row contains data, it must be valid and in the proper format (hyphen as separator). The value is ignored and not exported/imported if <i>Enable DHCP</i> = <i>FALSE</i> or <i>IdentifyByDeviceName</i> = <i>TRUE</i> .
<i>DTMDeviceName</i>	Name of the device that is used by the DHCP server to identify it.	A column with this header is required. The value is optional. If a row contains data, it must be valid. The value is ignored and not exported/imported if <i>Enable DHCP</i> = <i>FALSE</i> or <i>IdentifyByDeviceName</i> = <i>FALSE</i> .

NOTE: Unless mentioned otherwise, for optional parameters for which no value is entered, the default value of the specified template or of the topology domain is used.

Editing Rules

The table indicates the rules that apply when you edit a device CSV export file.

Item	Rule						
Rows of device data	Refer to the requirements, page 569. You can delete the row of an object if you do not want to import it or modify it through import.						
Empty rows	Cannot be added.						
Comment rows							
Empty columns							
Order of columns	Cannot be changed.						
List separator	Only comma, page 568 is accepted.						
Values	Use the invariant culture, page 568. To use commas or quotation marks in values of string data types when you edit the file by using a text editor, for example, Notepad, follow these rules: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Expected value after import</th> <th style="text-align: left; padding: 2px;">Value to be entered</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;"><i>Motor, Pump</i></td> <td style="padding: 2px;">“<i>Motor, Pump</i>”</td> </tr> <tr> <td style="padding: 2px;"><i>Motor”Pump</i></td> <td style="padding: 2px;">“<i>Motor”Pump</i>”</td> </tr> </tbody> </table>	Expected value after import	Value to be entered	<i>Motor, Pump</i>	“ <i>Motor, Pump</i> ”	<i>Motor”Pump</i>	“ <i>Motor”Pump</i> ”
Expected value after import	Value to be entered						
<i>Motor, Pump</i>	“ <i>Motor, Pump</i> ”						
<i>Motor”Pump</i>	“ <i>Motor”Pump</i> ”						

Importing a CSV File Into the Topology

Overview

By using the **Import > Devices** command of the **Topology Explorer**, you can modify the devices of the topology with the configurations that you have made in the CSV file.

The import process is performed in steps during which the software:

1. Validates the structure of the CSV file.
2. If valid, displays the devices contained in the imported file along with the action that you have configured for each one, informs you of detected issues, and lets you select which objects to import.
3. Imports selected devices that satisfy applicable rules (for example, uniqueness of device name and presence of template) or informs you of the detected conflicts, page 573 and lets you resolve them, if possible.
4. Displays a summary of the import operation, which includes information about incorrect configurations and devices that were not imported.

NOTE:

- Verify that the templates that are used by the devices that you are importing exist in the Global Templates library and that they are not open for editing before you proceed with the import; otherwise you cannot import these devices. Refer to the topic describing how to export/import templates, page 881.
- The import functionality does not manage DTMs that may be required by devices. Verify that the necessary DTM are present on the computers of the target EcoStruxure Process Expert infrastructure.

Device Import Behavior

The table describes the default import behavior when you configure the export file, page 567 to *create* devices in the following scenario.

Scenario	Result
A device parameter value is invalid or no value is configured for an optional parameter.	<p>The device is created and configured with parameters whose value is valid.</p> <p>For the invalid and not configured parameter, the default value of the template or topology domain is used.</p>

The table describes the import behavior when you configure the export file, page 567 to *update* devices in the following scenario.

Scenario	Result
A device parameter value is invalid or no value is configured.	<p>Only parameters whose value is valid are updated.</p> <p>For the invalid and not configured parameter, the current value of the device is retained.</p>

NOTE: Refer to the table describing the device export file content requirements, page 569 for additional information on specific parameters.

Aborting Import Tasks

After confirming the import by clicking **OK** in the **Import Devices or Resolve Uniqueness Conflicts** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are importing only a few objects, the task may complete before you are able to click the abort icon.

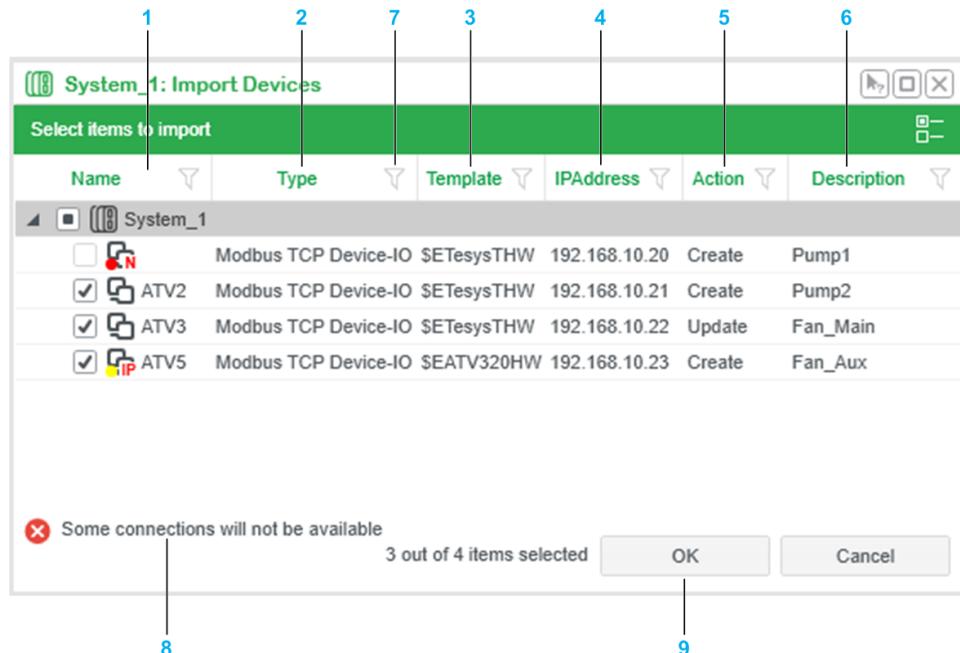
Opening the Import Devices Window

To open the **Import Devices** window, proceed as follows.

Step	Action
1	In the Topology Explorer , right-click the system root folder and select Import Devices . Result: The Open dialog box opens.
2	Select the device export file (.csv) and click Open . Result: The Import Devices window opens and displays the content of the file. NOTE: Not all device parameters and no Ethernet network data contained in the file are shown.

Import Devices Window Description

The following figure shows an example of the **Import Devices** window. This example illustrates a scenario where a conflict is detected for two devices and one of them cannot be imported.



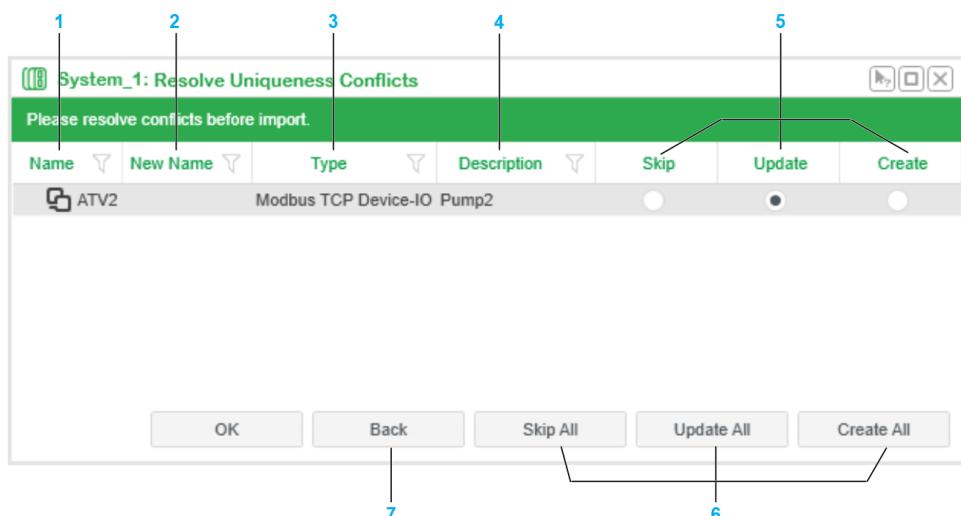
Item	Description										
1	<p>Name of the devices contained in the imported file. A selected check box indicates that the device is imported if possible. A color icon next to a device indicates a detected conflict. Details are shown in a tooltip.</p> <table border="1"> <thead> <tr> <th>Icon/letters</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Yellow dot/IP</td><td> <p>At the time of export, the device that you want to import was connected to an Ethernet network that exists in the target system and its IP address is already used on this network, page 540.</p> <p>In addition, the tooltip text indicates <conflicting IP address in the target system> - <path to the device in the target system> - <Ethernet network identifier in the target system>.</p> <p>You can still import the device but physical connections are impacted. For details, refer to the topic describing import rules for physical connections, page 558.</p> <p>NOTE: If the device that has the same IP address in the target system also has the same name, the software does not indicate an IP address conflict.</p> </td></tr> <tr> <td>Red dot/T</td><td> <p>The device that you want to import uses a template that is not present in the Global Templates library. You must import the template before you can import the device.</p> <p>For information on how to import templates, refer to the topic that describes managing Global Templates, page 881.</p> </td></tr> <tr> <td>Red dot/A</td><td>The <i>Action</i> parameter for the device in the device CSV export file is invalid. You cannot import the device.</td></tr> <tr> <td>Red dot/N</td><td>The <i>DeviceName</i> parameter of the device in the device CSV export file blank. You cannot import the device.</td></tr> </tbody> </table>	Icon/letters	Description	Yellow dot/IP	<p>At the time of export, the device that you want to import was connected to an Ethernet network that exists in the target system and its IP address is already used on this network, page 540.</p> <p>In addition, the tooltip text indicates <conflicting IP address in the target system> - <path to the device in the target system> - <Ethernet network identifier in the target system>.</p> <p>You can still import the device but physical connections are impacted. For details, refer to the topic describing import rules for physical connections, page 558.</p> <p>NOTE: If the device that has the same IP address in the target system also has the same name, the software does not indicate an IP address conflict.</p>	Red dot/T	<p>The device that you want to import uses a template that is not present in the Global Templates library. You must import the template before you can import the device.</p> <p>For information on how to import templates, refer to the topic that describes managing Global Templates, page 881.</p>	Red dot/A	The <i>Action</i> parameter for the device in the device CSV export file is invalid. You cannot import the device.	Red dot/N	The <i>DeviceName</i> parameter of the device in the device CSV export file blank. You cannot import the device.
Icon/letters	Description										
Yellow dot/IP	<p>At the time of export, the device that you want to import was connected to an Ethernet network that exists in the target system and its IP address is already used on this network, page 540.</p> <p>In addition, the tooltip text indicates <conflicting IP address in the target system> - <path to the device in the target system> - <Ethernet network identifier in the target system>.</p> <p>You can still import the device but physical connections are impacted. For details, refer to the topic describing import rules for physical connections, page 558.</p> <p>NOTE: If the device that has the same IP address in the target system also has the same name, the software does not indicate an IP address conflict.</p>										
Red dot/T	<p>The device that you want to import uses a template that is not present in the Global Templates library. You must import the template before you can import the device.</p> <p>For information on how to import templates, refer to the topic that describes managing Global Templates, page 881.</p>										
Red dot/A	The <i>Action</i> parameter for the device in the device CSV export file is invalid. You cannot import the device.										
Red dot/N	The <i>DeviceName</i> parameter of the device in the device CSV export file blank. You cannot import the device.										
Indicates the type of the topological object.											
Identifier of the template that device is using.											
IP address that is configured for the device.											

Item	Description
5	Indicates the action that the software performs for the selected device according to the value of the <i>Action</i> parameter in the imported file. NOTE: When you create a device, in the import summary that is shown in the notification panel, the action appears as <i>Create</i> for the device but as <i>Update</i> for its parameters.
6	Content of the <i>Description</i> column.
7	You can sort and filter, page 118 data.
8	Indicates that some physical connections, either configured in the imported file or that exist in the target topology, will not be available after import. This is because conflicts related to IP addresses and/or Ethernet networks have been detected.
9	If devices can be imported without opening the conflict resolution window, the OK button is displayed, which proceeds with the import. Otherwise, it is the Resolve button, which opens the Resolve Uniqueness Conflicts window.

Resolving Conflicts During Device Import

You can resolve the device-uniqueness conflicts that the software detects during import, for example, if you are creating a device through import and one with the same name already exists in the topology.

The following figure shows an example of the **Resolve Uniqueness Conflicts** window in which one device is displayed.



Item	Description
1	Name of the device as it appears in the imported file.
2	New name of the device if you select to create it instead of updating the existing one.
3	Indicates the type of the topological object.
4	Content of the <i>Description</i> column.
5	Buttons that let you set an action for individual devices: <ul style="list-style-type: none"> Skip: The device is ignored for the import. Update: Default action. Modifies the existing device with the imported parameter configuration. The name of the device is not modified. Create: Creates a new device with New Name and the imported parameter configuration.
6	Buttons that let you set an action for the conflicting devices as a whole. The actions are the same as when you set the action for devices individually. NOTE: Clicking these buttons overrides the selection you have made for individual devices.
7	Reverts to the Import Devices window. Your selection for individual devices is retained.

Importing Devices into the Topology

To import the content of a device CSV export file in the topology of a system, proceed as follows.

Step	Action
1	In the Import Devices window select the devices that you want to import.
2	If the Resolve button is displayed, proceed to step 3. Otherwise, click OK . Result: The software proceeds as follows: <ul style="list-style-type: none">• Imports the content of the file based on your selection.• Displays the imported devices in the Topology Explorer.• Displays a summary of the import process in the notification pane.
3	In the Import Devices window, click Resolve . Result: The Resolve Uniqueness Conflicts window opens.
4	Select the actions that you want to perform for conflicting devices.
5	Click OK . Result: The software proceeds as follows: <ul style="list-style-type: none">• Imports the devices that were previously shown in the Import Devices window and those in the Resolve Uniqueness Conflicts window, based on your selection.• Displays the entirety of the imported devices in the Topology Explorer.• Displays a summary of the entire import process in the notification pane.

Mapping Stage

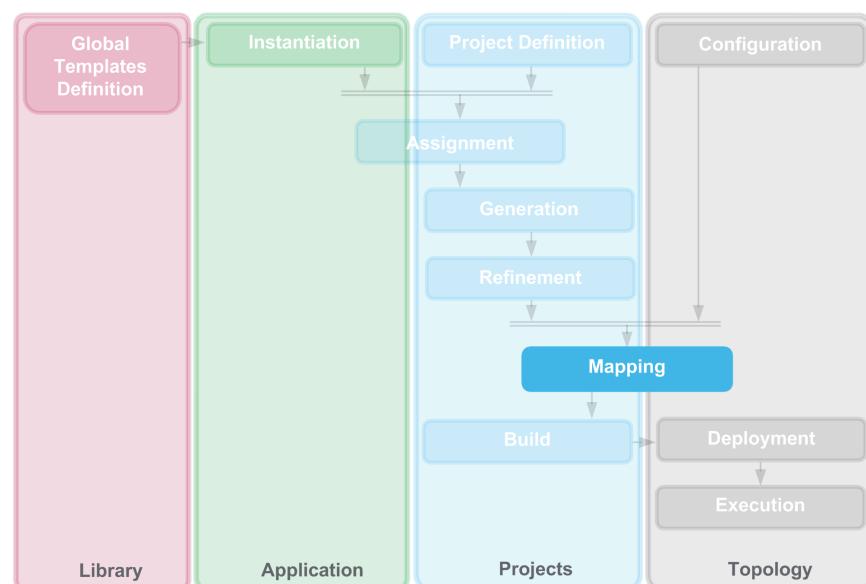
What's in This Chapter

Control Project Mapping Stage	577
Exporting and Importing Hardware Mappings	609
Supervision Project Mapping Stage	618

Overview

This chapter describes how to map the Control and Supervision projects to the topological entities of the system, and to create communication channels, which allow you to use the I/O scanner function of controllers.

The following figure shows the position of the **Mapping** stage within the system engineering life cycle.



Refer to the Mapping stage, page 51 for a description of the purpose of this stage.

Control Project Mapping Stage

Overview

This section describes how to use the **Project Explorer** to map a logical Control Participant project to one or more topological entities of the system to map the application to hardware channels. It also describes how to create and configure communication channels that allow using the controller I/O scanner function for communication with devices and peer to peer communication.

You must have completed the Generation stage, page 383 and the Configuration stage, page 498 of the Control Participant project to proceed with the mapping stage.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Creating Control Project Executables

Overview

A Control executable allows you to associate to the logical Control Participant project a topological entity acting as engine, page 499 by mapping the **Service**. Based on that, you can further map **Hardware** (I/O and communication modules) and **Communication** (communication channels for various communication modules) of this engine.

You can map a logical Control Participant project to several engines of the system. To do so, create one executable for each engine.

For example, you can map the same logical Control Participant project to both:

- The controller that is in the plant by using executable_A.
- Another controller that is in the laboratory for test purposes by using executable_B.

Executables are associated to an **Execution Domain**, which serves as a filter for selecting applicable services, for example, to define the boundaries for peer to peer communication or runtime navigation services.

Creating Control Executables

To create a Control executable, proceed as follows.

Step	Action
1	<p>In the Control Project Browser, right-click the Executables node of the Control project that you want to map and select Create Executable.</p> <p>Result: A Control executable is created under the Executables node.</p>

Control Executable Actions

The table describes the executable context menu commands.

Command	Description
Manage	<p>Opens the Manage window in the work area of the Project Explorer, which lets you perform the following actions:</p> <ul style="list-style-type: none"> • Service mapping, page 580: Associates the logical Control Participant project to an engine of the system. • View and edit the properties, page 579 of the executable. • Communication mapping, page 582: Defines the Modbus TCP Ethernet (I/O scanning) communication channels between a controller and devices and/or other controllers for peer to peer communication. • Hardware mapping, page 602: Lets you map the interfaces of the instances of the application that are assigned to the project to hardware channels of the controller.
Build , page 629	<p>Starts an automated process to integrate certain changes in the built Control Participant project by performing an incremental build.</p> <p>The command is available once you have done the service mapping.</p>
Build All	<p>Starts an automated process to create, page 630 the built Control Participant project the first time or to recreate it to integrate certain changes, page 636 that you have made to the Control Participant project or the system topology.</p> <p>The command is available once you have done the service mapping.</p>
Generate and Build , page 386	<p>Starts a generation of the Control Participant project followed by a build.</p> <p>If you have not built the logical Control Participant project yet, the build process that is executed is the same as when you select the Build All command.</p> <p>If the generation process does not complete successfully, no build is performed and information is displayed in the notification panel.</p> <p>NOTE: After selecting the command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task.</p>

Command	Description
Open Built Project, page 634	Opens the built Control project in the Control Participant, which allows you to view the program and configuration. The command is available once you have built the Control Participant project.
Delete	Deletes the executable including any associated: <ul style="list-style-type: none"> • Service and hardware mappings. • Communication mappings. NOTE: Deleting an executable does not delete data backup files, page 286.
Rename	Lets you enter a new identifier, page 579 for the executable.

Control Executable Properties

To open the properties of the executable, double-click it.

You can view and edit the following properties.

Property	Description
Identifier	The identifier must be unique within a Control Participant project.
Description	Optional. You can enter a description for the executable with free form text.
Execution Domain	Optional. String. Default value: Blank. For the following types of communication, you can exchange data only if these executables have the same execution domain: <ul style="list-style-type: none"> • For peer to peer communication: The executable of the owner project and the executable of the consumer project. • For runtime navigation services (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide): The executable of the Control Participant project and the executable of the Supervision Participant project. Values that you enter remain in the menu and you can select those of executables of other Control and Supervision Participant projects. You can assign a logical execution domain to the executable with free form text. You can modify the Execution Domain value at any time during the engineering life cycle. NOTE: Leaving Execution Domain blank is considered as a value.
Build State	Indicates the status, page 634 of the executable. Default value: Not Built
Built On	Date and time when you last built the executable successfully by using a build command.

Assigning an Execution Domain

To assign an execution domain to a Control executable, proceed as follows.

Step	Action
1	Double-click the executable to open its properties.
2	Select a value from the Execution Domain menu or enter a new value.

NOTE: To remove or change the associated execution domain, select the empty row at the top of the menu.

Mapping Services

Overview

The **Service Mapping Editor** opens within the **Manage** window when you double-click the Control executable, page 578. It allows you to associate the logical Control Participant project to entities acting as engines, page 499 through its Control executable.

These engines are controllers and Control services of station nodes that exist in the topology of the system.

NOTE: If the **Controller Family** is not the same for the entity and the Control project, page 261 that you want to map, you may not be able to map certain hardware I/O interfaces.

Mapping to Other Topological Entities

The software does not display STB islands, PROFIBUS Remote Masters (PRMs), Modbus TCP, nor EtherNet/IP I/O devices in the **Service Mapping Editor**. For these topological entities, you need to map to the Control executable the controller that communicates with the entity.

NOTE: For PRMs, verify that the family of the controller is the same as the family that is selected in the properties of the PRM, page 528.

Mapping to a Redundant Controller

The software displays the identifier of the controller entity in the **Service Mapping Editor** when you map the Control executable to a redundant controller. The mapping however, is performed with both the primary and the secondary controllers.

Mapping to a Station Node

When you map a Control executable to a station node to use the controller simulator, the station node is available in the **Service Mapping Editor** window only if you have created at least one ControlExpert service, page 517 in the station node.

If you have created several ControlExpert services, their identifiers are shown in brackets after the station node identifier and you can select the one that you want to use.

Changing the Service Mapping of Deployed Executables

If you have already deployed the Control executable to an engine of the system, you cannot change nor remove the service mapping of this executable. You need to create a new executable.

Mapping Services

To map the Control executable to an engine, proceed as follows.

Step	Action
1	<p>Double-click the executable of the Control project that you want to map and select Manage.</p> <p>Result: The Service Mapping Editor opens.</p>
2	<p>Click the Engine menu.</p> <p>Result: The menu displays the identifiers of compatible entities of the topology of the system.</p>

Step	Action
3	If you have created peer to peer communication mappings, page 462, the Data Server for Peer to Peer Communication menu is also shown.
4	Select an entity and/or the communication module that you want to use for peer to peer communication. Result: The software: <ul style="list-style-type: none">• Maps the Control executable to the selected entity.• Displays the Hardware Mapping tab at the bottom of the window. If the controller is connected to an Ethernet network, page 540, the Communication Mapping tab appears also.

NOTE: To remove a mapping, in the **Service Mapping Editor**, right-click an entity and select **Unmap**.

Mapping Communication

Overview

To open the **Communication Mapping Editor**, click the **Communication Mapping** tab within the **Manage** window, page 578. The tab appears once you have performed the service mapping, page 580 with a controller that is connected to an Ethernet network, page 540.

It allows you to define the communication channels that are used for communication between the controller that you have configured as client and service mapped and:

- Modbus TCP devices.
- STB islands.
- PRMs.
- Other controllers that are mapped to other logical Control Participant projects for peer to peer communication.

This method uses the I/O scanner service of the controller that is configured as client.

A communication channel is the logical representation, at the platform level, of the Control Participant I/O scanner row.

During the creation of a communication channel, you can configure its properties, page 586.

For peer to peer communication, the **Communication Mapping Editor** also allows you to map network variables, page 594 to communication channels.

The software creates the I/O scanner lines that correspond to the communication channels in the controller acting as client during the build process, page 630.

NOTE: For peer to peer communication, proceed with the communication mapping from the executable of the consumer project, page 452.

NOTE: The communication mapping feature is not supported for executables that you have mapped to a station node that is emulating a simulator.

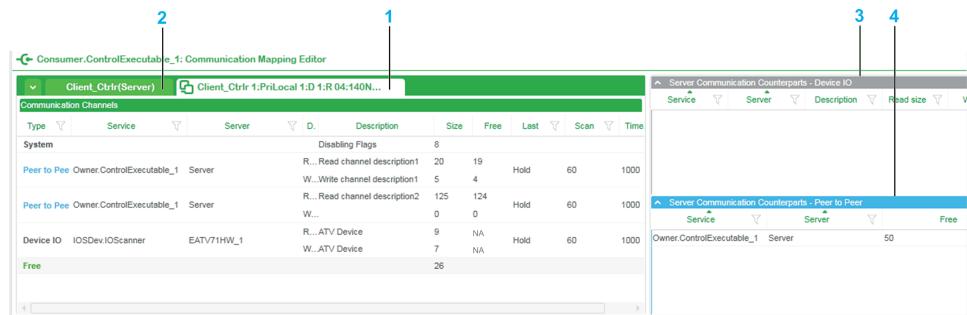
Prerequisites

For a description of the prerequisites for communication mapping, refer to the topics that describe the steps to complete:

- Communication with devices, page 780.
- Peer to peer communication, page 787.

Communication Mapping Editor

The following figure shows an example of the **Communication Mapping Editor** window, which displays two peer to peer communication and one Modbus I/O device communication channels. It also indicates that free memory remains in the server counterpart to create additional peer to peer communication channels and/or increase the size of existing channels.



Item	Description
1	Communication Channels pane, which displays the client memory area of the communication module of the controller that you have mapped to the Control executable. There is one tab for each communication module (including the CPU communication module) that is configured in the controller and whose I/O scanning service is enabled. Select a tab to display the communication channels that are configured in the memory area of the corresponding communication module of the controller acting as client.
2	<controller_identifier>(Server) tab, which displays the communication channels that are configured in the memory area of the controller when acting as server (if applicable). The information is displayed in the Server Memory Map pane.
3	Server Communication Counterparts - Device IO pane, which displays, topological entities such as Modbus TCP devices, PRMs, and STB islands, acting as servers, and with which you can define a communication channel. Click the arrow in the title bar to expand/collapse the pane
4	Server Communication Counterparts - Peer to Peer pane, which displays other Control projects, their executable, and the mapped controller entity with which you can define a communication channel. Click the arrow button in the title bar to expand/collapse the pane.

NOTE: When the executable of the Control project is mapped to a redundant controller, the **Communication Channels** pane displays only the client memory area of the communication module of the primary controller.

The figure shows an example of the **Server Memory Map** pane, which is displayed when you select the <controller_identifier>(Server) tab. It indicates that one peer to peer communication channel exists in the memory area of the client controller when acting as server, page 588.



The table describes the fields of the **Communication Channels** pane when you select the tab of the client communication module.

Column header	Description
Type	System Indicates in the Size column, the size of memory that is reserved, and in the Description column, the corresponding usage.
	Free Refer to the description of the Size column.
	Device IO Identifies channels that exist to communicate with Modbus TCP Ethernet devices, PRMs, and STB islands.
	Peer to Peer Identifies channels that exist to exchange data between Control Participant projects through the controllers that are mapped to these projects.
Service	<ul style="list-style-type: none"> For I/O device communication channels: <i>element category</i>. <i>communication element</i> Where <i>element category</i> is the category of the I/O scanner element, and <i>communication element</i> the name of the I/O scanner element for the I/O device. For peer to peer communication: <i>project.executable</i> Where <i>project</i> is the identifier of the Control Participant project (owner, page 452) that exposes the variables and <i>executable</i> the identifier of the executable of this project.
Server	Identifier of the server counterpart using the communication channel: <ul style="list-style-type: none"> The I/O device identifier for I/O device channels. The controller identifier for peer to peer communication channels.
Direction	Indicates the direction of the communication as Read , Write , or Both .
Description	<p>Displays the value of the parameters Read description and Write description that you can view in the Channel Properties dialog box, page 586.</p> <p>NOTE: For channels of type System indicates the usage of the reserved memory.</p>
Size	<p>For each channel, indicates the memory size that is allocated for reading from the server and writing to the server. It corresponds to the read and/or write size that you enter when you create a communication channel with a server counter part.</p> <p>You can create communication channels until there is not enough sufficient memory left.</p> <p>NOTE:</p> <ul style="list-style-type: none"> For channel type Free, indicates the remaining free memory size that you can use to create communication channels. It is calculated based on the memory size that you have configured in the memory heap of the client by using the ClientMemoryLength parameter, page 509 and on the memory allocation of existing communication channels. Additional words used by the System (for example, for disabling flags) are also deducted from the memory size. For controllers of the M580 platform and NOC communication modules of any controller, the size value for channel type Free is not indicated because it is managed during build only, page 630.
Free	<p>Displays the remaining free memory in words for each existing peer to peer communication channel. The size is indicated separately for read and write channels when they exist. It is calculated based on Size and the size of variables that you have mapped, page 594 to read and/or write communication channels.</p> <p>For example, if the read channel size is 100 and you have mapped variables to this channel, which use up 10 words, the displayed value is 89. If you have mapped variables that use less than a word (a variable of BOOL data type), the free size can be indicated in decimals. These correspond to 1/16th of a word. For example, a free memory size indicated as 8.14 (out of 10) corresponds to 8 words and 14/16th of a word (2/16th of a word are used by variables of BOOL data type, which use up 1/16th of a word each).</p> <p>NOTE: 1 word of each channel is reserved to move the channel address to manage the 32-bit alignment for peer to peer communication across Modicon Quantum and M340 platforms. This word is deducted from the remaining free memory size.</p>

Column header	Description
Last	Displays the value of the parameter Last value that you can select and adjust through the Channel Properties dialog box, page 586.
Scan	Displays the value of the parameter Scan rate that you can adjust through the Channel Properties dialog box, page 586.
Timeout	Displays the value of the parameter Timeout that you can adjust through the Channel Properties dialog box, page 586.

The table describes the fields of the **Server Communication Counterparts - Device IO** pane.

Field	Description
Service	<i>element category.communication element</i> , where <i>element category</i> is the category of the I/O scanner element, and <i>communication element</i> the name of the I/O scanner element for the I/O device.
Server	Identifier of the STB island, PRM, or Modbus TCP device entity, acting as server and with which you can create a communication channel.
Description	Description of the data to be exchanged through the channel.
Read Size	Number of words that are read from the server.
Write Size	Number of words that are written to the server.

The table describes the fields of the **Server Communication Counterparts - Peer to Peer** pane.

Field	Description
Service	<i>project.executable</i> Where <i>project</i> is the identifier of the Control Participant project (owner, page 452) that exposes the variables and <i>executable</i> the identifier of the executable of this project.
Server	Identifier of the controller mapped to the owner project described in the Service field, which acts as counterpart, and that is available to create a communication channel.
Free	When no communication channel exists, indicates the number of words that you have configured in the memory heap of the controller acting as server, which is defined in the Server field. When you create a communication channel, the read and write sizes that you define in the Peer to Peer: Properties dialog box are deducted from this value.

Customizing Column Display

In the various panes of the editor, you can choose which columns you want to see by right-clicking a column header and selecting **Customize**. Certain columns are always shown.

When you hide a column for which a filter is applied, the filter is cleared.

Communication Channel Actions

Right-click a communication channel in the **Communication Channels** pane to display a context menu with the following commands.

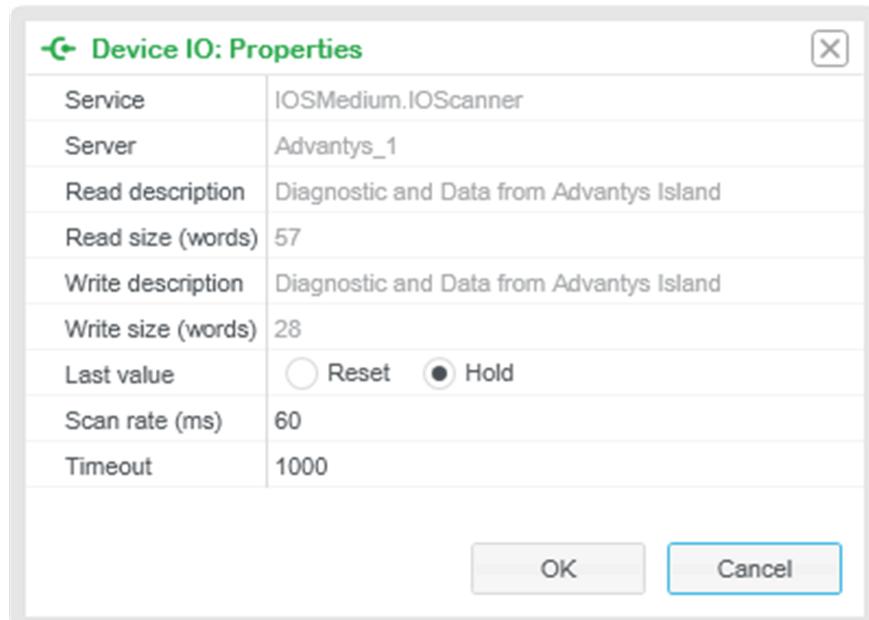
Command	Description
Map Variables	Opens the Variable Mapping Editor , page 594, which allows you to map network variables and define the relative position of such variables in the selected communication channel. The command is available only for peer to peer communication channels.
Unmap	Removes the selected communication channel, page 591.
Properties	Opens the Device IO: Properties or Peer to Peer: Properties dialog box, page 586, which lets you view and/or adjust parameter values of the selected communication channel.

Communication Channel Properties

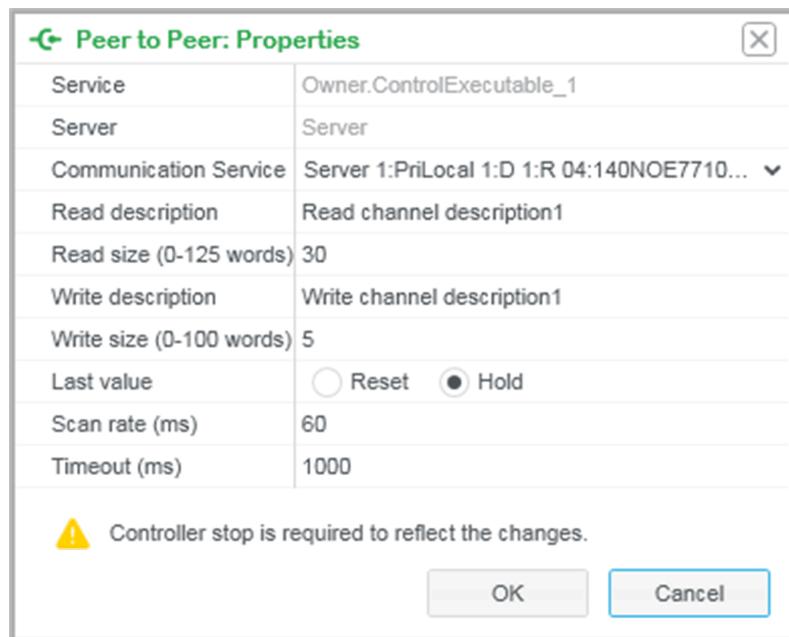
When you create a communication channel, page 590, the software opens the appropriate dialog box, which allows you to adjust the values of the parameters of the communication channel in the memory heap of the client.

NOTE: You can view and/or adjust the values of certain parameters of existing communication channels.

The following figure shows an example of the **Device IO: Properties** dialog box that the software opens when you create a channel to communicate with an STB island. The dialog box is also displayed when you right-click the existing communication channel and select **Properties**.



The following figure shows an example of the **Peer to Peer: Properties** dialog box that the software opens when you create a peer to peer communication channel. The dialog box is also displayed when you right-click the existing communication channel and select **Properties**.



Parameter	Description
Service	Refer to the description of the Service column of the Server Communication Counterparts - Device IO or Server Communication Counterparts - Peer to Peer pane.
Server	Refer to the description of the Server column of the Server Communication Counterparts - Device IO or Server Communication Counterparts - Peer to Peer pane.
Communication Service	The parameter appears only in the Peer to Peer: Properties dialog box. Identifier of the communication module device through which data is exchanged. You can select from the communication modules that exist in the server controller and which are connected to the same Ethernet network as the client controller communication module.
Read description	Description for the read channel. NOTE: For peer to peer communication channels, the default value is blank. You must enter a description.
Read size	Number of words that are allocated to the channel and read from the server counterpart. The corresponding memory size is reserved in both the client and server memory heap. NOTE: For peer to peer communication channels: <ul style="list-style-type: none">• Default value: 0• Maximum value: 125• You can adjust the value:<ul style="list-style-type: none">◦ The total value of columns Read size and Write size cannot exceed the lowest value indicated in either column Free of the Server Communication Counterparts pane or column Size (row Free) of the Communication Channels pane.◦ You cannot set the size to a value that is lower than the total size of variables that are mapped, page 594 to the channel.◦ You can increase the size as long as there is no overlapping channel in the server memory heap within the new size limit.
Write description	Description for the write channel. NOTE: For peer to peer communication channels, the default value is blank. If you enter a value in the Write size field, you must enter a description.
Write size	Number of words that are allocated to the channel and written to the server counterpart.

Parameter	Description
	<p>The corresponding memory size is reserved in both the client and server memory heap.</p> <p>NOTE: For peer to peer communication channels:</p> <ul style="list-style-type: none"> • Default value: 0 • Maximum value: 100 • You can adjust the value: <ul style="list-style-type: none"> ◦ The total value of columns Read size and Write size cannot exceed the lowest value indicated in either column Free of the Server Communication Counterparts pane or column Size (row Free) of the Communication Channels pane. ◦ You cannot set the size to a value that is lower than the total size of variables that are mapped, page 594 to the channel. ◦ You can increase the size as long as there is no overlapping channel in the server memory heap within the new size limit.
Last value	<p>Value that is retained when a communication interruption occurs.</p> <p>Select between:</p> <ul style="list-style-type: none"> • Reset: Set to 0 • Hold: Last value <p>Default value: Hold</p> <p>NOTE: This setting applies to the I/O scanner line created in the Control Participant project. When communicating with an I/O device modeled by a device template, confirm that the DFB contained in the device template supports the selected setting. If not, the value of variables read from the device may not be synchronized with the value sent to the Supervision layer by the DFB.</p> <p>For information on parameters of DFBs of device templates, refer to the help of the corresponding template library, which documents the Control resources (services).</p>
Scan rate	<p>The rate at which data is scanned in the channel.</p> <p>Range: 0...65535 ms</p> <p>Default value: 60 ms</p> <p>If you change the default value, verify that the new value is a multiple of the repetitive rate step.</p> <p>For more information, refer to the Control Participant help.</p>
Timeout	<p>The maximum interval between responses from the entity for the channel. After this time expires, the communication is considered interrupted, and the last value is retained according to the configuration of the Last value parameter.</p> <p>Range: 1...65535 ms</p> <p>Interval: 1 ms</p> <p>Default value: 1000 ms</p> <p>NOTE: Verify that the TimeOut value is greater than the ScanRate value.</p> <p>For more information, refer to the Control Participant help.</p>

Server-Side Communication Channel Properties

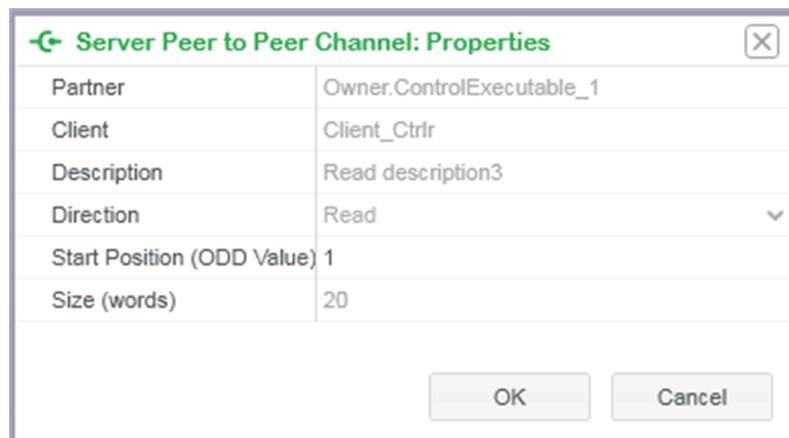
Once you have created a peer to peer communication channel in the memory heap of the client, you can view properties of the channel server-side by right-clicking the communication channel and selecting **Properties**.

To view the communication channel, first open, page 582 the **Communication Mapping Editor** from the owner project.

NOTE: You can open the an identical properties dialog box for the communication channel that exists in the client controller when acting as server by selecting the <controller_identifier>(Server) tab from the client-side communication mapping window, right-clicking the communication channel in the **Server Memory Map** pane, and selecting **Properties**.

NOTE: When you modify the properties of a communication channel server-side, and you have already deployed the associated built Control Participant project, you can only apply the change to the engine, using the **Deploy Built Project**, page 686 command, which requires stopping the controller.

The following figure shows an example of the **Server Peer to Peer Channel: Properties** dialog box, which indicates properties of the peer to peer communication channel server-side.



Parameter	Description
Partner	<i>project.executable</i> Where <i>project</i> is the identifier of the Control Participant project (consumer, page 452) that receives the variables and <i>executable</i> the identifier of the executable of this project.
Client	Identifier of the controller entity (server) to which the owner project is mapped.
Description	Description for the read or write channel.
Direction	Read or Write.
Start position	Starting address of the communication channel in the memory heap of the server. The value is editable, allowing you to move communication channels server-side, page 592.
Size	Number of words that are allocated to the communication channel server-side.

Communication Channel Position

Communication channels are contiguous in the memory area of the controller acting as client.

Communication Channel Order

Device I/O communication channels appear in the memory area reserved for this type of communication channel in the order you create them.

When both peer to peer communication channels and device I/O communication channels exist then device I/O communication channels are positioned after peer to peer communication channels.

Restrictions when Creating Communication Channels

For each communication module of a controller acting as client (a tab of the **Communication Channels** pane), you cannot create communication channels with several executables that are mapped to the same controller.

For example, if the following executables appear in the **Service** column of the **Server Communication Counterparts - Peer to Peer** pane and both are mapped (service mapping) to controller *Controller_1* (**Server** column):

- *Project_1.Executable_1*
- *Project_2.Executable_2*

Then, for a given client controller communication module, you can create a communication channel only with either one of these executables.

Creating Communication Channels

When you create a communication channel, and you have already deployed the associated built Control Participant project, you can only apply the change to the engine, using the **Deploy Built Project**, page 686 command, which requires stopping the controller.

NOTE: Following the creation of a peer to peer communication channel, you can view it in the memory heap of the server, in the **Server Memory Map** pane (<controllerIdentifier>(server) tab), by opening, page 582 the **Communication Mapping Editor** from the counterpart owner project.

Creating a peer to peer communication channel in the executable of the consumer project also affects the executable of the counterpart owner project. To apply changes, you need to start/stop the controllers to which these executables are deployed in the right order; otherwise incorrect data may be read and/or written.

⚠ WARNING

LOSS OF CONTROL

- Deploy the executables to the controllers between which the peer to peer communication channel exists.
- Start/stop controllers in the right order.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To create a communication channel between the client and the server counterpart, proceed as follows.

Step	Action
1	<p>From one of the Server Communication Counterparts panes, drag the server counterpart with which you want to create a communication channel to the Communication Channels pane.</p> <p>Result: The corresponding communication channel Properties dialog box opens.</p>
2	Adjust the values of the communication channel properties as required.
3	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> • Creates a communication channel between the selected server counterpart and the client with the parameter values that you have configured in the corresponding Properties dialog box. • Shifts the information of the server counterpart to the respective columns in the Communication Channels pane. • Updates values of the Size and Free columns. <p>NOTE: Click Cancel to close the Properties dialog box without creating the communication channel.</p>

NOTE:

- If there is no sufficient space in the memory of the client or server to accommodate the communication channel, the software displays a notification and you cannot create the channel. Adjust the client or server memory properties of the corresponding controller entity and repeat.
- The validation of sufficient space in the memory of the client is not performed for controllers of the M580 platform and NOC communication modules of any controller because the free memory size is managed during build only, page 630.

Modifying Communication Channel Properties

You can view and modify the values of certain parameters of existing communication channels in the **Properties** dialog box of the channel.

When you modify the properties of a communication channel, and you have already deployed the associated built Control Participant project, you can only apply the change to the engine, using the **Deploy Built Project**, page 686 command, which requires stopping the controller.

Modifying peer to peer communication channel properties in the executable of the consumer project also affects the executable of the counterpart owner project. To apply changes, you need to start/stop the controllers to which these executables are deployed in the right order; otherwise incorrect data may be read and/or written.

⚠ WARNING	
LOSS OF CONTROL	
<ul style="list-style-type: none"> Deploy the executables to the controllers between which the peer to peer communication channel exists. Start/stop controllers in the right order. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>	

To modify properties of an existing communication channel, proceed as follows.

Step	Action
1	<p>In the Communication Channels pane, right-click the communication channel and select Properties.</p> <p>Result: The software opens the Properties dialog box.</p> <p>NOTE: The Properties dialog box does not open if the Variable Mapping Editor window, page 594 of this communication channel is open.</p>
2	Adjust the values of the communication channel properties as required.
3	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> Adjusts the communication channel properties. Updates the values that are displayed in the Communication Mapping Editor. Displays information on the modification in the notification panel. <p>NOTE: Click Cancel to close the Properties dialog box without creating the communication channel.</p>

NOTE: If the software is not able to apply your modification, it displays a notification. The validation of sufficient space in the memory of the client is not performed for controllers of the M580 platform and NOC communication modules of any controller because the free memory size is managed during build only, page 630.

Unmapping Communication Channels

Unmapping a communication channel has the following impacts:

- The memory size allocated to the channel is released in the memory heap of the client and of the server.
- Other channels that exist in the memory heap of the client are realigned to be contiguous.
- Network variables that are mapped, page 594 to the deleted channel are unmapped from the client and server communication channels.
- The build status, page 636 of the Control executable to which the controller is mapped changes to out of date if it is built.

- When you unmap a communication channel and you have already deployed the associated built Control Participant project, you can only apply the change to the engine, using the **Deploy Built Project**, page 686 command, which requires stopping the controller.

The executables of both the consumer and the counterpart owner Control projects are affected. To apply changes, you need to start/stop the controllers to which these executables are deployed in the right order; otherwise incorrect data may be read and/or written.

⚠ WARNING

LOSS OF CONTROL

- Deploy the executables to the controllers between which the peer to peer communication channel exists.
- Start/stop controllers in the right order.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To unmap a communication channel, proceed as follows.

Step	Action
1	<p>In the Communication Channels pane, right-click the communication channel and select Unmap.</p> <p>Result: The software opens the Unmap dialog box.</p>
2	<p>Click OK to confirm.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> Unmaps the communication channel between the selected server counterpart and the client. Displays the server counterpart in the corresponding Server Communication Counterparts pane. Updates values of the Size and Free columns. <p>NOTE: Click No to close the Unmap dialog box without unmapping the channel.</p>

Moving Communication Channels Server-Side

When you move an existing peer to peer communication channel within the memory heap of the server counterpart:

- You can only enter an odd starting address.
- The starting address has to leave enough free memory area to fit the size of the communication channel; otherwise you cannot move the channel.

When you move a communication channel and you have already deployed the associated built Control Participant project, you can only apply the change to the engine, using the **Deploy Built Project**, page 686 command, which requires stopping the controller.

Moving a peer to peer communication channel in the executable of the owner project also affects the executable of the consumer counterpart project. To apply changes, you need to start/stop the controllers to which these executables are deployed in the right order; otherwise incorrect data may be read and/or written.

⚠ WARNING

LOSS OF CONTROL

- Deploy the executables to the controllers between which the peer to peer communication channel exists.
- Start/stop controllers in the right order.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To move a peer to peer communication channel in the memory heap of the server counterpart, proceed as follows.

Step	Action
1	Open, page 582 the Communication Mapping Editor from the owner, page 452 project.
2	Select the <controller_identifier>(Server) tab to display the memory heap of the server. Result: The software displays the Server Memory Map pane.
3	Right-click the communication channel that you want to move and select Properties . Result: The software opens the Server Peer to Peer Channel: Properties dialog box.
4	Enter a new positive, odd integer value in the Start position field.
5	Click OK . Result: If sufficient free memory is available at the specified position, the software: <ul style="list-style-type: none">• Moves the selected communication channel to the new start position.• Updates values of the Start Position and Size columns in the Server Memory Map pane of the Communication Mapping Editor. If not, the software displays a dialog box informing you that it cannot move the communication channel. NOTE: Click Cancel to close the Server Peer to Peer Channel: Properties dialog box without moving the channel.

Mapping Network Variables

Overview

The **Variable Mapping Editor** opens when you right-click a peer to peer communication channel in the **Communication Mapping** pane, page 583 and select **Map Variables**.

It allows you to map variables of the consumer project to peer to peer communication channels that exist in the memory heap of the client:

- Network variables, page 452: To read data from the server.
- Variables, which are the counterpart to network variables that exist in the owner project: To write data to the server.

You can assign a position manually or let the software manage positions.

The software indicates if variables are already mapped to another channel and lets you open the **Variable Mapping Editor** of this channel.

NOTE: You can work simultaneously on different communication channels. The variable mappings that are performed on a channel and that affect other channels are reflected right away on these channels.

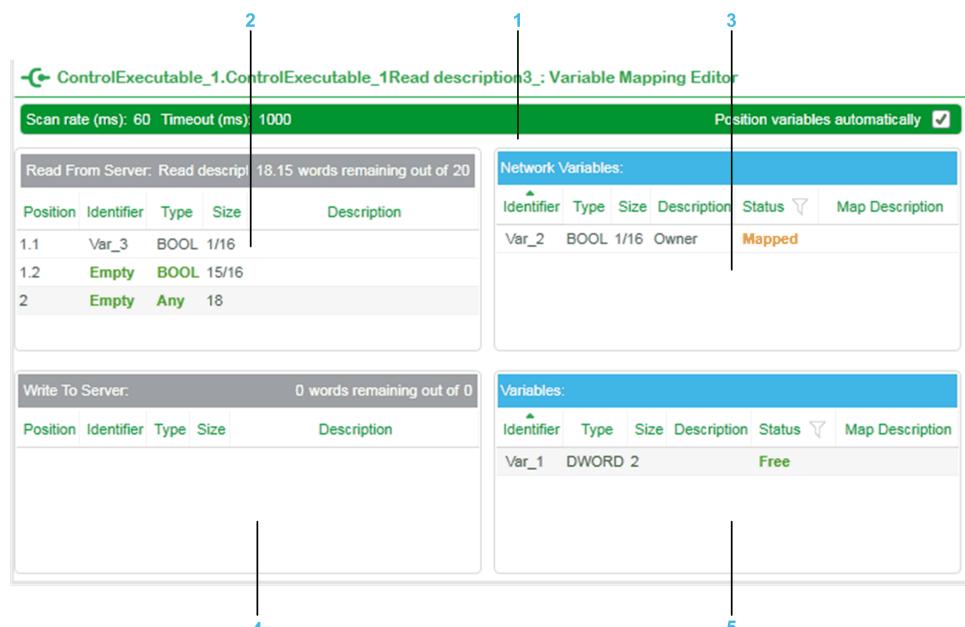
NOTE: The first time you open the **Variable Mapping Editor** and the first time you open it after refining the project, it may take time before the window appears.

Prerequisites

To proceed with the variable mapping, first create a peer to peer communication channel, page 582.

Variable Mapping Editor

The following figure shows an example of the **Variable Mapping Editor**.



Item	Description
1	Information bar, which displays scan rate and timeout properties of the selected communication channel, as well as the check box to position variables automatically, page 596.
2	Pane that displays the network variables that the client reads from its server counterpart, and that are mapped to the selected communication channel. In this example, network variable <i>Var_3</i> , which is created in the consumer project is mapped to the communication channel that exists in the client controller.

Item	Description
3	Pane that displays the network variables that exist in the consumer project and that you can map to the communication channel that is created in the client controller. In this example, network variable <i>Var_2</i> has already been mapped by opening the Variable Mapping Editor from the owner project and using its counterpart variable to write to the server, page 452. You cannot map it again. You can view its mapping information by right-clicking Mapped .
4	Pane that displays the variables that the client writes to its server counterpart, and that are mapped to the selected communication channel. In this example, no variables are mapped.
5	Pane that displays the variables that exist in the consumer project for which a corresponding network variable exists in the owner project and that you can map to the communication channel. In this example, <i>Var_1</i> exists as network variable in the owner project and here you can map its counterpart variable to the channel that exists in the client controller to write to the server, page 452.

The table describes the fields of the **Network Variables** and **Variables** panes.

Column header	Description
Identifier	Identifier of the variable.
Type	Data type of the variable. If the software detects inconsistencies in the type, the variable is shown in red color and a tooltip provides additional information. NOTE: When peer to peer communication with the topological entity acting as client is implemented by using an M580 CPU or a NOC communication module, only variables of the reference type (REF_TO) are shown, page 456.
Size	Size of the variable in words. NOTE: For variables of a size of less than 1 word, the size is displayed in corresponding fractions of words (for example, for data type BOOL, the size is displayed in 1/16th of words).
Description	Displays the contents of the Comment property of the Control Participant of the variable in the consumer project.
Status	Indicates the mapping status of the variable. Possible values: <ul style="list-style-type: none"> • Free: The variable is not yet mapped to a communication channel. • Mapped: The variable is already mapped to another communication channel of the Control executable. Click the filter icon in the column header to select the statuses that you want to see. NOTE: You can open the Variable Mapping Editor of the communication channel to which a variable with status Mapped is mapped by right-clicking the variable and selecting Go To Variable Mapping Editor .
Map Description	For variables that are mapped to another communication channel, displays the content of the Read description or Write description field of this channel; otherwise the field is empty.

The table describes the fields of the **Read from Server** and **Write to Server** panes.

Column header	Description
Pane title bar	<p>For the Read from Server pane, displays:</p> <ul style="list-style-type: none"> The contents of the field Read description. The read size (in words) of the selected channel in the client memory heap and the remaining words available to map variables. <p>For the Write to Server pane, displays:</p> <ul style="list-style-type: none"> The contents of the field Write description. The write size (in words) of the selected channel in the client memory heap and the remaining words available to map variables. <p>NOTE: One word of each channel is reserved to move the channel address to manage the 32-bit alignment, page 633 for peer to peer communication across Modicon Quantum, M580, and M340 platforms.</p>
Position	<p>Position of the word to which you have mapped the variable within the memory size that you have allocated to the channel or the next available empty position within a word or of the next word.</p> <p>The position in the communication channel is the same in the client and in the server.</p> <p>For variables of a size of less than 1 word, positions are displayed using the x.y format where:</p> <ul style="list-style-type: none"> x represents the word y represents the position inside the word
Identifier	<p>Identifier of the mapped variable or Empty.</p> <p>Empty: Appears for the next available position to which you can map variables.</p>
Type	<p>Type of the mapped variable, of the remaining space in a word, or Any.</p> <p>To map variables to the remaining space of a word, the data type of the variables needs to match.</p> <p>Any: Appears when Identifier is Empty. You can map a variable of any data type to this space.</p>
Size	<p>Indicates either, the size of the mapped variable in words, the size of the remaining space inside a word, or the size of the empty space available for mapping.</p> <p>NOTE: For mapped variables of a size of less than 1 word, the size is displayed in corresponding fractions of words (for example, for data type BOOL, the size is displayed in 1/16th of words). The empty space of a word is indicated similarly.</p>
Description	<p>Displays the contents of the Comment property of the Control Participant of the variable in the consumer project.</p>

Customizing Column Display

In the various panes of the editor, you can choose which columns you want to see by right-clicking a column header and selecting **Customize**. Certain columns are always shown.

When you hide a column for which a filter is applied, the filter is cleared.

Position of Mapped Variables

The **Position variables automatically** check box lets the software define automatically the position of the variables that you are mapping or moving:

- Selected (default setting): The software automatically maps or moves the variable to the compatible empty position that you have selected.

If you have selected several variables:

- If the space in the empty position that you have chosen allows it, variables are mapped or moved contiguously starting from this position.
- If there is not sufficient contiguous space in the empty position that you have chosen to map or move all variables but is sufficient for at least one variable, the software opens the **Mapping Confirmation** dialog box to inform you. You have the following options:

Yes: Maps or moves the variables for which sufficient space is available at the chosen empty position and maps or moves the other ones to the next available position.

No: Maps or moves only the variables for which sufficient space is available at the chosen empty position but does not map or move the other ones. It then opens a dialog box to inform you of which variables it could not map or move. These variables remain in the **Network Variables** or **Variables** section.

Cancel: Cancels the mapping or move operation.

- Cleared: The software opens the **Select Memory Position** dialog box, which allows you to select any available compatible position to map or move the variable within the read or write size that you have defined for the channel.

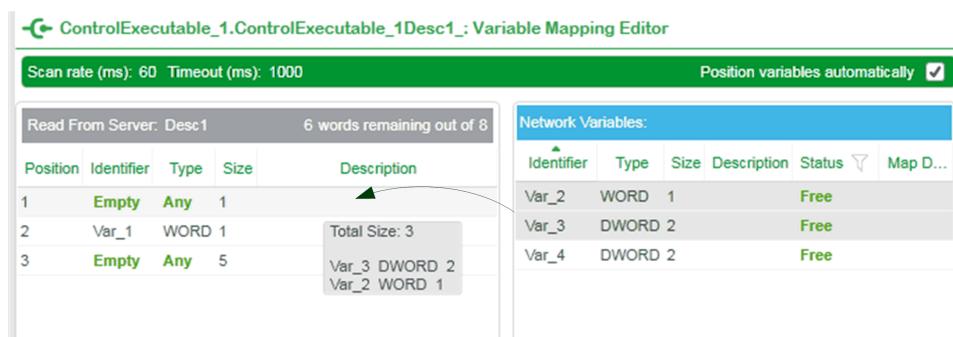
If you have selected several variables, the position that you select in the **Select Memory Position** dialog box applies to the first variable only. The software then maps or moves the other variables in the contiguous space that is available right after the position that you have selected. If not sufficient contiguous space is available, it informs you of which variables it was not able to map or move because of insufficient space.

In both cases, the software verifies that the position satisfies the variable mapping rules based on its size. The **Mapping Confirmation** dialog box opens only if there is empty space in a channel before and after an already mapped variable.

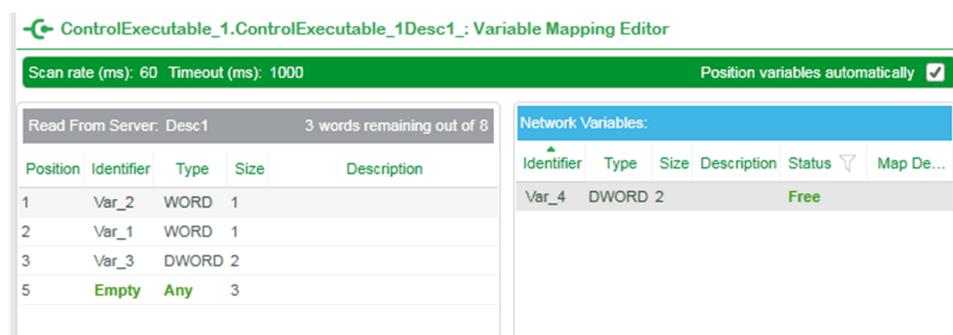
NOTE: You can map 32-bit variables (for example, variables of type REAL or DWORD) only to odd addresses in the **Variable Mapping Editor**. The variables are aligned correctly during the build, page 633 depending on the controller platform.

Example of Variable Mapping Editor Usage

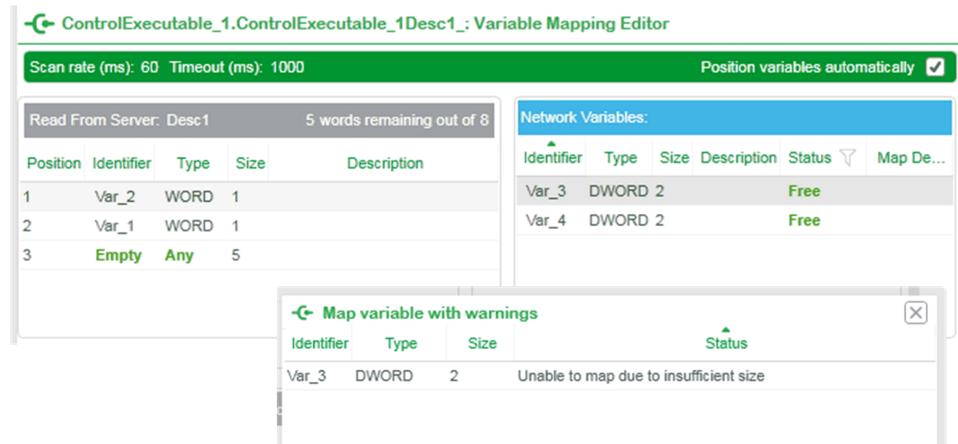
In this example, network variable *Var_1* is already mapped to a peer to peer communication channel and there is one word of free space available before (position 1) and five words available after (position 3). The **Position variables automatically** check box is selected. *Var_2* and *Var_3* are selected and dragged to be mapped at position 1. Because there is no sufficient space to map both variables but only one of them at position 1, the software displays the **Mapping Confirmation** dialog box when you release the mouse button.



If you click **Yes**, the software maps *Var_2* at position 1, which you had selected and *Var_3* at position 3, which is the next available space where the variable fits.



If you click **No**, the software maps only *Var_2* at position 1, which is the only variable out of the two selected ones that fits in this position. It then informs you that *Var_3* is not mapped by opening a dialog box. *Var_3* remains in the **Network Variables** section and is available for mapping.



Variable Mapping Rules

The software applies the following rules when mapping a variable:

- A variable cannot be mapped to a position overlapping with another mapped variable.
- A variable cannot be mapped if:
 - It is split up between two or more positions.
 - It does not fit entirely inside the communication channel.

Mapping Network Variables

Mapping variables to a peer to peer communication channel affects the executables of both the consumer and the counterpart owner projects. To apply changes, you need to start/stop the controllers to which these executables are deployed in the right order; otherwise incorrect data may be read and/or written.

⚠ WARNING

LOSS OF CONTROL

- Deploy the executables to the controllers between which the peer to peer communication channel exists.
- Start/stop controllers in the right order.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To map network variables (to read from the server) to a communication channel, proceed as follows.

Step	Action	Result
1	In the Communication Mapping Editor , right-click the communication channel to which you want to map variables and select Map Variables .	Result: The Variable Mapping Editor opens.
2	Clear the Position variables automatically check box if you want to select the position to map the variable yourself.	Refer to <i>Position of Mapped Variables</i> for details, page 596.
3	In the Network Variables pane, select the variables that you want to map and drag them onto a compatible empty position in the Read from Server pane.	Result: If you have unselected the Map on Top check box, the software opens the Select Memory Position dialog box. Proceed to step 4; otherwise the software:

Step	Action	Result
		<ul style="list-style-type: none"> Maps the variables, page 596 to the available space starting from the position that you have selected and satisfying the mapping rules. Displays them in the Read from Server pane. Removes the variable from the Network Variables pane. Informs you if it was not able to map any variable.
4	In the Select Memory Position dialog box, click the arrow button of the Memory Position field and select a position to map the variable.	-
5	Click OK . NOTE: Click Cancel to close the Select a Memory Position dialog box without mapping the variable.	<p>Result: The software:</p> <ul style="list-style-type: none"> Maps the variables, page 596 to the selected position. Displays them in the Read from Server pane. Removes the variable from the Network Variables pane. Informs you if it was not able to map any variable.

NOTE: Proceed in the same way to map a variable (to write to the server) to the selected communication channel by using the **Variables** and **Write to Server** panes of the **Variable Mapping Editor**.

Changing a Variable Mapping Position

To change the mapping position of a variable, drag it to the new position.

You can select several variables to move them at once.

Moving variables mapped to a peer to peer communication channel affects the executables of both the consumer and the counterpart owner projects. To apply changes, you need to start/stop the controllers to which these executables are deployed in the right order; otherwise incorrect data may be read and/or written.

⚠ WARNING	
LOSS OF CONTROL	
	<ul style="list-style-type: none"> Deploy the executables to the controllers between which the peer to peer communication channel exists. Start/stop controllers in the right order. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>

To change the mapping position of network variables or variables with the **Select a Memory Position** dialog box selected, proceed as follows.

Step	Action	Result
1	Select the variables in the Read from Server or Write to Server pane and drag them to the new compatible empty position.	<p>Result: The software:</p> <ul style="list-style-type: none"> Unmaps the variables from their current positions, which become available for mapping. Maps the variables, page 596 to the available space starting from the new position that you have selected and satisfying the mapping rules. Informs you if it was not able to map any variable.

To change the mapping position of network variables or variables with the **Select a Memory Position** dialog box cleared, proceed as follows.

Step	Action	Result
1	Select the variables in the Read from Server or Write to Server pane and drag them to the new compatible empty position.	Result: The software: <ul style="list-style-type: none"> Opens the Select Memory Position dialog box.
2	In the Select Memory Position dialog box, click the arrow button of the Memory Position field and select a position to map the variable.	-
3	Click OK . NOTE: Click Cancel to close the Select a Memory Position dialog box without mapping the variable.	Result: The software: <ul style="list-style-type: none"> Maps the variables, page 596 to the selected position. Informs you if it was not able to map any variable.

Removing a Variable from a Communication Channel

To remove a mapped variable from the communication channel, unmap it.

Unmapping variables mapped from a peer to peer communication channel affects the executables of both the consumer and the counterpart owner projects. To apply changes, you need to start/stop the controllers to which these executables are deployed in the right order; otherwise incorrect data may be read and/or written.

⚠ WARNING

LOSS OF CONTROL

- Deploy the executables to the controllers between which the peer to peer communication channel exists.
- Start/stop controllers in the right order.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To unmap a network variable or variable, proceed as follows.

Step	Action
1	right-click the variable in the Read from Server or Write to Server pane and select Unmap . Result: The software: <ul style="list-style-type: none"> Unmaps the variable from the selected position, which becomes available for mapping. Removes the variable from the corresponding Read from Server or Write to Server pane. Displays the variable in the corresponding Network Variables or Variables pane. NOTE: You can also right-click the variable and press Del .

Deleting or Changing Mapped Variables

A network variable or variable is unmapped from the communication channel if you delete either:

- The network variable itself.
- The corresponding variable in the owner project.

It is also unmapped from the communication channel if you change any of the following parameters of it, in either the owner or the consumer project:

- Name
- Type (ID or version)

- Size

Corresponding information is displayed in the notification panel.

Mapping Hardware

Overview

To open the **Hardware Mapping Editor**, click the **Hardware Mapping** tab within the **Manage** window, page 578. The tab appears once you have performed the service mapping, page 580 with a controller.

It allows you to map the interfaces of instances of the application to compatible hardware interfaces representing the I/O and communication modules of the entity that you have service mapped to the Control executable of the project.

Application objects that expose mapping interfaces are instances of either:

- HAL templates (for example, `DISignal_UL`).
- Communication port templates (for example, `$EMPortM`).
- Hardware application templates (for example, `$TesysTE` or `$TSignCptBmx`).

The mapping interfaces of application objects appear in the **Hardware Mapping Editor** once you assign their facets to the Control project; even without generating them. They are updated in the mapping editor when you make changes in the application (for example, rename an instance or disable a service) and/or generate the Control project (for example, after setting a facet to **Out Of Date** or **Deleted**).

You can map interfaces to modules that are connected:

- Physically (for example, in-rack I/O modules).
Their topological mapping interfaces are updated in the mapping editor when you save changes, page 511 in the Control Participant **Configuration** window.
- Logically (for example, Modbus TCP Device I/Os, PRMs, or STB islands).
Their topological mapping interfaces appear in the mapping editor when you connect, page 540 the device to the same Ethernet network as the controller.

If you have created several executables, proceed with the hardware mapping for each one of them.

To facilitate the management of many mappings, you can export and import hardware mappings, page 609.

NOTE: The software marks variables and DFB instances, page 639 that are used for the hardware mapping when you build the logical Control Participant project.

Mapping Modbus TCP Devices, PRMs, and STB Islands

For the following entities communicating with the controller through Modbus TCP implicit messaging (I/O scanning), and that are connected to the same Ethernet network as the controller, you can proceed with the hardware mapping:

- Modbus TCP I/O devices: Even without existing communication channels.
- STB islands and PRMs: Only after you create a communication channel, page 582.

NOTE: You need to instantiate the necessary application templates first. For details, refer to the topic describing how to implement communication, page 779.

Mapping Hardware with a Simulator

Hardware mapping is not supported, and not required, for executables that you have mapped to a station node running software emulating a simulator.

Mapping Generic Device Interfaces

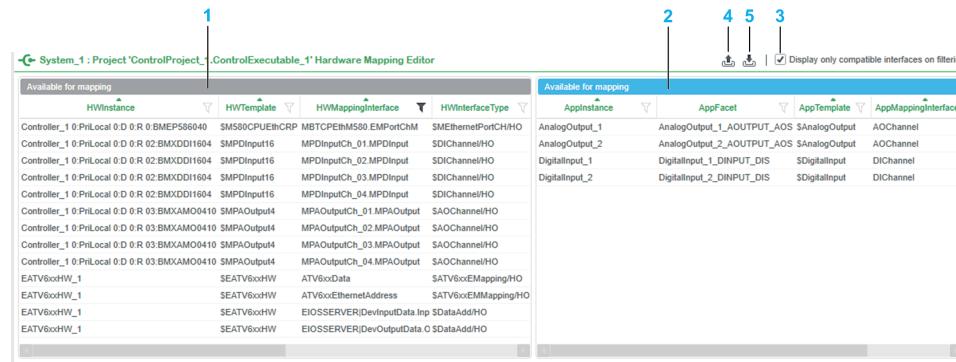
When a topological entity uses the `$EGenericDeviceHW`, `$EIPGenericDeviceHW`, or `$EGenericPBSlave` template to model a device, a generic hardware mapping interface is available for this device in the **Hardware Mapping Editor**. You can use it to map the interface of a project facet to this I/O device. However, because of the generic nature of the interface, the software does not verify if both interfaces are compatible, page 606 before proceeding with the mapping.

Opening the Hardware Mapping Editor in Read-Only Mode

The editor can open in read-only mode, for example, if the executable is locked by another process such as the configuration of the controller. In this case, hardware mapping creation, modification, and import are not allowed.

Hardware Mapping Editor

The following figure shows an example of the **Hardware Mapping Editor** window.



Item	Description
1	Topological devices pane (left, gray title bar), which displays the interfaces of the controller entity that you have mapped to the executable of the Control project. It also displays interfaces of Modbus TCP/EtherNet/IP device, PRM, and STB island entities that are connected to the same Ethernet network as the controller. A communication mapping for PRM and STB island entities needs to exist.
2	Project facet pane (right, blue title bar), which displays the mapping interfaces of the facets that you have assigned to the Control project, including interfaces of facets of Modbus TCP/EtherNet/IP explicit messaging device instances, Modbus serial device instances, and the corresponding communication port instances.
3	Check box to display only compatible interfaces, page 604 that meet the selected filter criteria in both panes. The functionality helps locate compatible interfaces when you have many different ones in either pane. The functionality is enabled by default.
4	Lets you export, page 609 to file (.csv) the existing hardware mappings and the interfaces of project facets that have not been mapped yet.
5	Lets you import, page 609 valid and compatible hardware mappings that you have created outside the Hardware Mapping Editor and that are contained in a file (.csv).

NOTE: When the executable of the Control project is mapped to a redundant controller, the topological devices pane displays only the hardware interfaces of the primary controller.

Editor section	Column header	Description
Topological devices pane	HWInstance	<p>Identifier of the topological device.</p> <p>Displays:</p> <ul style="list-style-type: none"> Controller devices. Modbus TCP and EtherNet/IP devices that are physically connected to the same network as the controller. PRM entities and STB island devices after you have completed the communication mapping. <p>NOTE: The designation of devices is described in the topic showing the representation of topological entities, page 512.</p>
	HWTtemplate	Identifier of the template that is used by the topological device.
	HWMapping Interface	Identifier of the facet that models the hardware channel followed by the identifier of the facet template.
	HWInterfaceType	Type of the mapping interface and role.
	AppInstance	These columns are blank before you start the mapping process.
	AppFacet	The software moves the information contained in the corresponding columns of the project facet pane to these columns when you map the project facet interface.
	AppTemplate	
	AppMapping Interface	
Project facet pane	AppInterface-Type	
	AppInstance	Identifier of the instance that references the facet.
	AppFacet	Identifier of the facet that is assigned to the Control project.
	AppTemplate	Identifier of the template that is used by the instance.
	AppMapping Interface	Identifier of the hardware mapping interface that is exposed by the facet.
	AppInterface-Type	<p>Type of the mapping interface and role.</p> <p>The prefix of the type indicates the type of I/O hardware channel:</p> <ul style="list-style-type: none"> DI: Digital input DO: Digital output AI: Analog input AO: Analog output

Customizing Column Display

In the various panes of the editor, you can choose which columns you want to see by right-clicking a column header and selecting **Customize**. Certain columns are always shown.

When you hide a column for which a filter is applied, the filter is cleared.

Displaying Compatible Interfaces

To show only interfaces that are compatible in both panes, proceed as follows.

Step	Action
1	Verify that the Display only compatible interfaces on filtering check box, page 603 is selected.
2	<p>In either pane, click the filter icon of the column that you want to use to filter interfaces.</p> <p>Result: The filter menu, page 118 opens.</p>

Step	Action
3	<p>Enter criteria to be used as filter and click Filter.</p> <p>Result:</p> <ul style="list-style-type: none"> In the pane in which you have applied the filter, only interfaces that meet the filter criteria are shown. In the other pane, a filter is automatically applied to the *InterfaceType column to show only interfaces that are compatible with those that result of the filter that you have applied.
4	You can apply additional filters in either pane to further refine results.

NOTE:

- When you clear the filter in either pane, a filter is applied to the ***InterfaceType** column in the other pane to show only interfaces that are compatible with those displayed in the pane where you have cleared the filter.
- To clear filters in both panes, clear the **Display only compatible interfaces on filtering** check box.

The following example illustrates the use of the automatic filter mechanism to display only compatible interfaces. The figures shows only a partial view of the **Hardware Mapping Editor** to simplify the example.

HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_12 MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_13 MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_14 MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_15 MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_16 MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPMModErr.ModuleStatus	SModuleDiagMapping/HO
Controller_1 0 PrILocal 0 D 0: SMPAOutput4		MPAOutputCh_01.MPAOutput	SAOChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPAOutput4		MPAOutputCh_02.MPAOutput	SAOChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPAOutput4		MPAOutputCh_03.MPAOutput	SAOChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPAOutput4		MPAOutputCh_04.MPAOutput	SAOChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPAOutput4		MPMModErr.ModuleStatus	SModuleDiagMapping/HO
EATVboxxHW_1	SEATVboxxHW	ATVboxxData	SATVboxxMapping/HO
EATVboxxHW_1	SEATVboxxHW	ATVboxxEthernetAddress	SATVboxxEMMapping/HO
EATVboxxHW_1	SEATVboxxHW	EIGENPDUERUN	SDIChannel/HO

AppInstance	AppFacet	AppTemplate	AppMappingInterface
AnalogOutput_1	AnalogOutput_1_AOUTPUT_AOS	AnalogOutput	AOChannel
AnalogOutput_2	AnalogOutput_2_AOUTPUT_AOS	AnalogOutput	AOChannel
DigitalInput_1	DigitalInput_1_DINPUT_DIS	SDigitalInput	DIChannel
DigitalInput_2	DigitalInput_2_DINPUT_DIS	SDigitalInput	DIChannel

Item	Description
1	A controller is configured with an analog input, an analog output, and a digital input module. The corresponding interfaces appear in the HWInterfaceType column of the topological devices pane.
2	Facets of analog input, digital input and digital output templates are assigned to the project and the template identifiers appear in the AppTemplate column of the project facet pane.

In this step, the objective is to show only topological interfaces that are compatible with digital input interfaces to perform the hardware mapping between the two.

HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_01.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_02.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_03.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_04.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_05.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_06.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_07.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_08.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_09.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_10.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_11.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_12.MPDInput	SDIChannel/HO
Controller_1 0 PrILocal 0 D 0: SMPDInput16		MPDInputCh_13.MPDInput	SDIChannel/HO

AppInstance	AppFacet	AppTemplate	AppMappingInterface
DigitalInput_1	DigitalInput_1_DINPUT_DIS	SDigitalInput	
DigitalInput_2	DigitalInput_2_DINPUT_DIS	SDigitalInput	

Item	Description
1	Verify that the check box, page 603 is selected.
2	The AppTemplate column, a filter was applied to show only interfaces of digital input templates. One interface meets this criteria.
3	In the topological devices pane, the software has automatically applied a filter to the HWInterfaceType column to show only interfaces of digital input modules. Interfaces of other modules are not shown anymore. You can map the interface of the project facet to any of these topological input channels.

In this step, the objective is to show only project facet interfaces that are compatible with the topological interfaces of the modules present in the controller configuration. This can be helpful to verify if any hardware-compatible project facet interfaces have not yet been mapped.

HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType
Controller_1_0.PriLocal.0.D.0: SMPDInput16		MPDInputCh_12.MPDInput	SDIChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPDInput16		MPDInputCh_13.MPDInput	SDIChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPDInput16		MPDInputCh_14.MPDInput	SDIChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPDInput16		MPDInputCh_15.MPDInput	SDIChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPDInput16		MPDInputCh_16.MPDInput	SDIChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPDInput16		MPMDODErr.ModuleStatus	SModuleDiagMapping/HO
Controller_1_0.PriLocal.0.D.0: SMPAOutput		MPAOutputCh_01.MPAOutput	SAOChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPAOutput		MPAOutputCh_02.MPAOutput	SAOChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPAOutput		MPAOutputCh_03.MPAOutput	SAOChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPAOutput		MPAOutputCh_04.MPAOutput	SAOChannel/HO
Controller_1_0.PriLocal.0.D.0: SMPAOutput		MPMDODErr.ModuleStatus	SModuleDiagMapping/HO
EATV6x0HW_1	SEATV6x0HW	ATV6x0Data	SATV6x0EMapping/HO
EATV6x0HW_1	SEATV6x0HW	ATV6x0EthernetAddress	SATV6x0EMMapping/HO

AppInstance	AppFacet	AppTemplate	AppMap...	ApplInterfaceType
AnalogOutput_1		AnalogOutput_1_AOUTPUT	\$AnalogOutput	AOChannel
AnalogOutput_2		AnalogOutput_2_AOUTPUT	\$AnalogOutput	AOChannel
DigitalInput_1		DigitalInput_1_DINPUT_DIS	\$DigitalInput	DIChannel
DigitalInput_2		DigitalInput_2_DINPUT_DIS	\$DigitalInput	DIChannel

Item	Description
1	In the topological devices pane, the filter that the software had automatically applied to the HWInterfaceType column was cleared by the user. The column now shows all the interfaces of the various modules of the controller to which you can map project facet interfaces.
2	In the project facet pane, the software has cleared the filter of the AppTemplate column. Instead, it has applied a filter to the ApplInterfaceType column to show only interfaces that are compatible with the topological interfaces that are exposed by the configuration. As a result, the interface of the digital output template is not shown anymore because the configuration does not contain a digital output module.
3	The check box must be selected.

Mapping Hardware

To map facets of the Control project to topological interfaces of an entity, proceed as follows.

Step	Action
1	<p>In the right-hand project facet pane, locate and select the facets of the Control project that you want to map. Use filters if necessary.</p> <p>Result: The selected facets are highlighted.</p>
2	<p>In the left hand, topological device pane, locate the topological interfaces to which you want to map the facets that you have selected in step 1. Use filters if necessary.</p> <p>Result: The selected instance is highlighted.</p> <p>NOTE: For the mapping to succeed, the type of the interface that is displayed in both *InterfaceType columns needs to be identical and the interface roles need to be compatible.</p> <p>For example, interface type \$DI/Channel with roles HO (Hardware Object) and SO (Signal Object).</p>
3	<p>From the project facet pane, drag the selected project facets onto the topological interfaces.</p> <p>Result: The software shifts the information of the project facets to the respective columns of the topological device pane.</p> <p>NOTE: If the interfaces do not match, a tooltip opens and they are not mapped.</p>

NOTE: To map an already mapped project facet interface to a different topological device with matching interface, drag the mapped project facet onto another compatible topological device interface.

Deleting Hardware Mappings

To delete the mapping of one or more already mapped project facet interfaces, proceed as follows.

Step	Action
1	<p>In the topological device pane, select the rows that correspond to the mappings.</p>
2	<p>Proceed either way:</p> <ul style="list-style-type: none"> • Drag the selection back to the project facet pane. • Press Del and confirm the command. • Right-click the selection, click Unmap, and confirm the command. <p>Result: The selected facet appear again in the project facet pane.</p> <p>NOTE: To delete all the mappings at once, click inside the topological device pane, press Ctrl+A, and use Del or Unmap.</p>

Mapping Several Project Facet Interfaces Simultaneously

You can drag more than one project facet interface onto the topological device pane at once by using multiple selection. The software maps the project facet interfaces to contiguous topological interfaces in the order they are displayed in the tooltip while dragging them.

The interfaces of each project facet/topological device pair need to match to complete the mapping process.

If there is a missing or incompatible topological interface in between, the first matching interfaces are mapped but not the remaining ones, which appear again in the right-hand project facet pane. A message informs you of the project facet interfaces that could not be mapped.

NOTE: Use the filters to display topological devices in the desired sequence.

Changing the Application and Projects After Mapping

If you make any of the following changes to a facet that you have already mapped by using the **Hardware Mapping Editor**, the software deletes the mapping and makes the facet available again for mapping:

- Unassigning, page 378 the facet and generating the project or section.

- Deleting the instance referencing the facet.
- Unselecting the element, page 176 that corresponds to the facet.

Changing the Topology After Mapping

If you make any of the following changes to topological devices whose interfaces you have already mapped by using the **Hardware Mapping Editor**, the software may delete the mapping. In such case, it makes the corresponding project facets available again for mapping:

- Deleting the controller entity.
- Changing the CPU module of the controller entity.
- Deleting the I/O device.
- Moving the I/O device.
- Deleting the communication channel with an STB island or a PROFIBUS PRM Master.
- Updating templates of topological entities.

Changing Hardware Mapping After Deploying

Consider a Control project that contains one or more facets representing signals that need to be mapped to I/O modules.

You build and deploy the Control project but without doing these hardware mappings.

If you then do the hardware mapping, build the Control project, and select the **Deploy changes / Undo Online Changes** command to deploy the hardware mapping change, page 682, the new initial value of the variables that correspond to the mapped signals are deployed.

However, the current value of these variables remains unchanged in the deployed Control project.

The change becomes effective, either through a cold start of the controller or by replacing the current value of the variables in the animation table by their new initial value.

For example, you deploy a Control project without mapping the facet interface representing the signal quality of an *AnalogInput* instance to the facet interface representing the analog input module channel of the controller.

If you refine the deployed Control project online, the initial value of the variable assigned to the **ComStatus** pin of the corresponding DFB is displayed as *False* because the hardware mapping is not done. Its current value is *0*.

If you proceed with the hardware mapping, build the Control project, and execute the **Deploy changes / Undo Online Changes** command, the initial value of this variable is changed to *True*.

However, its current value remains *0*.

NOTE: The same applies when you change an existing hardware mapping after deploying it and deploy it, using the **Deploy changes / Undo Online Changes** command.

Exporting and Importing Hardware Mappings

Overview

The functionality lets you manage, in a three-step process, a large number of hardware mappings, page 602 by:

- Exporting the data that is shown in the **Hardware Mapping Editor** to comma-separated data format (CSV).
- Editing the file so that it contains the hardware mappings that you want to create or modify.
- Importing the file into the **Hardware Mapping Editor** of the same or a different Control executable.

Before proceeding with the import, the software validates the data and informs you of invalid entries, which will not be imported.

Exporting Hardware Mapping Data

Overview

In a first step, the export functionality lets you create export files that contain the data of the **Hardware Mapping Editor** for a given Control executable.

Good Practices

- Before modifying hardware mappings, back up all the existing hardware mappings by exporting them. If needed, it lets you restore the original configuration by importing the file.
- Apply a filter before exporting so that the export file contains only the topological interfaces whose hardware mapping you want to create or modify.

Export File Compatibility

Hardware mapping export files are forward compatible unless mentioned otherwise.

Export File Description

When you export hardware mapping data, two export files are created. Each one contains data in a comma-separated, page 228 format (CSV).

By default, export files are saved at the path %ProgramData%\Schneider Electric\Process Expert x\Db, where x represents the software version. When this location does not exist (for example, if only clients are installed on the computer), files are saved at either location:

- At the path where you have last exported or imported any file of EcoStruxure Process Expert.
- If the above does not apply or is not reachable, in the Windows® **Documents** folder.

Default file name	Description
<Executable identifier>_HWMapping where _HWMapping is appended to the file name that you enter once the file is created.	<p>The file contains the same data as shown in the topological devices pane, page 603 of the Hardware Mapping Editor at the time of export.</p> <p>It is a snapshot of:</p> <ul style="list-style-type: none"> • The mapping interfaces of the topological entity that is mapped to the Control executable (columns A to D⁽¹⁾). • The mapping interfaces of application objects that are already mapped (columns E to I⁽¹⁾). <p>If you have applied a filter, page 604 before exporting, only the filtered data is exported.</p> <p>The first row contains the column headers of the editor.</p> <p>You need to edit the content, page 612 of this file to create, modify, or delete hardware mappings.</p>
<Executable identifier>_Pending where _Pending is appended to the file name that you enter once the file is created.	<p>This supporting file contains the same data as shown in the project facet pane, page 603 of the Hardware Mapping Editor at the time of export.</p> <p>It is a snapshot of the mapping interfaces of application objects that are available for mapping.</p> <p>If you have applied a filter, page 604 before exporting, only the filtered data is exported.</p> <p>The first row contains the column headers of the pane in the editor.</p> <p>You can use this file to perform cut-and-paste operations to move to the <Executable identifier>_HWMapping.csv file the data that corresponds to hardware mappings that you want to create.</p>

(1) When you open the file by using Microsoft® Excel®

Example of Export Files

The following figure shows an example of the **Hardware Mapping Editor** where:

- In the topological devices pane, four interfaces of digital input project facets are mapped to the corresponding hardware channel interfaces.
- In the project facet pane, four interfaces of analog output project facets are shown as available for hardware mapping.

The screenshot shows the 'Hardware Mapping Editor' interface with two main sections:

- Available for mapping:** This pane lists hardware instances and their corresponding software project facets. It includes columns for HWInstance, HWTemplate, HWMappingInterface, HWInterfaceType, AppInstance, AppFacet, AppTemplate, AppMappingInterface, and AppInterfaceType. The data shows various digital input mappings from Controller_1 to hardware ports like MPDInputCh_01 to MPDInputCh_04.
- Applicable for mapping:** This pane lists available analog output project facets. It includes columns for AppInstance, AppFacet, AppTemplate, AppMappingInterface, and AppInterfaceType. The data shows four analog output facets: AnalogOutput_1 to AnalogOutput_4.

The following figure shows the content of the exported hardware mapping data file (`*_HWMapping.csv`) that corresponds to the previous example.

The screenshot shows a CSV file named 'ControlExecutable_1.HWMapping' with the following data:

	A	B	C	D	E	F	G	H	I
1	HWInstance		HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
2	Controller_1 0:PrlLocal 0:D 0:R 0:BMEPS586040	SMSB0CPUetherCRP	MBTCPETHMSB0.EMPortCHM	SMEthernetPortCH/HO					
3	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXD01604	MPDInputCh01	MPDInputCh_01.MPDInput	SDIChannel/HO	DigitalInput_1	DigitalInput_1.DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO
4	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXD01604	MPDInputCh02	MPDInputCh_02.MPDInput	SDIChannel/HO	DigitalInput_2	DigitalInput_2.DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO
5	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXD01604	MPDInputCh03	MPDInputCh_03.MPDInput	SDIChannel/HO	DigitalInput_3	DigitalInput_3.DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO
6	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXD01604	MPDInputCh04	MPDInputCh_04.MPDInput	SDIChannel/HO	DigitalInput_4	DigitalInput_4.DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO
7	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXD01604	MPMODerModuleStatus	MPMODer.ModuleStatus	SModuleDiagMapping/HO					
8	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXAM00410	MPAOutputCh01	MPAOutputCh_01.MPAOutput	SAOChannel/HO					
9	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXAM00410	MPAOutputCh02	MPAOutputCh_02.MPAOutput	SAOChannel/HO					
10	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXAM00410	MPAOutputCh03	MPAOutputCh_03.MPAOutput	SAOChannel/HO					
11	Controller_1 0:PrlLocal 0:D 0:R 0:BMAXAM00410	MPAOutputCh04	MPAOutputCh_04.MPAOutput	SAOChannel/HO					

And the following figure shows the content of the supporting file (`*_Pending.csv`).

The screenshot shows a CSV file named 'ControlExecutable_1.Pending' with the following data:

	A	B	C	D	E
1	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
2	AnalogOutput_1	AnalogOutput_1.AOUTPUT_AOS	\$AnalogOutput	AOChannel	SAOChannel/SO
3	AnalogOutput_2	AnalogOutput_2.AOUTPUT_AOS	\$AnalogOutput	AOChannel	SAOChannel/SO
4	AnalogOutput_3	AnalogOutput_3.AOUTPUT_AOS	\$AnalogOutput	AOChannel	SAOChannel/SO
5	AnalogOutput_4	AnalogOutput_4.AOUTPUT_AOS	\$AnalogOutput	AOChannel	SAOChannel/SO

Exporting Hardware Mapping Data

To export the data shown in the **Hardware Mapping Editor**, proceed as follows.

Step	Action
1	In the Hardware Mapping Editor , select the data that you want to export by using the filters, page 118.
2	Click the export button, page 603. Result: The Save As dialog box opens.
3	Enter a file name, select a location, and click Save . Result: The two export files are created. Note: The export files do not contain the Control project and the system identifiers. If required, include this information in the file name.

Editing the Hardware Mapping Export File

Overview

In this second step, the objective is to edit the exported hardware mapping data file (*_HWMapping.csv) so that it contains the hardware mappings that you want to create or modify for a Control executable.

You can use the supporting file (*_Pending.csv) which contains the project facet interfaces that are available for mapping.

Schneider Electric recommends using a spreadsheet editor to facilitate the editing of data contained in the export files.

NOTE: Although you can delete hardware mappings by using the import functionality, you can delete them directly in the **Hardware Mapping Editor** by using the **Unmap** command, page 607.

Good Practices

- Before modifying hardware mappings, back up all the existing hardware mappings by exporting them. If needed, it lets you restore the original configuration by importing the file.
- Edit only an up-to-date export file to avoid data conflicts.
- Verify that the list separator format, page 228 is set to comma on the computer if you are using a spreadsheet editor.
- When you use data from the supporting file to create hardware mappings, cut the data rather than copying it to help avoid mapping the same project facet interface several times.

The other way around, add to the supporting file project facet interfaces that you are unmapping.

Compatibility With Hardware Mapping Files of Earlier Versions

You can use hardware mapping data files (topologyexport.csv and supporting file projectexport.csv) that have been exported by using the system server console of earlier versions of the software if you edit the column headers to match the current format.

The table describes which column headers of the topologyexport.csv file need to be updated and their new value (not case-sensitive).

Column ⁽¹⁾	Header	Change to
A	Instance	HWInstance
B	Template	HWTtemplate
C	HW Mapping IF	HWMappingInterface
D	HW Mapping Type	HWInterfaceType
E	Instance	AppInstance
F	Project Facet	AppFacet
G	Facet Type	AppTemplate
H	HW Interface	AppMappingInterface
I	HW Mapping Type	AppInterfaceType

(1) When you open the file by using Microsoft® Excel®

Editing the Hardware Mapping Data File

Edit only rows that correspond to topological interfaces whose hardware mapping you want to create, modify, or delete. Refer also to the examples, page 614.

Entries that exist in the **Hardware Mapping Editor** but not in the hardware mapping data file (for example, because a filter was applied before exporting or because they have been deleted from the file) are not modified.

The tables describe the result of various editing actions in the hardware mapping data file, page 610 (*_HWMapping.csv) once you import the file.

Editing to create, modify, and delete hardware mappings:

Edited data	Action	Result after import
Rows of columns E to I ⁽¹⁾	New, compatible ⁽²⁾ project facet interface data is added by cutting and pasting it from the *_Pending.csv file.	The corresponding hardware mapping is created. The hardware mapping is valid if the values of the two *InterfaceType columns match (same interface type and compatible roles).
	Existing project facet interface data is modified appropriately. For example: <ul style="list-style-type: none"> • By mapping another project facet interface to the topological interface (the data is cut and pasted from the *_Pending.csv file). • By swapping two mapped project facet interfaces. 	The corresponding hardware mapping is modified.
	Existing project facet interface data is deleted.	The corresponding hardware mapping is deleted.

(1) When you open the file by using Microsoft® Excel®

Other editing:

Edited data	Action	Result after import
Rows of columns A to D ⁽¹⁾	Existing topology interface data is deleted.	If columns E to I ⁽¹⁾ for the same row: <ul style="list-style-type: none"> • Are empty, no impact. • This lets you remove from the file rows that you do not need to work with. • Contain data, an invalid entry is reported.
The first row	The first row is deleted.	You cannot import the file.
	A header of columns A to E ⁽¹⁾ is deleted or modified.	
Other rows	A row is not edited.	No impact given the same row is still present in the Hardware Mapping Editor .
	An entire row is deleted.	No impact. NOTE: This lets you remove from the file rows that you do not need to work with.
	A row containing a string is added. Enter the string in column A ⁽¹⁾ and start with a semi-colon (for example, ;My comment).	No impact. NOTE: This lets you add comments to the file.
	An empty row is added.	No impact.
	The order of complete rows is changed.	
	The topology or project facet interface data (or both) of a row is duplicated.	You cannot import the original hardware mapping nor its copy.
Any columns	A column is added in first position (A ⁽¹⁾) or between columns A to E ⁽¹⁾ .	You cannot import the file.
	One or more columns are added after the last column (E ⁽¹⁾).	No impact. NOTE: This lets you add comments at the end of a row.
	The order of columns A to E ⁽¹⁾ is modified.	No impact.

(1) When you open the file by using Microsoft® Excel®

Example of Editing to Create Hardware Mappings

The following example shows how to create hardware mappings with two channels of an analog output module by pasting the corresponding project facet interface data of the supporting file (shown outlined) to the hardware mapping data file.

The following figure shows the result in the topological devices pane of the **Hardware Mapping Editor**. Also, the corresponding analog output project facet interfaces have been removed from the project facet pane of the editor (not shown).

Available for mapping								
HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
Controller_1_0@P1Local 0 D 0: S5M80CUEBICRPMTCBPEIhMS80.EmpfChM		SMEBmemPortCH/HIO						
Controller_1_0@P1Local 0 D 0: SMPDInp16		MPDInpCh_01.MPDInp16	SDIChannel/HIO	DigitalInput_1	DigitalInput_1	DINPUT_DIS	DigitalInput	DIChannel
Controller_1_0@P1Local 0 D 0: SMPDInp16		MPDInpCh_02.MPDInp16	SDIChannel/HIO	DigitalInput_2	DigitalInput_2	DINPUT_DIS	DigitalInput	DIChannel
Controller_1_0@P1Local 0 D 0: SMPDInp16		MPDInpCh_03.MPDInp16	SDIChannel/HIO	DigitalInput_3	DigitalInput_3	DINPUT_DIS	DigitalInput	DIChannel
Controller_1_0@P1Local 0 D 0: SMPDInp16		MPDInpCh_04.MPDInp16	SDIChannel/HIO	DigitalInput_4	DigitalInput_4	DINPUT_DIS	DigitalInput	DIChannel
Controller_1_0@P1Local 0 D 0: SMPDInp16		MPMODeH.ModuleStatus	SModuleDiagMapping/HIO					
Controller_1_0@P1Local 0 D 0: SMPAOOutput4		MPAOutputCh_01.MPAOutput4	SAOChannel/HIO	AnalogOutput_1	AnalogOutput_1	_AOUTPUT_	SAAnalogOutput	AOChannel
Controller_1_0@P1Local 0 D 0: SMPAOOutput4		MPAOutputCh_02.MPAOutput4	SAOChannel/HIO	AnalogOutput_2	AnalogOutput_2	_AOUTPUT_	SAAnalogOutput	AOChannel
Controller_1_0@P1Local 0 D 0: SMPAOOutput4		MPAOutputCh_03.MPAOutput4	SAOChannel/HIO					
Controller_1_0@P1Local 0 D 0: SMPAOOutput4		MPAOutputCh_04.MPAOutput4	SAOChannel/HIO					

Example of Editing to Delete Hardware Mappings

Continuing with the previous example, the figure shows how to delete hardware mappings with two channels of a digital input module by deleting the corresponding project facet interface data (shown outlined) from the hardware mapping data file.

HWInstance	A	B	C	D	E	F	G	H	I
	HWTemplate	HWMappingInterface	HWInterfaceType	D	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
2	Controller_1@P1Loco1_0@D_0@R-0.8MEPS586040	SMS300ModC_PMPB100ModSMS300ModC_PMPB100Mod	MgmtHcmCh	SMHmmedPortCh/HO					
3	Controller_1@P1Loco1_0@D_0@R-0.8MAMX011604	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16
4	Controller_1@P1Loco1_0@D_0@R-0.8MAMX011604	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16
5	Controller_1@P1Loco1_0@D_0@R-0.8MAMX011604	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16
6	Controller_1@P1Loco1_0@D_0@R-0.8MAMX011604	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16
7	Controller_1@P1Loco1_0@D_0@R-0.8MAMX011604	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16	SMODP16Mod16
8	Controller_1@P1Loco1_0@D_0@R-0.8MAMX00401404	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04
9	Controller_1@P1Loco1_0@D_0@R-0.8MAMX00401404	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04
10	Controller_1@P1Loco1_0@D_0@R-0.8MAMX00401404	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04
11	Controller_1@P1Loco1_0@D_0@R-0.8MAMX00401404	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04
12	Controller_1@P1Loco1_0@D_0@R-0.8MAMX00401404	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04	SMPAOutput04
		PMMD0ER.ModuleStatus	SModuleDiagMapping/HO						

NOTE: It is recommended to cut the project facet interface data and paste it in the supporting file to keep track of project facet interfaces that are available for mapping.

The following figure shows the result in the topological devices pane of the **Hardware Mapping Editor**. Also, the corresponding project facet interfaces have been added to the project facet pane of the editor (not shown).

Available for mapping								
HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
Controller_0_0_PtLocal_0_0_D_0_0_SMS80CUEB1CRP	MTCBPElthMS80_EMPTChM	SMBEthernetPortCH/R0						
Controller_0_0_PtLocal_0_0_D_0_0_SMPDInput16	MPDInputCh_01MPDInput	SDIChannel/H0						
Controller_0_0_PtLocal_0_0_D_0_0_SMPDInput16	MPDInputCh_02MPDInput	SDIChannel/H0						
Controller_0_0_PtLocal_0_0_D_0_0_SMPDInput16	MPDInputCh_03MPDInput	SDIChannel/H0	DigitalInput_3	DigitalInput_3_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/S0	
Controller_0_0_PtLocal_0_0_D_0_0_SMPDInput16	MPDInputCh_04MPDInput	SDIChannel/H0	DigitalInput_4	DigitalInput_4_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/S0	
Controller_0_0_PtLocal_0_0_D_0_0_SMPDInput16	NPIODMod_ErrModuleStatus	SModuleDiagMapping/H0						
Controller_0_0_PtLocal_0_0_D_0_0_SMPAOOutput4	MPAOutputCh_01.MPAOutput	SAOChannel/H0	AnalogOutput_1	AnalogOutput_1_AOUTPUT_	AnalogOutput	AOChannel	SAOChannel/S0	
Controller_0_0_PtLocal_0_0_D_0_0_SMPAOOutput4	MPAOutputCh_02.MPAOutput	SAOchannel/H0	AnalogOutput_2	AnalogOutput_2_AOUTPUT_	AnalogOutput	AOchannel	SAOChannel/S0	
Controller_0_0_PtLocal_0_0_D_0_0_SMPAOOutput4	MPAOutputCh_03.MPAOutput	SAOchannel/H0						
Controller_0_0_PtLocal_0_0_D_0_0_SMPAOOutput4	MPAOutputCh_04.MPAOutput	SAOchannel/H0						

Example of Editing to Modify Hardware Mappings

Continuing with the previous example, the figure shows how to move an existing hardware mapping with the channel of an analog output module (*Ch_02*) to another channel (*Ch_01*). This is achieved by moving the corresponding project facet interface data of *AnalogOutput_2* to the other topological interface, thus overwriting the original interface data of *AnalogOutput_1*.

A	B	C	D	E	F	G	H	I
HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
Controller_1_0_PriLocal 0:D:0:R:0:BMPE586040	SM5860CPUETHCRP	M8TCPEBhM580_EMPortChM	SMEthernetPortCh/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_01.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_02.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_03.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_04.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_05.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_06.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_07.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_01.MPAOutput	SAOChannel/HO	AnalogOutput_1	AnalogOutput_1_AOUTPUT_AOS	AnalogOutput	AOChannel	SAOChannel/SO
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_02.MPAOutput	SAOChannel/HO	AnalogOutput_2	AnalogOutput_2_AOUTPUT_AOS	AnalogOutput	AOChannel	SAOChannel/SO
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_03.MPAOutput	SAOChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_04.MPAOutput	SAOChannel/HO					

NOTE: Before overwriting the project facet interface data of *AnalogOutput_1*, it is recommended to cut the data and paste it in the supporting file to keep track of project facet interfaces that are available for mapping.

The following figure shows the result in the topological devices pane of the **Hardware Mapping Editor**. The interface of *AnalogOutput_2* is now mapped to *Ch_01*. The project facet interface of *AnalogOutput_1* has been unmapped and added to the project facet pane of the editor (not shown).

System_1 : Project 'ControlProject_1.ControlExecutable_1' Hardware Mapping Editor								
HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SM5860CPUETHCRP	M8TCPEBhM580_EMPortChM	SMEthernetPortCh/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_01.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_02.MPDInput	SDIChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_03.MPDInput	SDIChannel/HO	DigitalInput_3	DigitalInput_3_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_04.MPDInput	SDIChannel/HO	DigitalInput_4	DigitalInput_4_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO
Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPMODER_ModuleStatus	SModuleDiagMapping/HO						
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_01.MPAOutput	SAOChannel/HO	AnalogOutput_2	AnalogOutput_2_AOUTPUT_S	AnalogOutput	AOChannel	SAOChannel/SO
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_02.MPAOutput	SAOChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_03.MPAOutput	SAOChannel/HO					
Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_04.MPAOutput	SAOChannel/HO					

Example of Editing to Add Comments to the Export File

The following figure illustrates the various ways to add comments (shown in bold) to the hardware mapping data file (*.*HWMapping.csv*) so that they are not detected as invalid entries. Comments are not imported.

A	B	C	D	E	F	G	H	I	J	K
HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInterface	AppInterfaceType	Comments in columns after I	Comment header
1										
2										
3	You can add comments as rows - 1									
4	Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SM5860CPUETHCRP	M8TCPEBhM580_EMPortChM	SMEthernetPortCh/HO						
5	Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_01.MPDInput	SDIChannel/HO	DigitalInput_1	DigitalInput_1_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO	Comment 1
6	Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_02.MPDInput	SDIChannel/HO	DigitalInput_2	DigitalInput_2_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO	Comment 2
7	Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_03.MPDInput	SDIChannel/HO	DigitalInput_3	DigitalInput_3_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO	
8	Controller_1_0_PriLocal 0:D:0:R:0:2:BMKD01604	SMPDInput16	MPDInputCh_04.MPDInput	SDIChannel/HO	DigitalInput_4	DigitalInput_4_DINPUT_DIS	DigitalInput	DIChannel	SDIChannel/SO	
9	You can add comments as rows - 2									
10	Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_01.MPAOutput	SAOChannel/HO						
11	Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_02.MPAOutput	SAOChannel/HO						
12	Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_03.MPAOutput	SAOChannel/HO						
13	Controller_1_0_PriLocal 0:D:0:R:0:3:BMXAM00410	SMPAOutput4	MPAOutputCh_04.MPAOutput	SAOChannel/HO						

NOTE: The CSV format does not allow saving special formatting.

Importing Hardware Mappings

Overview

In this third step, the data contained in the hardware mapping data file (`*_HWMapping.csv`) is imported into the topological devices pane **Hardware Mapping Editor** that is open for the Control executable.

Each row in the hardware mapping data file is compared to the same row in the **Hardware Mapping Editor** and if different, triggers an action that corresponds to the editing, page 612 that you have performed. Rows in the editor that are not present in the hardware mapping data file remain unchanged.

You can import a file into the same or a different system, Control, or executable as long as the prerequisites are fulfilled.

Prerequisites

To be able to import hardware mapping data, the following conditions must be fulfilled:

- You must have performed the service mapping, page 580 for the Control executable.
- Hardware mapping data files that were created by using earlier versions of the software must be compatible, page 612.
- Both the mapping interfaces of the topological entity and of the application objects, page 610 that are in the hardware mapping data file must exist when you open the **Hardware Mapping Editor** for the Control executable. Additional interfaces of both types can appear in the editor and they have no effect on the import.

Invalid Hardware Mapping Log File

When you attempt to import a hardware mapping data file and the software detects invalid mappings, it creates a log file with these mappings.

- The file is located in the folder from where you have imported the hardware mapping data file.
- It has the same name as the imported file but with the `_log` suffix and .txt file extension.

Thereafter, each time you try to import the same hardware mapping data file (same file name), invalid mappings that the software detects are added to the log file.

Importing Hardware Mapping Data

To import hardware mapping data in the **Hardware Mapping Editor**, proceed as follows.

Step	Action
1	<p>In the Hardware Mapping Editor, click the import button, page 603.</p> <p>Result: The Open dialog box opens.</p>
2	<p>Select the file (<code>*_HWMapping.csv</code>) that contains the hardware mapping data and click Open.</p> <p>Result: The Import Mappings dialog box opens.</p>
3	<p>Click Yes.</p> <p>Result: If the hardware mapping data is valid, the corresponding hardware mappings are performed in the Hardware Mapping Editor; otherwise, if the software detects invalid hardware mappings, page 612 in the imported file, a dialog box opens, which contains information about the invalid entries. Proceed to step 4.</p> <p>NOTE: Once the import operation is completed, you can see a summary in the notification pane.</p>

Step	Action
4	<p>Click:</p> <ul style="list-style-type: none">• Yes to import only the valid data; invalid entries will be skipped.• No to cancel the entire import operation. <p>NOTE: If you proceed, you can see a summary in the notification pane once the operation is completed.</p>
5	If required, correct the invalid entries in the hardware mapping file or make the necessary adjustments in the Control project and import the file again.

Supervision Project Mapping Stage

Overview

This section describes how to use the **Project Explorer** to map the Supervision project of the system.

You must have completed Generation, page 383 and the Configuration, page 498 stages of the Supervision project to proceed with the mapping.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Creating Supervision Project Executables

Overview

The **Executables** node gives you access to the service and communication mapping functionality of the Supervision Participant project.

As such, it allows you to associate:

- The Supervision infrastructure (I/O, alarm, trend, and/or report servers, and/or runtime clients) that is defined in the Supervision Participant project to station nodes of the topology representing operation servers and operator stations.
- The I/O device to a controller and its communication module and/or to compatible Modbus TCP devices and their Modbus TCP server service by using the OFSOPC and/or OPCUA communication protocol.

You can create several executables for a Supervision Participant project, associating the Supervision infrastructure to various topological entities, and its I/O device to various controllers and Modbus TCP devices.

An executable is associated to an **Execution Domain**, which serves as a filter to select applicable services, for example, to define the boundaries for runtime navigation services.

Creating Executables

To create an executable, proceed as follows.

Step	Action
1	<p>In the Supervision Project Browser, right-click the Executables node of the project and select Create Executable.</p> <p>Result: A Supervision executable is created under the Executables node.</p>

Executable Actions

The table describes the executable context menu commands.

Command	Description
Manage	<p>Opens the Manage window in the work area of the Project Explorer, which lets you perform the following actions:</p> <ul style="list-style-type: none"> • Service mapping, page 622: Associates the Supervision servers and clients to one or more station nodes of the system topology. • View and edit the properties, page 620 of the executable. • Communication mapping, page 624: Maps the Supervision I/O device to: <ul style="list-style-type: none"> ◦ A controller and its communication module. ◦ Compatible Modbus TCP devices and their Modbus TCP server service.
Build , page 643	<p>Starts an automated process to integrate changes quickly in the built Supervision Participant project by performing an incremental build. The command applies only for changes that you have made while editing Supervision pages (Edit command).</p> <p>This command becomes available once you have completed the service and communication mapping.</p> <p>NOTE: If the Supervision Participant project is not built yet, you can use this command in place of Build All indifferently.</p>
Build All	<p>Starts an automated process to create, page 643 the built Supervision Participant project the first time or to recreate it to integrate certain changes, page 647 that you have made to the Supervision Participant project.</p> <p>This command becomes available once you have generated the Supervision project and completed the service and communication mapping.</p>
Generate and Build , page 386	Starts a generation of the Supervision Participant project followed by a build of the project executable.

Command	Description
	<p>If you have not built the executable yet, the build process that is executed is the same as when you select the Build All command.</p> <p>If the generation process does not complete successfully, no build is performed and information is displayed in the notification panel.</p> <p>NOTE: After selecting the command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task.</p>
Deploy Built Project , page 693	Allows you to deploy the built Supervision project to the station nodes that are mapped to the executable.
Deploy Changes , page 703	The command is available only if you have already deployed the built Supervision project to at least one station node that is mapped to the executable and if the number and type of changes are supported.
Run Supervision Client	<p>Launches the Supervision runtime locally with the corresponding Supervision project after you confirm the command.</p> <p>The command is available only if the following conditions are satisfied:</p> <ul style="list-style-type: none"> • The executable is mapped to a station node whose NIC is configured with the IP address of the local computer. • The build state of the executable is Built or Out Of Date. <p>For the Supervision project to launch in runtime (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide), some actions need to be performed first, such as deploying the executable.</p> <p>If the runtime is already launched, you can restart it by using the command.</p> <p>The runtime is not launched if the Supervision Participant is being used by EcoStruxure Process Expert (for example, if a Supervision project is being refined or its executable built).</p>
Download Projects , page 646	<p>Opens the Download Projects tab, which allows you to select Supervision project files individually to save them on the computer, such as:</p> <ul style="list-style-type: none"> • The master project (.ctz) with the name <i>ProjectIdentifier_ExecutableIdentifier</i>, which was created by the last successful build process. • Included projects (.ctz) that are related to animated graphics assigned to Supervision pages and those that you have added manually, page 496. • OFS/OPC UA configuration files (.xml).
Delete	Deletes the executable including the associated configurations.
Rename	Lets you to enter a new identifier for the executable.
Properties	Opens the Properties tab in which you can view and edit properties of the executable.

Executable Properties

To open the properties of the executable, double-click it.

You can view and/or edit the following items.

Property	Description
Identifier	The identifier must be unique within a Supervision project.
Description	<p>Optional.</p> <p>You can enter a description for the Executable with free form text.</p>
Execution Domain	<p>Optional.</p> <p>String.</p> <p>Default value: Blank.</p> <p>You can use runtime navigation services (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide) only if the executable of the Control Participant project and the executable of the Supervision Participant project have the same execution domain value.</p> <p>You can assign a logical execution domain to the executable with free form text.</p>

Property	Description
	<p>Values that you enter remain in the menu and you can select them from executables of other Control and Supervision Participant projects.</p> <p>You can modify the Execution Domain parameter at any time during the engineering life cycle.</p> <p>NOTE: Leaving the Execution Domain parameter empty is considered as a value.</p>
Build State	<p>Indicates the status, page 645 of the executable.</p> <p>Default value: Not Built</p>
Built On	Date and time when you last built the executable successfully by using a build command.

Assigning an Execution Domain

To assign an execution domain to an executable, proceed as follows.

Step	Action
1	Double-click the executable to open its properties.
2	Select a value from the Execution Domain menu or enter a new value.

NOTE: To remove or change the associated execution domain, select the empty row at the top of the menu.

Mapping Services

Overview

The **Service Mapping Editor** opens within the **Manage** window when you double-click the Supervision executable, page 619.

The service mapping defines to which station nodes that represent the operation server and operator station computers you deploy, page 693 the Supervision project.

Before you proceed with the service mapping, create the necessary station nodes in the topology of the system and configure them with the services as needed:

- A Supervision service, page 517.
- An **OFS** service, page 517.

You must map all the services of the Supervision Participant project to station nodes in order to complete the build stage, page 643.

NOTE: Install OPC Factory Server or OPC UA Server Expert on the station node that you map to the *IOServer* service.

Mapping Station Nodes Configured with Several NICs

The service mapping does not let you select a specific NIC for each station node.

The IP address of each NIC is created as a network address, page 305 in the Supervision Participant project and associated to the I/O server.

For details on the behavior when deploying to a station node that is configured with more than one NIC, refer to the topic describing the target computer setup, page 694.

Changing the Service Mapping of Deployed Executables

If you have already deployed a Supervision executable to one or more station nodes of the system, when you change the service mapping of this executable, the software clears the related entry in the **Last Deployed On** field, page 699 of the station nodes that you have unmapped. As a result, if you recreate this mapping right away and deploy, it is handled like a first deployment.

NOTE: The service mapping of a station node is unmapped if you delete its Supervision service. It is unmapped if you delete its **OFS** service, page 517 and the station node was mapped to the *IOServer* service of the Supervision project.

Mapping Services

To map a Supervision server or client to a station node, proceed as follows.

Step	Action
1	Double-click the executable of the Supervision project that you want to map and select Manage . Result: The Service Mapping Editor opens.
2	Click the Engine menu next to the server or client that you want to map. Result: The menu displays the identifiers of compatible station nodes, page 622.
3	Select a station node. Result: The software maps the service to the selected station node. NOTE: Once you have mapped the I/O server, the Communication Mapping tab is shown at the bottom of the window.

NOTE: To remove a mapping, in the **Service Mapping Editor**, right-click a service and select **Unmap**.

Mapping Communication

Overview

To open the **Communication Mapping Editor**, click the **Communication Mapping** tab within the **Manage** window, page 619. The tab appears once you have performed the service mapping, page 622 with an I/O server.

The communication mapping defines from which topological entity I/O devices acquire data during runtime by using the selected driver, page 302.

The editor lets you map the I/O devices of the Supervision project to:

- A controller of the system and its communication modules. This includes modules that embed an OPC UA server.
- Modbus TCP devices that feature the Modbus TCP server service (for example, Altivar drives, power meters, STB islands, or PRMs).
- Modbus serial devices by using a topological Modbus TCP device acting as gateway.
- The Control service of a station node acting as an emulated simulator and to which a Control project has been deployed.

For each I/O server that you map, the software creates a protocol-dependent OFS/OPC UA configuration file, page 643. Exceptions apply.

NOTE: Changing the protocol of a tag container, page 341 whose associated I/O device is already mapped to a topological entity removes the mapping.

Physical Connection Prerequisites

To be able to proceed with the communication mapping, the following need to be connected to the same Ethernet network, page 540:

- The NIC of the station node that is mapped to the I/O server in the service mapping, page 622.
- The communication modules of the controller entity.
- The Modbus TCP device (if present).

Mapping Modbus Devices

Before you can map a Modbus TCP or Modbus serial device to be able to monitor data directly from the Supervision infrastructure that is defined in the Supervision project, you need to perform some additional steps.

For details, refer to the topic describing how to monitor Modbus device data directly from Supervision, page 830.

Mapping Modules Embedding an OPC UA Server

When **Protocol** of the tag container that is associated to the I/O device is set to **OPCUA Embedded**, **Data Server (Address B)** is disabled. You can map only one module per I/O device.

When modules embedding the OPC UA server are mounted in both local racks of a redundant configuration, either module can be selected under **Data Server (Address A)**.

To map also the module embedding an OPC UA server located in rack B, create a second I/O device configured as standby, page 299 and associate it to the same tag container.

Mapping Redundant Communication Ports

For controllers and station nodes, you can map up to two communication ports or NICs to a same I/O device if the following conditions are satisfied:

- The communication ports/NICs belong to the same topological entity.
- They are connected to the same physical network.
- Direct TCP/IP Addressing Mode** of the tag container, page 341 that is associated to the I/O device is set to *Unallocated/Allocated Data, PAC Modicon (/U)*.

NOTE: When **Protocol** of the tag container that is associated to the I/O device is set to *OPCUA Embedded*, communication port redundancy for the same I/O device is not possible.

Communication Mapping Tab

The following figure shows an example of the **Communication Mapping Editor** in which one controller and two Modbus TCP devices are mapped to I/O devices through their communication modules.

Assignment(Supervis... × Manage(Executable_1... × System_3 : Project 'Supervision_1' : Executables > Executable_1 : Communication Mapping Editor				
IO Device	Topological Entity	Data server (Address A)	Data server (Address B - Optional)	Modbus Unit ID
IODevice_1	Controller_1	Controller_1 0:Prilocal 0:D 0:R 0:BMEP586040	Controller_1 0:Prilocal 0:D 0:R 02:BMENOC0321	
IODevice_3	EGenericDeviceHW_1	EGenericDeviceHW_1:EIOSERVER	Not Assigned	10
IODevice_2	EATV6xxHW_1	EATV6xxHW_1:EIOSERVER	Not Assigned	255

Header	Description
IO Device	I/O devices that you have created in the cluster and which is associated to a tag container.
Topological Entity	<p>Topological entities that have at least one communication module (data server) connected to the same Ethernet network as the station node to which the I/O server is mapped. These can be controllers, station nodes, Modbus TCP devices, and generic Modbus TCP device entities acting as gateways for Modbus serial devices.</p> <p>The selected entity becomes the data source.</p> <p>NOTE: Restrictions apply when the protocol of the tag container that is associated to the I/O device has the following value:</p> <ul style="list-style-type: none"> <i>OPCUA</i>: Controllers of the Quantum platform are not shown. <i>OPCUA Embedded</i>: Only controllers that have a module, which embeds an OPC UA server and that is connected to the same Ethernet network, page 624 are shown.
Data Server (Address A)	<p>Communication modules of the selected topological entity that are connected to the same Ethernet network as the station node to which the I/O server is mapped.</p> <p>(The IP address of the module is set as the primary address in the OFS/OPC UA configuration file.)</p> <ul style="list-style-type: none"> For Modbus I/O device entities, it is the Modbus TCP server service of the topological entity. For a station node, if more than one ControlExpert service exists, it is the NIC followed by the controller simulator port that is configured for the service. Otherwise, only the NIC appears. <p>NOTE: When the protocol of the tag container that is associated to the I/O device is set to <i>OPCUA Embedded</i>, only modules that embed an OPC UA server are shown.</p>

Header	Description
Data Server (Address B)	<p>Optional. The menu is available only if the topological address A has been mapped.</p> <p>Communication modules of the selected topological entity that are connected to the same Ethernet network as the station node to which the I/O server is mapped excluding the one that has been mapped to address A.</p> <p>The IP address of the module is set as the secondary address in the OFS/OPC UA configuration file.</p> <p>NOTE: The menu is disabled in either case:</p> <ul style="list-style-type: none"> • The topological entity is a Modbus TCP device. • Protocol of the tag container that is associated to the I/O device is set to <i>OPCUA Embedded</i>.
Modbus Unit ID	<ul style="list-style-type: none"> • For Modbus TCP devices, it is the ID parameter of the topological entity. The value corresponds to the default value of the template or to the value that you had configured (when available) before selecting the device and its data server. You can change the value for the device. The change is not propagated to the topological entity. Changing the value in the topological entity is not propagated to devices that are already mapped. • For Modbus serial devices, if you have already configured an application template of the device (for example, an instance of <i>\$ATV71MB</i>), enter the ModbusAddress parameter value of the instance. Otherwise, enter the unique serial slave ID of the device. <p>Valid range: 0 to 255</p> <p>Refer to the documentation of the device for information on the parameter configuration.</p> <p>NOTE:</p> <ul style="list-style-type: none"> • The software does not verify the uniqueness of entries. • The parameter is disabled for station nodes and controllers.

Mapping Station Nodes

When you map an I/O device to the Control service of a station node, you can simulate runtime services.

The station node is available in the **Communication Mapping Editor** only if its NIC is connected to an Ethernet network and if you have created at least one Control Expert service, page 517 in the station node.

NOTE: Map, page 626 the Control executable of the corresponding Control Participant project to the Control service of the station node that uses the same port (service mapping).

Mapping Communication

To map the I/O device to the communication module/card of a topological entity, proceed as follows.

Step	Action
1	Select the Communication Mapping tab within the Manage window.
2	Click the Topological Entity menu next to the I/O device you want to map. Result: The menu displays the identifiers of the compatible entities of the system.
3	Select an entity.
4	Click the Data Server (Address A) menu. Result: The menu displays the identifiers of the available communication modules/cards of the selected entity.
5	Select a communication module/card.
6	If needed, click the Data Server (Address B) menu and select a second communication module/card of the same entity.

NOTE: To remove the entire communication mapping for an I/O device, right-click a communication module/card in either **Data Server** column and select **Unmap**.

To remove the mapping of one data server only, select **Not Assigned** from the menu.

Build Stage

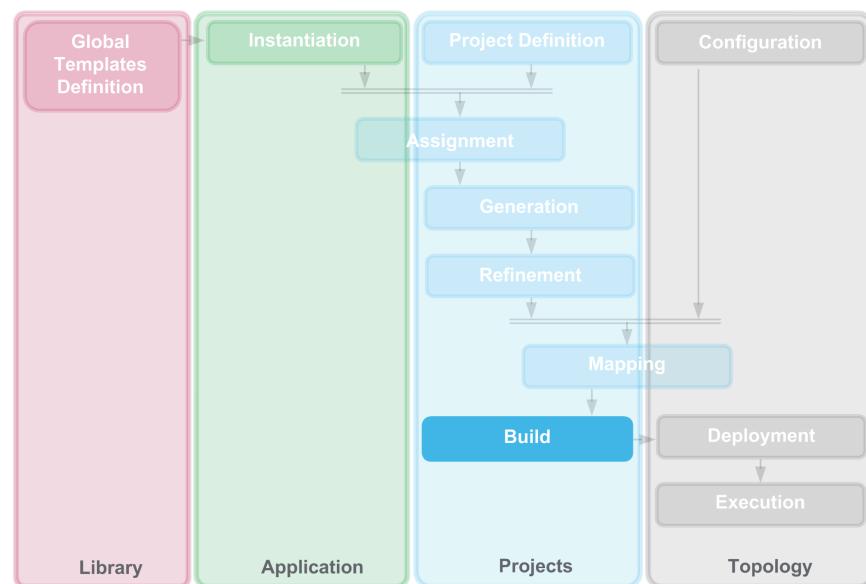
What's in This Chapter

Control Project Build Stage	629
Supervision Participant Project Build Stage	642

Overview

This chapter describes how to create the built Control and Supervision Participant projects, which are the executables and project files that you deploy to controllers and/or station nodes of the system.

The following figure shows the position of the **Build** stage within the system engineering life cycle.



Refer to the Build stage, page 53 for a description of the purpose of this stage.

Control Project Build Stage

Overview

This section describes how to use the **Project Explorer** to create the built Control Participant project for the first time and, in a second step, to integrate changes that you have made to the Control Participant project and to the topology of the system.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Building the Logical Control Participant Project

Overview

In this automated process, the software creates a unique built Control Participant project, which merges:

- The logical Control Participant project information, including changes made during refinement.
- The mapping information, which is associated to the Control executable of the logical project:
 - Services
 - Hardware
 - Communication (if applicable)
- The topological information, which is associated to the mapped services

The software stores the resulting program file (.stu) in the content repository, page 937.

The build process also:

- Creates the necessary DDDT variables with the number of input and output words that you have configured to allocate memory.
- Creates the I/O scanner lines for the communication channels that exist in the communication mapping given that sufficient memory is available, page 512.
- Defines the value of REF_TO data types (see EcoStruxure™ Process Expert, Control Participant Services, User Guide):
 - As *REF(%MWx)* or *REF(%CH<@mod>.<c>)* (IODDT) for direct addressing data instances (HAL2.0 variables) that are used in hardware mapping.
 - As *REF(DDDT)* for variables that are used in communication mapping.
 - As *REF(PES_CONST_TRUE)* for variables that need to be initialized as *TRUE*.
- Introduces the REF_TO ANY_BOOL data type because of the legacy feature to convert BOOL to EBOOL, and the other way around becoming obsolete. If you have updated existing Schneider Electric HAL templates in systems with HAL2.0 templates (see EcoStruxure™ Process Expert, Control Participant Services, User Guide), both data types are converted to REF_TO ANY_BOOL.

NOTE: The conversion does not apply to templates created by users except if they reference Schneider Electric HAL templates (for example, *\$DISignal_UL*).

To perform a subsequent build or if the **Build** command has become unavailable, refer to the topic [Changing the Built Control Participant project](#), page 636.

NOTE: You may need to update the firmware version of topological entities to support the ANY_BOOL data type during build. For information on compatible firmware versions, refer to the platform release notes.

Device Type Manager and Device DDT Variables for Modbus TCP Devices

During build, the software adds a generic **Modbus Device** device type manager (DTM) for each Schneider Electric Ethernet Modbus TCP device, page 531 that you have configured in the topology of the system and which communicates with a controller of the M580 platform or by using a NOC communication module. You do not need to add it.

Modbus TCP devices can be:

- An STB Island
- A PRM
- A Modbus TCP I/O device

The DTM name is composed of four elements. It has the following syntax.

<I/O device name><Counter>_<Client communication module><Position>

Where:

- <I/O device name> is the identifier of the topological entity, page 499. The identifier is truncated not to exceed 10 characters.
- <Counter> is an incremental 1 or 2-digit number generated by the software so that truncated device names are unique.
- <Client communication module> is the commercial reference of the communication module of the client controller that is used to communicate with the device. The reference is truncated to not exceed 12 characters.
- <Position> is the position of the communication module on the rack.

For example, EATV71HW_11_BMENOC030122 (1 in 11th position from the left is the counter and 2 in last position is the position of the module. EATV71HW_1 is the I/O device name. BMENOC03012 is the commercial reference of the communication module).

The Control Participant creates the corresponding DDDT variable.

Device Type Manager and Device DDT Variables for Peer to Peer Communication

During build, the software adds a generic **Modbus Device** device type manager (DTM) in the consumer (client) Control project for each communication module of a server counterpart with which a peer to peer communication channel exists. The DTM is added if the controller acting as client is of the M580 platform or if communication client-side is implemented by using a NOC communication module.

The DTM name is composed of four elements. It has the following syntax.

<Server entity><Counter>_<Server communication module><Position>

Where:

- <Server entity> is the identifier, page 509 of the topological entity acting as server, page 583. The identifier is truncated to not exceed 10 characters.
- <Counter> is an incremental 1 or 2-digit number generated by the software so that truncated server entity names are unique.
- <Server communication module> is the commercial reference of the communication module of the controller acting as server. The reference is truncated to not exceed 12 characters.
- <Position> is the position of the communication module on the rack.

For example, Controller1_140NOE771012 (1 in 11th position from the left is the counter and 2 in last position is the position of the module. Controller_1 is the server entity name, which has been truncated. 140NOE77101 is the commercial reference of the communication module server side).

The Control Participant creates the corresponding DDDT variable.

Device DDT Variables Created in the Logical Control Project

If you have created a device DDT variable, page 445 in the logical Control project and this variable does not exist in the controller that is mapped to the executable (for example, because you have removed the module to which it is associated), the resulting built Control project contains it in unmapped (unlocked) state.

If a device DDT variable with the same name exists in both the logical Control project and the controller that is mapped to the executable but with a different data type or type version, the resulting built Control project contains the variable of either type based on the following rule.

DDDT variable in the controller	DDDT variable in the logical Control project	Result in the built Control project
Mapped	Unmapped	Variable of the same type as in the controller mapped to the executable.
Unmapped	Unmapped	Variable of the same type as in the logical Control project.

Validity Indicators

If a validity icon, page 324 is displayed, you may not be able to build the project successfully. Some exceptions apply, page 388.

In such case, Refine the Control Participant project, page 436 to fix the issue. The Control Participant features tools that help you identify validity issues.

When you save the Control Participant project after refining, page 439 it and project analysis is enabled, page 263, the project is analyzed. The status of the Control project is indicated by a validity icon.

Building the Control Project for the First Time

Use the **Build** or **Build All** command indifferently to create the built Control Participant project for the first time.

NOTE: If you have not built the Control Participant project yet, you can also use the **Generate and Build** command. For information on the generation process, refer to the topic describing the generation of Participant projects, page 385.

Control Participant Project Settings

When you build a logical Control Participant project by using the **Build All** command, the software may override some project settings (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) that you have set during configuration.

To view the **Project Settings** of the built Control Participant project, use the **Open Built Project** command, page 634.

NOTE: This applies also when you are using the **Generate and Build** command to build a Participant project executable for the first time.

Allocation of Memory in I/O Scanner Lines

The build process allocates the memory in the following order: First write, then read.

Read and write addresses are created contiguously for clients of the M580, M340, and Quantum platforms.

The read/write memory allocation varies depending on the platform.

Platform	Memory allocation process
Quantum	<p>The write address should be the end of the read address of the previous channel. However, for example, if 2 channels exist for entity A, and 1 channel exists for entity B, then the memory allocation is the following:</p> <ol style="list-style-type: none"> 1. Write memory allocation for any existing channels of entity A. 2. Read memory allocation for any existing channels of entity A. 3. Write memory allocation for the channel of entity B. 4. Read memory allocation for the channel of entity B. <p>NOTE: The first 8 words are allocated for the device control block.</p>
M340 M580	<ol style="list-style-type: none"> 1. Allocates the memory for writing for any existing channels. 2. Allocates the memory for reading for any existing channels.

NOTE: The software detects an error in the configuration during build if you have not configured sufficient memory for a client controller based on the Quantum platform with NOC communication module (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

Allocation of Peer to Peer Communication Variables

The build process allocates addresses or values to the network variables that you have mapped, page 594 to communication channels.

The start address of a variable in the channel needs to be even and contiguous with the channel. Therefore, if the start address of the communication channel created in the client is:

- Odd:
 - No variable is allocated to the first word of the client communication channel.
 - Addressing of variables in this channel starts from the second word.
 - The offset of the server read/write memory counterpart is the first word of the server communication channel.
- Even:
 - Addressing of variables in this channel starts from the first word.
 - Read and write size of the memory is equal to the value of the **ServerMemoryLength** parameter - 1.
 - The offset of the server read/write memory counterpart is the second word of the server communication channel.

NOTE: If by using the **Refine** command, you delete or change the name, type (ID or version), or size of any variable or network variable that is mapped to a communication channel, the build process cannot complete successfully. The software displays diagnostic information in a dialog box. In such case, open the **Variable Mapping Editor** to remove, page 600 the inconsistent variable or undo the changes to the variable. Start the build process over.

The figure shows examples of entries created during build in the consumer project, page 452 (client based on the M580 platform) for the **Value** attribute of two sample peer to peer communication variables that are used to read data from the server (*DISignal_UL_1V*) and write data to the server (*Valve_1_VALVE_OPV*).

Name	Type	Value	Address
DISignal_UL_1V	REF_TO ANY_BOOL	REF(M580_Server_BMENOC030122.Inputs.Free[0].0)	
Valve_1_VALVE_OPV	REF_TO ANY_BOOL	REF(M580_Server_BMENOC030122.Outputs.Free[0].0)	

The figure shows examples of entries created during build in the owner project (server based on the M580 platform) for the **Value** attribute of the two counterpart sample peer to peer communication variables.

Name	Type	Value	Address
DISignal_UL_1V	REF_TO ANY_BOOL	REF(%MW2.0)	
Valve_1_VALVE_OPV	REF_TO ANY_BOOL	REF(%MW52.0)	

Aborting Build Tasks

After selecting a build command or confirming it, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. No build is performed.

Only the user who has initiated the command is allowed to abort it.

NOTE: If you are building only a small Control Participant project, the task may complete before you are able to click the icon.

Using the Build Commands

To build the Control Participant project, modify the built Participant project through an incremental build, or build it again, proceed as follows.

Step	Action	Result
1	Right-click the executable and select Build , Build All , or Generate and Build depending on the situation, page 638.	If no validity icon, page 632 is displayed, the software executes the following actions; otherwise it displays a confirmation dialog box. In such case, proceed to step 2. <ul style="list-style-type: none"> • Proceeds with the creation or update of the built Control Participant project. • Stores the program file (.stu) in the content repository. • Displays information in the notification panel. • Displays a dialog box containing a summary of the build process.
2	In the confirmation message, click No . NOTE: Click Yes to proceed with the build anyway; the process may not complete successfully.	The confirmation dialog box closes without building the Control Participant project.
3	Refine the Control Participant project, page 439 to fix the issue.	–
4	Start the build process over.	–

NOTE: If the build process does not complete successfully, the software displays a notification to inform you and provides indications about the cause. In such case, make the necessary corrections and build the project again.

Build Status

The table describes the build status of an executable, page 579.

Status	Description
Not Built	No built Participant project exists. Default status when you create an executable.
Built	The built Participant project is up to date.
Out Of Date	The built Participant project is out of date and needs to be built again, page 638.

NOTE: You may need to build the Control Participant project again if you have made certain changes, page 636 to the system even though the software displays the **Build State** as **Built**.

Viewing the Built Participant Project

Once you have built the Control Participant project, you can open the project in the Control Participant to view it.

You may also be able to open the built Control Participant project if the software reports detected errors during the build process to help you with the troubleshooting.

NOTE: Do not make changes to the built Control Participant project from within the Control Participant as the software discards any changes upon closing the reduced Participant window.

To make changes to the system, you can use the following methods:

- Open the system using the corresponding explorer.
- Refine, page 436 the logical Control project.

To open the built Control Participant project in the Control Participant, proceed as follows.

Step	Action
1	<p>Right-click the Control executable and select Open Built Project.</p> <p>Result: The Control Participant opens extracted, page 116 and maximized. You can view the contents of the built executable.</p> <p>NOTE: For more information, refer to Viewing the Built Control Participant project (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).</p>
2	Close the Control Participant window.

NOTE: To save a copy of the built Control Participant project, use the **Save as** command of the **Content Repository** explorer, page 939.

Changing the Built Control Participant Project

Overview

This automated process allows you to change the built Control Participant project by integrating changes that you have made to the Control Participant project, or to the topology of the system.

Depending on the modifications you made, the status of a built Control Participant project changes and you may need to use a specific command to integrate changes.

Changes Modifying the Build Status

The table describes how changes affect the executable of a Control Participant project whose status is **Built**.

Change description	Resulting build state
Changes to the topology of the system (for example, properties, controller rack configuration, IP addresses, physical connections, replacing templates of I/O device entities, deleting STB island entities, Ethernet networks entities, safety password, and so on).	Out Of Date
Changing the logical Control Participant project and generating it.	
Deleting a section in the Project Explorer .	
Selecting the Refine command after adding, renaming, or changing the order of sections in the Project Explorer .	
Refining the Control Participant project.	
Selecting Built Participant Project for an executable when importing a Control Participant project in the Project Explorer . It applies in the following cases: <ul style="list-style-type: none"> An executable with the same identifier already exists in the target project and its status is Built. You are importing an executable that does not exist yet in the target project and its status was Built at the time of export. 	
Updating, page 748 the logical Control Participant project.	
Changing the service, communication, or hardware mapping of the Control executable. For communication mapping, in case a peer to peer communication channel exists, changes include creating, modifying, or deleting a variable mapping, page 594 ⁽¹⁾ .	
Removing the service mapping.	Not Built
Deleting a controller or station node that is selected in the service mapping, or deleting the ControlExpert service, page 517 of such a station node.	

(1) When a communication channel and variable mappings for peer to peer communication exist between two built Control project executables, making a change affecting the communication channel or variable mapping in one executable, which sets its build status to **Out Of Date** also changes the build status of the other executable to **Out Of Date**. Such change can be, for example, deleting the Control project or its executable, deleting a mapped variable, deleting a section containing a mapped variable, disconnecting a mapped controller entity, and so on.

NOTE: After making a change, which modifies the build status, build the logical Control Participant project again using the appropriate command, page 638.

Building after Creating, Updating, or Deleting Device DDT Variables in the Logical Control Project

If you have created a device DDT variable, page 445 in the logical Control project and this variable does not exist in the built Control project, the updated built Control project contains it in unmapped (unlocked) state.

The following table describes the result when building changes if a device DDT variable with the same name exists in both the logical and the built Control project but with a different data type or type version.

DDDT variable in the built Control project	DDDT variable in the logical Control project	Result in the updated built Control project
Mapped	Unmapped	Variable of the same type as it was in the built Control project.
Unmapped	Unmapped	Variable of the same type as in the logical Control project if the type that is updated is not used by a mapped device DDT variable.

The following table describes the result when building changes if a device DDT variable that exists in the built Control project has been deleted from the logical Control project.

DDDT variable in the built Control project	DDDT variable in the logical Control project	Result in the updated built Control project
Mapped	Variable is deleted and references to it in code are removed.	<ul style="list-style-type: none"> The device DDT variable is not removed. Code is updated.
Unmapped		<ul style="list-style-type: none"> The device DDT variable is removed. Code is updated.

Using the Appropriate Command to Integrate Changes

The table indicates which command to use to update the built Control Participant project depending on the change that you have made.

Change description	Command	Action
<ul style="list-style-type: none"> • Adding, deleting, or changing: <ul style="list-style-type: none"> ◦ Any section (including the order of sections) ◦ Variables⁽¹⁾ (including device DDT variables, page 445) ◦ The hardware mapping • CCOTF-compatible changes, page 505 in the configuration of a controller⁽²⁾ 	Build ⁽⁵⁾	Performs an incremental build.
Changing the topology of the system ⁽³⁾ .	Build All	Performs a complete build.
Changing the communication or network variable mapping ⁽³⁾ .		
After an import into a Control Participant project ⁽³⁾ .		
After regenerating a Control Participant project ⁽³⁾ .		
<ul style="list-style-type: none"> Adding, deleting, or changing: <ul style="list-style-type: none"> • Animation tables⁽⁴⁾ • Operator screens⁽⁴⁾ • DDTs or DFBs⁽³⁾ 	Build All	Performs a complete build.
Changing the order of FBD or non-FBD sections in the Control Participant project by using the Refine command ⁽⁴⁾ .		
Moving FBD or non-FBD sections from one task folder to the other (for example, from MAST to FAST) while refining the Control Participant project (Refine command).		
Changes that set the build status to Not Built , page 636.	Build or Build All	Either command performs a complete build.

- (1) To integrate changes to attributes of variables used in hardware mapping, page 639, use the **Build All** command.
- (2) Except changes to the S908 architecture, which require using the **Build All** command to integrate these changes even if the **Build** command is available.
- (3) Such changes disable the **Build** command. Except for CCOTF-compatible changes.
- (4) Using the **Build** command after such change does not update the built Control Participant project.
- (5) You can also use the **Generate and Build** command, page 386.

NOTE: When you modify an existing peer to peer communication channel, rebuild the owner and the consumer, page 452 Control Participant projects.

Performance Considerations

When you have made changes to a built Control Participant project and the appropriate command to update it is **Build**, in certain cases you may be able to reduce the time required to update the built Participant project by using the **Build All** command instead.

This can be the case, for example when you make changes to many sections of a built Control Participant project at once.

Managing Hardware Abstraction Layer (HAL) Variables and DFBs

Overview

The Global Templates library contains HAL application templates, which generate elementary variables representing the value and/or quality of hard-wired signals.

HAL is a mechanism that emulates access to hardware resources to maintain an application device-independent. By using such templates, you can use the elementary variables that they generate although the hardware channels are not yet part of the logical Control Participant project. Data related to the topology is associated to the logical Control project during build.

The following types of templates exist:

- HAL templates generating elementary digital or analog Control Participant variables (for example, signal conditioning templates).
- Special card templates containing DDTs, DFBs, and/or IODDTs, allowing to use the data exchanged with the special in-rack modules (for example, Modicon M340 high speed counter module device templates).

Facets referenced by these templates are used during the hardware mapping.

For details on these templates, refer to the help, page 101 of the Schneider Electric libraries for EcoStruxure Process Expert.

Marking HAL Variables and DFBs

When you build the logical Control Participant project, the software marks the following code that is used in the hardware mapping with the *Custom* attribute **HAL**:

- Elementary variables coming from mapped facets of hardware abstraction layer templates or special card templates.
- DFB instances and IODDTs coming from mapped facets of special card templates and that are added to a section during build.

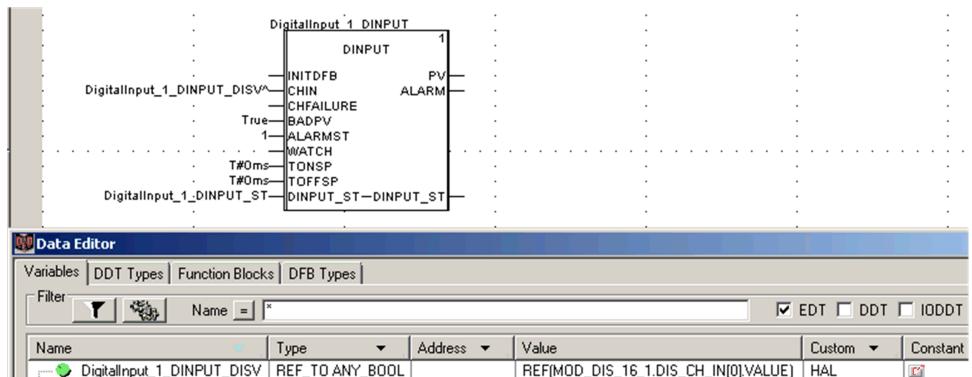
Limitations can apply when you:

- Refine sections containing marked DFB instances, using the **Refine Online** command (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
- Update the logical Control Participant project, page 748 with changes made to HAL variables or sections containing marked DFB instances.

NOTE: DFB instances and IODDTs are marked with the attribute **HALFB** to distinguish them from EDTs.

Properties of HAL Variables

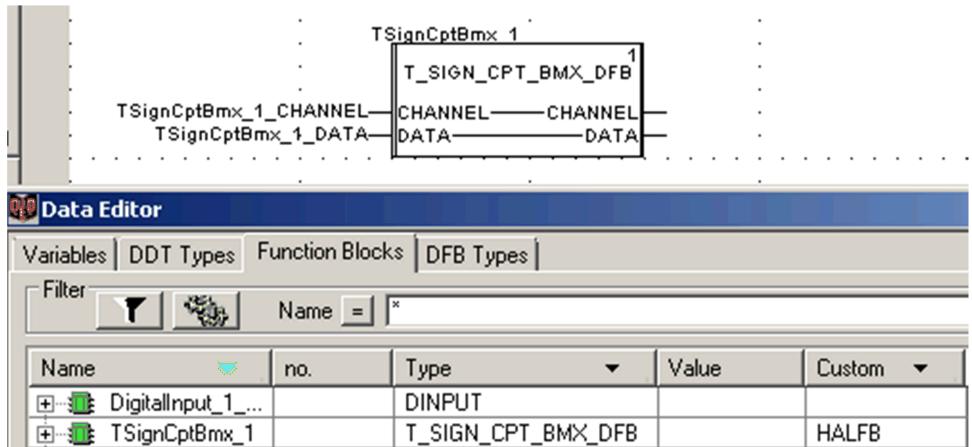
In this example, facets of instance **DigitalInput_1** have been assigned to an FBD section and generated. In the hardware mapping, the **DigitalInput_1_DINPUT_DIS** facet is mapped to a channel of the digital input module of the controller. The controller entity is based on the M580 CPU. The following figure shows the built Control Participant project where the software has marked the **DigitalInput_1_DINPUT_DISV** variable that is used in the hardware mapping.



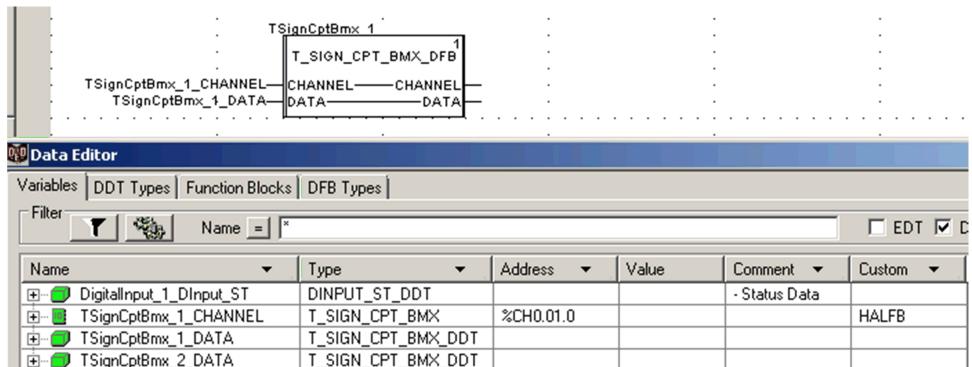
Properties of HAL IODDT Variables and DFBs

In this example, the facet of instance **TSignCptBmx_1** (modeling a high speed counter module) has been assigned to an FBD section and generated. In the hardware mapping, the **TSignCptBmx_1** facet is mapped to a channel of the high speed counter module of the controller. The following figure shows the built Control Participant project where the software has added the **TSignCptBmx_1** DFB encapsulated in this facet to the section and marked it.

NOTE: The data type and addressing method for variables shown in this example may vary depending on the version of the template generating them.



The following figure shows the same built Control Participant project where the software has created the corresponding IODDT **TSignCptBmx_1_CHANNEL** and marked it.



NOTE: A **TSignCptBmx_2_DATA** facet of another high speed counter instance, for the second channel of the counter module, has been assigned to the FBD section and generated. However, the hardware mapping for the second channel has not been done, thus the second DFB instance and the corresponding IODDT have not been added to the section.

Viewing Sections Containing HAL DFBs and Variables

Once you have built the Control Participant project, use the **Open Built Project** command to view the **HAL** attribute, allowing you to identify variables and DFB instances that the software has marked.

NOTE: After deployment, you can also view this attribute by using the **Refine Online** command.

NOTE:

This attribute is not visible when you refine the logical Control Participant project by using the **Refine** command.

Variables	DDT Types	Function Blocks	DFB Types
Filter			
<input type="text"/> <input type="button"/> <input type="button"/>	Name = <input type="text" value="*"/>	<input checked="" type="checkbox"/> EDT <input type="checkbox"/> DDT <input type="checkbox"/> IODDT <input type="checkbox"/> Device DDT	
Name	Type	Address	Value
DigitalInput_1_DINPUT_DISV	REF_TO ANY_BOOL		Custom

Changing HAL Variables

To integrate into the built Control Participant project changes that you make to variables that the software marks as **HAL**, using the **Refine** command, refer to Using the Right Command to Integrate Changes, page 638.

Supervision Participant Project Build Stage

Overview

This section describes how to use the **Project Explorer** to create the built Supervision Participant project for the first time and, in a second step, to integrate changes that you have made to the Supervision Participant project.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Building the Supervision Participant Project

Overview

In this automated process, by using the compilation mechanism of the Supervision Participant, the software creates a unique built Supervision Participant project, which merges:

- The logical Supervision Participant project information.
- The mapping information, which is associated to the executable of the logical Participant project:
 - Services
 - Communication
- The topology information, which is associated to the mapped services.

The software stores the resulting Supervision project file (.ctz) in the content repository, page 936.

This is the file that you can deploy to computers running Supervision software.

I/O Device Addresses

When the protocol for a tag container is **OPCUA** or **OPCUA Embedded**, the build process replaces the temporary IP address and port value, page 299 in the address of the I/O device that is associated to the container in order to create the actual I/O device address in the Supervision Participant project.

Protocol for tag container	I/O device address format
OPCUA	<p><i>opc.tcp://<IP address>:<TCP listening port>/OPCUAServerExpert</i></p> <p>Where:</p> <ul style="list-style-type: none"> • <IP address> is the IP address of the station node, page 520 that is mapped, page 622 to the <i>IOServer</i> service. • <TCP listening port> is the value that is configured for TCP Port for this station node.
OPCUA Embedded	<p><i>opc.tcp://<IP address>:<TCP listening port></i></p> <p>Where:</p> <ul style="list-style-type: none"> • <IP address> is the IPv4 address (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) of the communication module embedding the OPCUA server and that is selected in the communication mapping, page 624 for the I/O device associated to the tag container. • <TCP listening port> is the value that is configured for this module in the Control Participant.

NOTE: For other protocols, refer to the **Address** property, page 299 of the I/O device.

OFSOPC and OPC UA Configuration Files

During the build process, the software generates the necessary OPC Factory Server and OPC UA Server Expert configuration files and stores them in the contents repository.

One configuration file is created for each station node that is mapped, page 622 to an *IOServer* service. The name of the configuration file contains a suffix that identifies the protocol that is used.

In case more than one protocol is used in the Supervision project, at least one configuration file per protocol is generated.

Protocol used	Configuration file name	Configuration file aliases
OFSOPC	<Topological entity>_DeviceConfig_OFSDA.xml For example, OperationServer_DeviceConfig_OFSDA.xml	One device alias exists for each I/O device of the same I/O server. The alias name is <i>Alias_<I/O device></i> where <I/O device> is the identifier of the I/O device that is mapped, page 624 to the data source. For example, <i>Alias_IODevice_1</i> .
OPCUA	<Topological entity>_DeviceConfig_OPCTUA.xml For example, OperationServer_DeviceConfig_OPCTUA.xml	One device alias exists for each tag container. The alias name is the tag container identifier, page 341. For example, <i>TagContainer_1</i> . NOTE: The User authentication and X509 user token properties are enabled by default. For details, refer to <i>User Authentication and Authorization</i> in the help of OPC UA Server Expert.
OPCUA with OPC UA server embedded in module of controller, page 341	No configuration file is generated.	Not required. The OPC UA server is embedded in a module of the controller that is the data source. The variable tag address has the format <i>ns=2;s=0:<variable name></i> where <i>ns</i> indicates the namespace and 2 corresponds to the OPC UA server as defined by Schneider Electric. For example, <i>ns=2;s=0:InputVar1</i>
Where <Topological entity> is the identifier of the station node to which the I/O Server service is mapped in the service mapping, page 622		

If you rename the station node, the name of the configuration file is updated without the need to build the Supervision Participant project.

NOTE: For information on how to generate symbol table files (.xvm), refer to the topic describing the Control project deployment process, page 652.

Network Addresses of Supervision Project Components

When you build a Supervision Participant project, and you have not done the service mapping, page 622 of a server and/or client of the project, the software does not generate an IP address for this component. Because of this, the build process of the Supervision Participant project cannot complete successfully.

Proceed with the service mapping of any Supervision Participant project component that you have created before building the project.

For example, if you have not mapped the alarm server of the project, the network address of the associated topological entity is not generated into the Supervision Participant project.

Building When Instance Identifiers Start with a Digit

When the logical Supervision Participant project that you want to build contains facets referenced by instances whose identifier starts with a digit, you need to set the *tagstartdigit* parameter of the Supervision Participant configuration file (.ini) to 1 for the build process to complete successfully.

Building the Supervision Project for the First Time

Use the **Build** or **Build All** command indifferently to create the built Supervision Participant project for the first time.

NOTE: If you have not built the Supervision Participant project yet, you can also use the **Generate and Build** command. For information on the generation process, refer to the topic describing the generation of Participant projects, page 385.

Aborting Build Tasks

After selecting a build command, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. No build is performed.

Only the user who has initiated the command is allowed to abort it.

NOTE: If you are building only a small Supervision Participant project, the task may complete before you are able to click the icon.

Using the Build Commands

To create the Supervision Participant project, modify it, or to create it again, proceed as follows.

Step	Action	Result
1	Right-click the executable and select Build , Build All , or Generate and Build depending on the situation, page 647.	<p>The software executes the following actions:</p> <ul style="list-style-type: none"> Creates the creation or update of the Supervision Participant project. Stores the project files in the contents repository. Displays information in the notification panel. Displays a dialog box containing a summary of the build process. <p>NOTE: Building the Supervision project may take time.</p>

NOTE: If the build process does not complete successfully, the software displays a notification to inform you, and provides indications about the cause. In such case, make the necessary corrections and build the project again.

Build Status

The table describes the build status of an executable, page 620.

Status	Description
Not Built	No built Participant project exists. Default status when you create an Executable .
Built	The built Participant project is up to date.
Out Of Date	The built Participant project is out of date and needs to be built again.

NOTE: You may need to build the Supervision Participant project again if you have made certain changes, page 645 to the system even though the software displays the **Build State** as up to date.

Changes Modifying the Build Status

The table describes how changes affect the status of the executable of a Supervision Participant project that is already built.

Change	Resulting build state
Renaming an I/O device or the Supervision project.	Out Of Date
Generating the project after making changes.	
Creating a page.	
Changing the project through refinement.	
Any import operation in the Project Explorer .	
Changing service or communication mapping of the Supervision executable.	
Changes to the topology of the system, for example: <ul style="list-style-type: none"> <li data-bbox="508 516 1151 595">• Changing the IP address of the NIC of a station node or of a module that is selected in the service or communication mapping. <li data-bbox="508 595 1151 662">• Deleting the ControlExpert or OFS service, or the NIC instance of a station node that is selected in the communication mapping. <li data-bbox="508 662 1151 729">• Deleting the Supervision service of a station node that is selected in the service mapping. <li data-bbox="508 729 1151 797">• Deleting a controller that is selected in the communication mapping. <li data-bbox="508 797 1151 842">• Changing the TCP Port of a station node that is selected in the service mapping for the I/O server when the protocol that is used for at least one tag container is OPCUA. 	
Removing all service mappings of the Supervision executable.	Not Built
Deleting a station node that is selected in the service mapping.	

Downloading Supervision Project Files

After the build process completes successfully, the following files are available for download in the **Download Projects** window, page 619:

- The master project (.ctz) with the name *ProjectIdentifier_ExecutableIdentifier* (for example, *Supervision_1_Executable_1*).
The project also contains any user files, page 494 that are attached to the Supervision Participant project.
- The included projects (.ctz) that are related to animated graphics, which have been added to Supervision pages, page 494.
- Any other included project that you have added manually, page 496 to the Supervision Participant project.
- The OFSOPC/OPC UA configuration files (.xml) if you have configured the communication board, page 302 driver as **OFSOPC** or **OPCUA**.

NOTE: You can download included projects as soon as the software has added them to the Supervision Participant project.

To download a project file, proceed as follows.

Step	Action
1	Right-click the executable that you have built and select Download Projects . Result: The Download Projects window opens and shows the project files associated to the built Supervision Participant project.
2	Select one file and click Download . Result: The Save As dialog box opens.
3	Browse to the location where you want to save the file and click Save .

Changing the Built Supervision Participant Project

Overview

This automated process allows you to change the built Supervision Participant project by integrating changes that you have made to the Supervision Participant project.

For performance considerations, when possible, use the **Build** command. It requires less time to complete than **Build All**. However, you can use the **Build All** command in any case.

NOTE: You can also use the **Generate and Build** command, page 386.

Using the Right Command to Integrate Changes

The table indicates which command to use to update the built Supervision Participant project depending on the type of change that you have made.

Change	Command	Action
Changes involving Supervision pages: <ul style="list-style-type: none"> Creating or deleting a page. Editing a page (Edit command): <ul style="list-style-type: none"> Adding or deleting one or more genies or graphical elements. Editing page properties. Generating a page that contains a genie facet with the status Out Of Date or Deleted. 	Build ⁽¹⁾⁽²⁾	Performs an incremental compilation.
Changes involving Supervision data facets ⁽²⁾ : <ul style="list-style-type: none"> Assigning or unassigning a facet from a tag container. Moving an existing facet to another tag container. Generating a tag container that contains a facet with the status Out Of Date or Deleted. 		
Modifying a parameter of the Message category, page 176 of a Supervision element of an instance and generating the corresponding Supervision project (for example, modifying the description of an interlock condition of the InterlockTags element under the Supervision node of an instance).		No compilation is performed.
Other changes, for example, to: <ul style="list-style-type: none"> Clusters. The trend, alarm, report, and/or I/O server. Service and communication mappings. Advanced project settings. Generating or refining the Supervision Participant project (except when generating only changes that involve a Supervision data facet as described in this table). Topological entities, for example, changing the TCP Port of the I/O server station node when the OPCUA protocol is used or changing the configuration of a module that embeds an OPC UA server. 	Build All ⁽²⁾⁽³⁾	Performs a complete compilation.
(1) You are limited in the number of changes that you can make per Supervision Participant project to be able to use the Build command: <ul style="list-style-type: none"> A maximum of 5 changes involving Supervision pages. A maximum of 30 changes involving data facets. When you exceed either limit, you must use the Build All command. (2) You can also use the Generate and Build command, page 386. (3) When you make this type of change, the Build command is not available.		

Deployment Stage

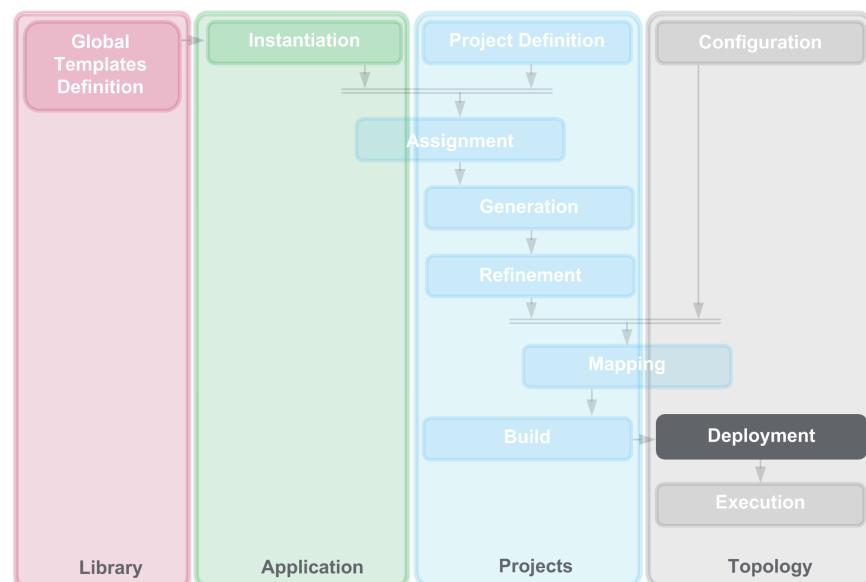
What's in This Chapter

Control Project Deployment Stage	649
Supervision Project Deployment Stage	693

Overview

This chapter describes how to deploy the built Control and Supervision Participant projects to entities that exist in the topology of the system.

The following figure shows the position of the **Deployment** stage within the system engineering life cycle.



Refer to Deployment Stage, page 55 for a description of the purpose of this stage.

Control Project Deployment Stage

Overview

This section describes how to use the **Topology Explorer** to deploy entire built Control Participant projects or parts of them to physical controllers or to PCs emulating simulators, which are modeled by engines in the topology of the system.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Control Participant Project Deployment Process

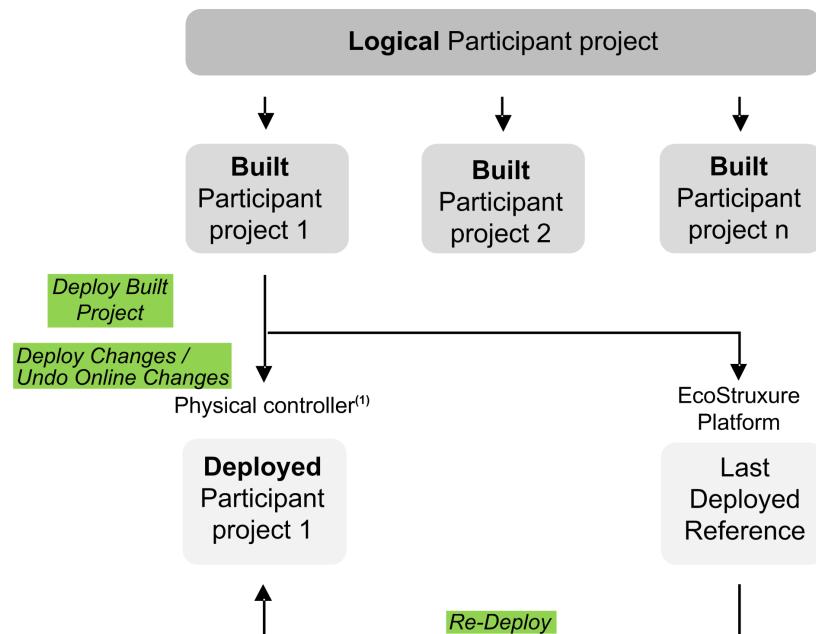
Overview

The software allows you to deploy a built Control Participant project, page 630 to engines by using the following commands:

- **Deploy Built Project**
- **Re-Deploy Last Project**
- **Deploy Changes / Undo Online Changes**

Deployment Flowchart

The following figure illustrates the different steps that are part of the deployment process of the built Control Participant project and their associated commands.



(1) Or emulated simulator (Control service of a station node).

Prerequisites

The following table describes the actions that are prerequisite for using a command.

Command	Actions
Deploy Built Project	Build the Control Participant project and configure the corresponding topological entity.
Re-Deploy Last Project	Deploy the built Control Participant project.
Deploy Changes / Undo Online Changes	<p>Update the project, page 748 with hardware changes made online (if applicable). In addition, the following are prerequisite for specific actions:</p> <p>For Deploy Changes:</p> <ul style="list-style-type: none"> • Deploy the built Control Participant project. • Change the logical Control Participant project and build it. <p>For Undo Online Changes:</p> <ul style="list-style-type: none"> • Deploy the built Control Participant project. • Change the deployed Control Participant project, using the Refine Online command. <p>NOTE: Certain changes cannot be deployed, page 686 by using the Deploy Changes command.</p>

For the other actions that are required, for example, mapping the Control executable to a controller, refer to the system engineering life cycle.

Supported Engines for Deployment

The software allows you to deploy/re-deploy a built Control Participant project, and deploy changes to engines representing:

- A controller
- A redundant controller of the M580 or Quantum platform.
- A computer running software that emulates a simulator, page 516 (modeled by the Control service of a station node in the topology of the system).

Engines need to exist in the topology of the system and be mapped to a Control executable of the project.

Identifying Engines

The software uses the IP address of each engine to identify it on the Ethernet network.

⚠ WARNING	
UNINTENDED EQUIPMENT OPERATION	
Before deploying to a controller, verify that:	
<ul style="list-style-type: none"> • You have assigned a valid IP address to the controller. • The IP address corresponds to the physical address printed on the hardware. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>	

NOTE: To obtain a list of IP addresses of nodes that are connected to the same physical network and their corresponding physical address (media access control (MAC) address), use the Address Resolution Protocol (ARP).

The procedure describes how to use the ARP to obtain a list of IP addresses of nodes connected to the same physical network and their corresponding physical address (MAC address).

Step	Action
1	Write down the MAC address that is printed on the communication module through which you want to deploy or execute.
2	Connect the computer to the Ethernet network to which this communication module is connected and through which you want to deploy or execute.
3	On the computer, execute the <i>cmd.exe</i> program. Result: A command prompt opens.
4	Type <i>arp -a</i> and press Enter . Result: The local ARP cache table opens.
5	In the table, locate the MAC address matching the one printed on the communication module.
6	In the table, verify that for this address, the corresponding IP address matches the IP address that you have assigned to the engine in the Control Participant; otherwise proceed to step 7.
7	Assign to the communication module the IP address that you have assigned to the corresponding engine in the Control Participant and repeat the procedure, starting from step 2.

Client/Server Connection

In case of an unexpected stop of the system server, the notification panel of engineering clients that were connected to the system server before the communication interruption may contain information about operations that did not complete successfully because of this interruption. It may also indicate corrective actions.

After an interruption of the client/server connection, the states of both the engineering client and the system server may have become inconsistent if a deployment operation was in progress.

NOTICE

DATA CORRUPTION

After an interruption of the client/server connection, verify that the last operation executed by the software was completed successfully, and if necessary, repeat the last operation.

Failure to follow these instructions can result in equipment damage.

NOTE: For more information on client and server behavior in case of a communication interruption, refer to the topic describing client/server connection, page 64.

Start/Stop Commands

The software can send a start or stop command to the controller or software emulating a simulator.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING

UNKNOWN OPERATIONAL STATE OF EQUIPMENT

Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Generating Symbol Table Files

You cannot generate symbol table files (.xvm) by selecting the `saveXVM` property in the project settings of the Control Participant. You need to edit a dedicated configuration file instead before deploying a Control project.

To generate symbol table files (.xvm) of the Control Participant and retrieve them from the content repository, proceed as follows.

Step	Action
1	Open the <code>ControllerSettings.xml</code> file that is located at the path <code>C:\Program Files\Schneider Electric\EcoStruxure\Process Expert\System Server</code> on the computer on which the system server is installed.
2	Set the value for the <code>SaveXVM</code> attribute to 1 and save changes. Result: When you deploy changes, page 682 to a Control project or refine it online, page 727, the symbol table file is added to the content repository.
3	In the content repository, open the <code>XvmHolder</code> folder located under Systems > <System name> > Project > Upro .
4	Save the file, page 939.

Deploying to the Controller Simulator

Overview

You can deploy built Control Participant projects to computers for simulation purposes. EcoStruxure Process Expert supports deployment to multiple Controller simulator instances running on a single or several computers.

In the topology, each of these computers is represented by a station node, page 516 on which one or more ControlExpert services (engine) are created, representing simulator instances.

By default, each simulator instance is configured to start with a password-protected Control project (see EcoStruxure™ Process Expert, Installation and Configuration Guide) provided with the software. You can start a simulator instance with a different password-protected project file or change the password, page 661 of the project that is loaded by default.

NOTE: If your Windows® session is not the one that was used to install the software, you need to configure the simulator manually, page 656 to load a password-protected Control project at startup.

Before Deploying to the Simulator

The table describes the actions you need to perform to be able to proceed with the deployment of a Control project that is mapped to a station node/ControlExpert service, depending on the computer on which the simulator is running.

Computer selected for deployment	Actions
Computer on which Process Expert is installed. (Recommended method)	Start a simulator instance.
Other computer on which no software component is installed.	<ol style="list-style-type: none"> 1. Install the simulator. 2. Start a simulator instance. <p>Refer to the topic describing how to install the Controller Simulator. (see EcoStruxure™ Process Expert, Installation and Configuration Guide)</p>

Starting the Controller Simulator

To start an instance of the simulator that is provided with the software, proceed as follows.

Step	Action
1	<p>Double-click the shortcut of the controller simulator on the desktop.</p> <p>Result: The Process Expert Controller Simulator dialog box opens.</p> <p>NOTE: If you had installed the simulator manually, double-click the <i>PLCSimulatorStarter.exe</i> file.</p> <p>NOTE: You can also start a simulator instance by using the command prompt. Refer to the topic describing how to load a project in the simulator in the help of the simulator.</p>
2	<p>Edit the port number that you want the simulator instance to use (port 502 by default) and click Start.</p> <p>Result: The simulator instance starts minimized as an icon in the notification area and uses the configured port. By default the password-protected Control project (see EcoStruxure™ Process Expert, Installation and Configuration Guide) is loaded.</p> <p>NOTE: To be able to deploy a Control project, the port number that is used by the simulator instance must match the value of the Port property of the ControlExpert service that is used in the service mapping of the executable of the project.</p>
3	To start another instance, edit the port number and click Start again.

NOTE: For information on using the simulator, refer to the simulator help, which is located in the PLC_Simulator folder. You can open the help of the simulator also directly from the simulator panel after starting it.

Starting the Simulator With Another Password-Protected Control Project

You can modify the startup setting and have simulator instances load your own password-protected Control project file (.sta).

Once you have configured the simulator to start by loading your own Control project, you will need to provide the password to use it, page 659.

NOTE: The change does not apply to simulator instances that are already running until you restart them.

To start simulator instances by loading a specific password-protected Control project, proceed as follows.

Step	Action
1	Start a simulator instance.
2	Right-click the simulator icon in the Windows® taskbar and select Options.... Result: The Simulator Panel Options dialog box opens.
3	Select the Use default application to start simulator (enforce security) check box.
4	Click the browse button and select your password-protected Control Expert Control project file (.sta).
5	Click OK . Result: The simulator stops.
6	Start the simulator. Result: The simulator starts by loading the selected Control Expert Control project file.

Starting the Simulator Without Password-Protected Control Project

You can modify the startup setting and start simulator instances without loading a password-protected Control project file (.sta). However, modifying this setting is not recommended because it leaves the Ethernet port of the computer, which is used by the simulator vulnerable to cyber attack.

NOTE: The change does not apply to the other simulator instances that are already running. You need to restart them.

To start simulator instances without loading a password-protected Control project, proceed as follows.

Step	Action
1	Start a simulator instance.
2	Right-click the simulator icon in the Windows® taskbar and select Options.... Result: The Simulator Panel Options dialog box opens.
3	Clear the Use default application to start simulator (enforce security) check box.
4	Click OK . Result: You do not require to enter a password to use this simulator instance and the ones you start from now on.

Starting the Simulator Under Another Windows® session

Once you have configured simulator instances to start by loading a password-protected Control project, you will need to provide the password to use it, page 659.

To start simulator instances by loading the default Control project provided with the software or another one when you are logged on to a Windows® that is different from the one used to install the software, proceed as follows.

Step	Action
1	Start a simulator instance.
2	Right-click the simulator icon in the Windows® taskbar and select Options.... Result: The Simulator Panel Options dialog box opens.
3	Select the Use default application to start simulator (enforce security) check box.
4	Click the browse button and select the password-protected Control project file (.sta) that you want to use. This is either of the following files: <ul style="list-style-type: none"> • Simulatorprofile.sta that is provided with the software and that you have copied to the computer. The file is located in the AFS folder in the root of the installation package of the software. • Your own Control Expert Control project file.
5	Click OK . Result: The simulator stops.
6	Start a simulator instance. Result: The simulator instance starts by loading the selected Control project file.

Obtaining the IP Address of the Simulator

If you leave 127.0.0.1 as IP address for the NIC of the station node, page 521 on which the simulator is running, it automatically defaults to the simulator that is installed on the system server computer. To use the simulator that is installed on a computer on which no system server is installed (for example, an engineering station on which only the engineering client is installed), you need to enter the actual IP address of the local computer on the subnet (host IP address).

To obtain the host IP address used by a simulator instance, proceed as follows.

Step	Action
1	Start a simulator instance.
2	Double-click the simulator icon in the notification area. Result: The Controller Simulator Panel is displayed.
3	Read the IP address in the Host IP address field.

Using Deployment and Execution Commands With the Simulator

The way to use the controller simulator is similar to using deployment and execution commands with a controller. You need to use the commands, page 517 that appear in the **Control** context menu of the station node that you have mapped to the Control project.

If you have created several ControlExpert services on the station node and mapped them to Control projects, a dialog box lets you select the Control project to which you want the command to apply.

Communication Between the Computer and Engines

Controllers

To deploy a built Control Participant project, ensure that the computer that is running the engineering client can communicate through the Ethernet network with the controller or its communication module, which is modeled by the entity that you want to deploy to.

For more information on connecting controllers to an Ethernet network, refer to the Control Participant help.

Station Nodes

To deploy a built Control Participant project, ensure that the computer that is running the system server can communicate through the Ethernet network with the computer, which is modeled by the station node that you want to deploy to.

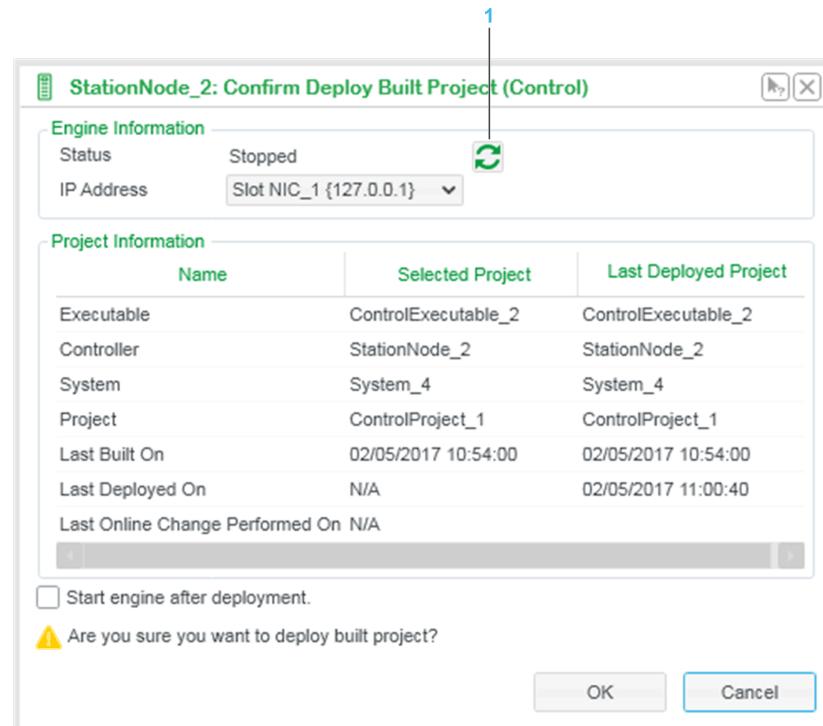
Refreshing Engine Status and Reconnecting Engines

When you use the various deployment and execution commands, a confirmation dialog opens in which you need to select the IP address of the target engine.

The engine needs to be reachable to be able to proceed with the deployment or execution command. Once you have selected the IP address, the status of the engine is indicated in the **Status** field. If the status of the engine changes while the confirmation dialog box is open, its status is not refreshed.

This can be the case, for example, if the status of the engine is detected as **Not Reachable** because a network cable is unplugged. After you reconnect the cable, the status is unchanged. Click the reconnect button to refresh the status of the engine and attempt to establish the connection.

The following figure shows an example of deployment confirmation dialog box featuring the reconnect button.



1 Reconnect button

Managing Passwords of Deployed Control Participant Projects

Overview

Using the **Controller/Simulator** password protection restricts deployment and execution operations for Control Participant projects that you perform by using an engineering client. It also restricts operations performed by using the operation client.

This topic describes how you can perform the following actions on controllers and the simulator running on a station node:

- Setting a password for the first time.
- Changing a password. This includes changing a password for a controller for which a password was already set by using other applications than the software.
- Clearing a password.
- Resetting the password if you forgot it.

Setting a password is mandatory if the **Controller Access Password at Creation**, page 137 (formerly **Optional Security Services By Default**) property of the system is enabled; otherwise, you cannot deploy Control Participant projects.

For an overview of the entire Control project password management functionality, refer to the topic describing the system and Control facet template passwords (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

NOTE: For information on the default password for the simulator (to be entered in the **Current Password** text box when you want to set a password), refer to the topic describing how to install the controller simulator (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

NOTE: Deploying a Control Participant project after setting or changing the password takes slightly longer than deploying with an existing or without password.

Scope of the Password Protection

The **Controller** password that you can set by using the commands in the context menu of the topological entity corresponds to the application password of the Control Participant project that is deployed.

The application password and file encryption settings are disabled in the Control Participant (**Protection** tab of the properties of the project).

NOTE:

The management of the other passwords of the Control Participant is not supported by EcoStruxure Process Expert, except for the following:

- The **System Access Password**, which is the application password, page 138 of Control Participant projects that are managed from the **Projects Explorer** (offline). This password does not replace **Controller** password when you deploy a Control project.
- The safe area password, page 770 for M580 safety controllers.

The table describes which Control projects managed by the software benefit of the **Controller/Simulator** password protection.

Control Participant project	Password protection applies
Logical Control project	No ⁽¹⁾
Built Control project	No ⁽¹⁾
Deployed Control project	Yes ⁽²⁾

(1) You need to use the **System Access Password** property, page 137 of the system to enable password protection and file encryption.

(2) File encryption is not enabled. To enable it, deploy a Control project after enabling system password protection.

Actions That Require Providing the Password

The table outlines the various actions for which you need to provide a password if **Controller/Simulator** password protection is enabled for the controller or simulator running on a station node respectively.

Topological entity	Action	Password that is required
Station node/ simulator only	Setting the Simulator password.	<p>The password that is used in the Control project file (.sta) that is loaded when the simulator starts.</p> <p>By default, this is the Control project file that is provided with the software (see EcoStruxure™ Process Expert, Installation and Configuration Guide).</p> <p>NOTE: The simulator may have been configured, page 655 to load another Control project file when starting or the password may have been changed, page 661.</p>
	First deployment.	You need to set a Simulator password first.
	Subsequent deployment and execution operations.	No password required as long as you keep the engineering client open.
Controller only	Setting the Controller password.	<p>No password is required.</p> <p>If a password is already set for the controller (for example, by using an application other than the software), the current password is required to be able to set a new one.</p> <p>NOTE: Setting the password when no Control Participant project is deployed to the controller requires stopping the controller.</p>
	First deployment.	You need to set the Controller password first.
	Any deployment and execution operation after setting or changing the password.	The current Controller password.
	Subsequent deployment and execution operations.	No password required as long as you keep the engineering client open.
	Navigating to a section by using Runtime Navigation Services (operation client).	<p>The current password.</p> <p>If you close the operation client or lock it and another user logs in, the current password is required again.</p>
	Resetting the password if you have forgotten it.	The temporary password provided by Schneider Electric support.

Topological entity	Action	Password that is required
Controller or station node/simulator/ ControlExpert service	Changing the Controller/Simulator password.	The Controller/Simulator password that is currently set.
	Clearing the Controller/Simulator password.	The Controller/Simulator password that is currently set.
	Using the Re-Deploy last Project command.	The Controller/Simulator password that is currently set. NOTE: If before redeploying the initial project (on which password A is set), another project on which password B is set and belonging to a different system has been deployed from a different client to the same engine, during re-deployment of the initial project, you need to enter password B. The deployment operation performed from a different system and client has changed the password that was set during the initial deployment (password A).
	Using the Back Up Data command for the first time after deploying a Control project.	The Controller/Simulator password that is currently set.
	Any deployment and execution operations after importing the controller or station node.	The password Controller/Simulator that is currently set. If no password is set yet, you need to set one (or, for a station node, provide the password that is used in the Control project file (.sta) that is loaded when the simulator starts).
	Any deployment and execution operations after restoring a system containing the controller or station node.	
	Any deployment and execution operations after restoring a database containing a system with the controller or station node.	
	Recovering from a signature mismatch.	The Controller/Simulator password that is currently set.
Disabling the Controller or Simulator password property when a password has been set.		The Controller/Simulator password that is currently set.
		For a station node, if several ControlExpert services exist, you need to enter the current password of each one. NOTE: Disabling the password property opens the Clear Password dialog box. Entering the password there clears it.

Password Requirements

To be valid, the **Controller** and **Simulator** passwords must contain the following:

- Between 8 and 16 characters.
- At least one uppercase letter from the classical Latin alphabet (A...Z).
- At least one lowercase letter from the classical Latin alphabet (a...z).
- At least one base-10 digit (0...9).
- At least one special character (~, !, @, \$, %, ^, &, *, _, +, -, =, ` , |, \, (,), [,], :, “ , ‘ , < , > , { , } , ; , #).

Managing Application Passwords with Redundant Controllers

The following are the specific password management rules that apply to redundant controllers:

- Both primary and standby engines of the topological entity must be reachable by using the IP addresses that you have configured and the Hot Standby cable link must be present.
- The same application password is set for the Control project of both engines. Changing or clearing the password also applies to both engines.
- When the **ControllerExist** property, page 665 of a redundant controller is set to **Primary**, passwords are managed the same way as for a standalone controller.

The following describes the operating mode for redundant controllers when a password mismatch occurs:

- If setting, changing, or clearing the password for the Control project of the standby controller did not succeed, you need to synchronize it by using the **Sync (Primary→Standby)** command. You can still proceed with the deployment to the primary controller.
- When you are prompted to provide the password for security purposes, the password is compared with that of the Control project in both engines. If the password of the project in the standby engine does not match, you cannot perform the operation on the standby engine.

Setting the Application Password of a Control Project

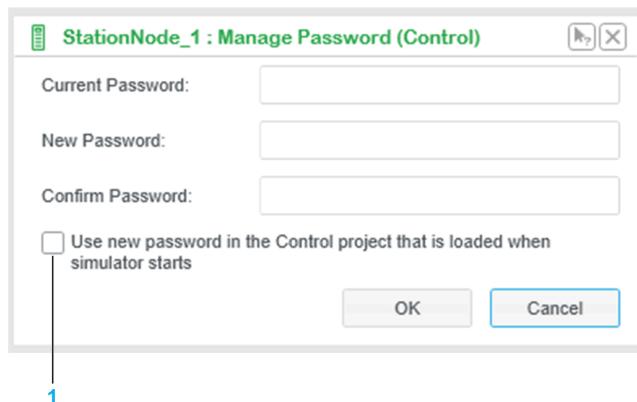
The following are prerequisites to be able to set the application password for a Control project:

- You have created a topological entity of type controller or station node.
- The topological entity is configured:
 - For a controller: You have used the **Configure** command and saved changes.
 - For a station node: You have created a ControlExpert service.
- The **Controller** or **Simulator** password property of the topological entity is enabled.
- The engine of the topological entity must be reachable by using the IP address that you have configured. In case of the simulator, it must be running.

You can set a password even if you have already deployed a Control project while the password protection for the topological entity was disabled. The password is set for the already deployed Control project without stopping the controller.

NOTE: You can use this functionality also to change the password of the Control project that is loaded when you start the simulator.

The following figure shows the **Manage Password** dialog box for the simulator running on a station node.



1 Check box that lets you also change the password, page 663 of the Control project that is loaded when you start the simulator.

To set the application password for a Control project, proceed as follows.

Step	Action
1	Right-click the topological entity and select: <ul style="list-style-type: none"> • For a controller: Manage Password For an M580 safety controller, select Controller in the submenu. • For a station node (simulator): Control > Manage Password If more than one ControlExpert service exists, you can select the one to use. Result: The Manage Password dialog box opens.
2	Enter the current password in the Current Password text box if required, page 659.
3	Enter the new password and confirm the new password in the corresponding text boxes.

Step	Action
4	If you want to use this password also in the Control project that is loaded when the simulator starts, select the check box. It replaces the password that is currently used in this Control project file.
5	Click OK . Result: The password is saved and set as application password.

Verifying the Password to Perform Deployment and Execution Tasks

When you select a command and are prompted to enter the password that is set for a controller or simulator running on a station node, proceed as follows.

Step	Action
1	In the Verify Password dialog box, enter the required password. NOTE: Alternatively, you can select the Use last entered password check box. It fills the text box with the last password that you have entered in a Verify Password dialog box during the current engineering client session. The text box becomes read-only.
2	Click OK . Result: If the password is correct, the command is executed. If not, an information message is displayed, prompting you to reenter the password. You can enter the password three times before the command is canceled. NOTE: Click Cancel to cancel the command you had selected.

Changing the Application Password of a Control Project

To be able to change the password, the engine of the topological entity must be reachable by using the IP address that you have configured.

To change the application password for a Control project, proceed as follows.

Step	Action
1	Right-click the topological entity and select: <ul style="list-style-type: none"> For a controller: Manage Password For an M580 safety controller, select Controller in the submenu. For a station node (simulator): Control > Manage Password If more than one ControlExpert service exists, you can select the one to use. Result: The Manage Password dialog box opens.
2	Enter the current password in the Current Password text box.
3	Enter the new password and confirm the new password in the corresponding text boxes.
4	If you want to use this password also in the Control project that is loaded when the simulator starts, select the check box, page 661. It replaces the password that is currently used in this Control project file.
5	Click OK . Result: The password is saved and set as application password.

Clearing the Application Password of a Control Project

To be able to clear the application password, the engine of the topological entity must be reachable by using the IP address that you have configured.

NOTE: Clearing the application password does not clear the password of the Control project that is loaded when the simulator starts.

To clear the application password for a Control project, proceed as follows.

Step	Action
1	<p>Right-click the topological entity and select:</p> <ul style="list-style-type: none"> • For a controller: Clear Password For an M580 safety controller, select Controller in the submenu. • For a station node (simulator): Control > Clear Password <p>If more than one password-protected ControlExpert service exists, you can select the one to use.</p> <p>Result: The Clear Password dialog box opens.</p>
2	Enter the current password in the Current Password text box.
3	<p>Click OK.</p> <p>Result: The password is cleared.</p>

Resetting a Forgotten Password

To reset the application password of a Control project, proceed as follows.

Step	Action
1	<p>Right-click the topological entity and select Forgot Password.</p> <p>For an M580 safety controller, select Controller in the submenu.</p> <p>Result: The Forgot Password dialog box opens and an authentication code is displayed.</p> <p>NOTE: If you close the Forgot Password dialog box and open it again, the same authentication code is displayed.</p>
2	<p>Contact Schneider Electric support, page 104 and provide the authentication code.</p> <p>Result: You receive a temporary password.</p> <p>NOTE: You can use the temporary password as long as you do not modify the application that is deployed to the controller.</p>
3	<p>Right-click the topological entity and select Manage Password.</p> <p>For an M580 safety controller, select Controller in the submenu.</p> <p>Result: The Manage Password dialog box opens.</p>
4	Follow the procedure to change the password, page 662 and enter the temporary password in the Current Password text box.

Changing the Password of the Control Project Loaded at Simulator Start-Up

You can change the password of the default Control project file or your own file, page 655 that is loaded when the simulator starts if the functionality is enabled.

The folder containing the project file must not be write-protected.

The table indicates from where you can change the password of the Control project file depending on the location of the controller simulator.

Computer on which the controller simulator is running	Software component on which you need to perform the password change
System server computer.	<p>Any engineering client.</p> <p>The IP address of the NIC of the station node in the Topology Explorer must correspond to the system server computer.</p>
Engineering client computer.	The local engineering client.

Computer on which the controller simulator is running	Software component on which you need to perform the password change
Operation client computer.	
Other computer on which the controller simulator was installed manually.	<ol style="list-style-type: none"> 1. Copy the project file loaded in the controller simulator to the local engineering client computer (if the project file loaded in both simulators is different; otherwise, proceed to step 3). 2. Configure the simulator, page 655 on the local engineering client computer to load this project file at start-up. 3. Change the password of the project file by using the local engineering client. 4. Replace the project file on the operation client or the other computer.

To change the password of the Control project file that is loaded when the simulator starts, proceed as follows.

Step	Action
1	Start the controller simulator or restart it.
2	<p>Right-click the station node that represents the computer on which the controller simulator is running and select Control > Manage Password.</p> <p>If more than one ControlExpert service exists, you can select the one to use.</p> <p>Result: The Manage Password dialog box opens.</p>
3	In the Current Password text box, enter the password of the Control project file that is loaded when the simulator starts.
4	Enter the new password and confirm the new password in the corresponding text boxes.
5	Select the check box, page 661.
6	<p>Click OK.</p> <p>Result: The password is changed in Control project file.</p> <p>NOTE: The change does not apply to other simulator instances that are already running until you restart them.</p>

Deploying to Redundant Controllers

Overview

This topic describes aspects that are specific to the deployment of Control Participant projects to redundant controllers, which are modeled in the topology of the system.

The same commands that are used on standalone configurations are also used for redundant controllers.

Detailed information on the commands that are available as part of the deployment stage are described later in the chapter.

For a description of the sequence of actions that Control performs on the engines, refer to the topic describing main redundant controller scenarios, page 741.

ControllerExist Property

Verify that the value of the **ControllerExist** property, page 509 that is selected in the properties of the redundant controller entity matches the physical configuration of the hardware.

If the software detects a mismatch, you cannot proceed with the deployment.

NOTE: Selecting **PrimaryAndStandby** for the **ControllerExist** property requires that both controllers are reachable and connected by a Hot Standby cable link.

NOTE: The value of the property also affects execution commands, page 707.

Engine Status for Redundant Controllers

When you select a command for a redundant controller, composed of one primary and one standby engine to which IP addresses *IP* and *IP + 1* are assigned respectively, before proceeding with the deployment, the software opens various dialog boxes, depending on the command that you have selected.

Each of these dialog boxes features the **Status** field, which informs you of the current state of each of the engine roles (A and B) that the software has detected.

States for M580 platform controllers	States for Quantum platform controllers
Run Primary	Run Primary
Run Standby	Run Standby
Wait	Run Offline
Stop	Stop Offline
No conf (non-configured state)	No conf (non-configured state)

NOTE: If you have selected **Primary** for the **ControllerExist** property of the controller entity, the status of the standby engine is not available.

General Deployment Behavior for Redundant Controllers

When you deploy to a redundant controller:

- The software deploys to both the primary and the standby engines unless you have selected **Primary** for the **ControllerExist** property of the controller entity. You can view and/or select the IP address for the primary engine (*IP*) and the software automatically selects IP address *IP + 1* as the standby engine.
- If you select to start the engines after deployment, both engines are started.

If deployment to the first engine (*IP* or *IP + 1*, depending on the status of the engine, page 741) does not complete successfully, the entire deployment process is canceled.

If deployment to the second engine (*IP* or *IP* + 1, depending on the status of the engine) or if synchronization does not complete successfully, logic that was deployed successfully to the first engine is preserved.

NOTE: For controllers of the M580 platform, you cannot perform deployment tasks on the CPU module or a NOC communication module by using *IP address A* and *IP address B*. You can only select main IP address, page 683.

Redundant Controller Stop Sequence

When deployment requires to stop the controller and both engines are running, the software stops in the following order:

- The standby engine
- The primary engine

Deploying Changes to Redundant Controllers

If the primary controller is in **Running Primary** state, and you deploy changes that do not require stopping the controller by using the **Deploy Changes / Undo Online Changes** command, page 682:

- The software enables the configuration mismatch feature of the controller so that the standby controller does not enter the offline operating mode when changes are deployed to the primary controller.
- The software deploys to the primary controller first, and upon successful completion of the deployment, automatically synchronizes the standby controller.
- Then, the software disables the configuration mismatch feature.

When you deploy changes to a redundant controller, although deployment to the primary controller succeeded, it is possible that synchronization of the standby controller, page 744 does not complete. If an interruption of the connection to the system server or an unexpected stop of the system server is the cause, the notification panel also contains additional information about steps that you must complete when the connection to the system server is re-established. This is the case, for example, when the software cannot disable the configuration mismatch feature. If this feature is enabled and a switchover occurs, the standby controller operates with a program that does not contain the changes that you have deployed to the primary controller.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that deployment to the standby controller completed successfully.
- Do not operate the primary controller if deployment to the standby controller does not complete successfully.
- Synchronize the standby controller manually, using the **Sync (Primary→Standby)** command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: If the primary controller is not in **Running Primary** state, synchronization of the standby controller cannot complete successfully. In such case, starting, page 713 the primary engine synchronizes the standby engine.

Deploying the Control Participant Project

Overview

The **Topology Explorer** allows you to deploy the complete Control Participant project to a controller or station node that is modeled by an engine in the topology of the system by using the following commands:

- **Deploy Built Project:** Deploys the built Control Participant project to the controller or Control service of the station node that is mapped to the corresponding executable, page 580.
- **Re-Deploy Last Project:** Deploys the same built Control Participant project that you have deployed to the controller/Control service of the station node last. This is done by using a mirror of the deployed Participant project, called the last deployed reference, which is stored in the contents repository, page 935.

To deploy or redeploy a built Participant project, the controller or the emulated simulator that you are deploying to must be in **Stop** state before proceeding. If it is not, the software informs you of it, and stops the engine when you proceed with the deployment.

Also, the software gives you the option to start the controller following the deployment.

Built Participant Project Build Status

The software lets you deploy a built Control Participant project whose executable is either marked as being out of date or that you have not built again after making changes to the system, page 636, which affect the built Participant project.

The software displays a message in the confirmation dialog boxes of deployment commands if the **Build State**, page 634 of the associated executable is **Out Of Date**.

Selecting Participant Project and Control Executable

When you select the **Deploy Built Project** command for an engine, the software displays the **Deploy Built Project** dialog box, which allows you to select:

- **Project:** Any Control project that exists in the system and:
 - Whose Controller family, page 260 property has the same value as the engine.
 - Contains a built or out of date executable that is mapped to the engine.
- **Executable:** For each project, any Control **executable**, page 578 that is mapped to the engine, page 580.

NOTE: When you select **Deploy Built Project** from a station node, you can select any Control project that has a Control executable that is mapped to a Control service of this station node.

Deploying to a Redundant Controller

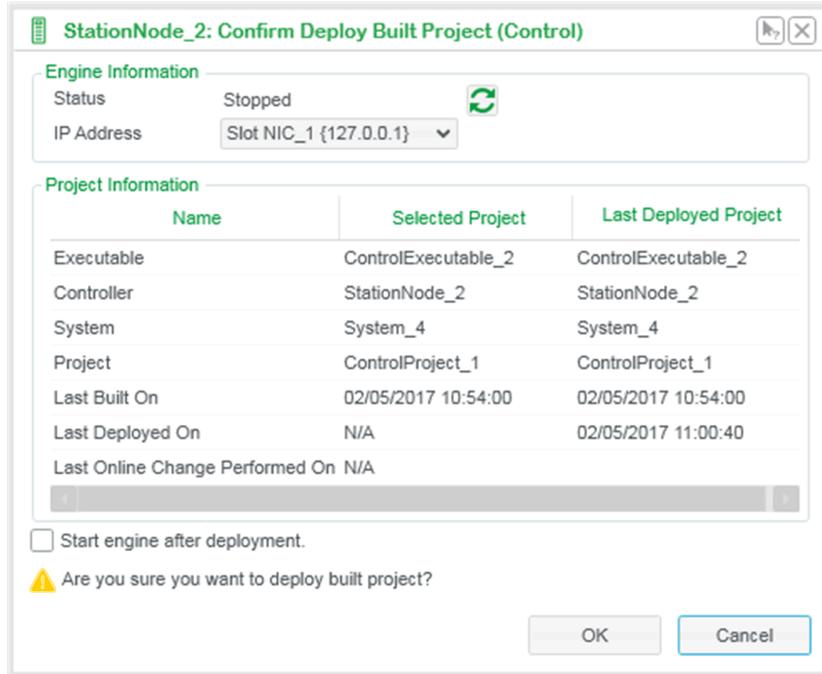
For information that applies specifically to the deployment to redundant controllers, refer to the related topic, page 665.

Control Participant Project Information

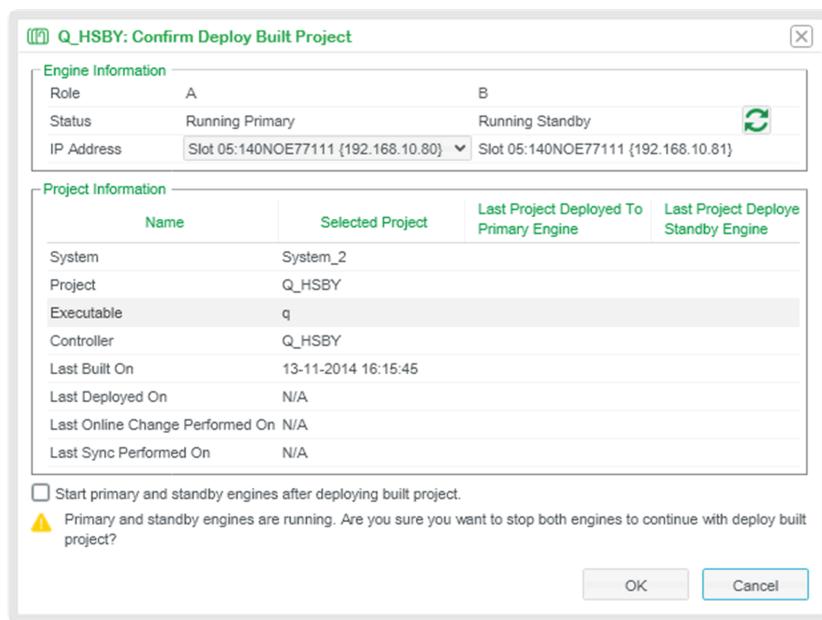
Before proceeding with the deployment or redeployment of a built Participant project, the software displays information in the **Confirm Deploy Built Project** dialog box about:

- The engine that you are deploying to.
- The built Participant project to be deployed.

- The Participant project deployed last to the engine.



The figure shows an example of the **Confirm Deploy Built Project** dialog box for a redundant controller.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm Deploy Built Project** dialog box.

Label	Description
Role⁽¹⁾	Role of the engine: A or B . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for deployment. If the status is Not Reachable you cannot proceed with the deployment. In such case, verify the IP address, the connection to the engine, and the status of the engine. NOTE: If the status is Running , a notification is displayed in the dialog box to inform you that the engine will be stopped if you proceed with the deployment.

Label	Description
IP Address	<p>Identifier of the communication module through which the software will deploy the built Participant project and its IP address that is configured in the Topology Explorer.</p> <p>If the topological entity has several communication modules, you can select which one you want to deploy to.</p> <p>NOTE: For controllers of the M580 platform, you cannot deploy to the CPU module or a NOC communication module by using <i>IP address A</i>. You can only select main IP address, page 683.</p> <p>NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.</p>
(1) Displayed only when you deploy a built Participant project to a redundant controller.	

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Deploy Built Project** dialog box. It also describes the check box.

Name column	Selected Project column	Last Deployed Project column ⁽²⁾
System	Identifier of the system that both the engine and the Control project, which you have selected belong to.	Identifier of the system that both the engine and the Control project, from which you have deployed the Participant project last belong to.
Project	Identifier of the Control Participant project that you have selected.	Identifier of the Control Participant project associated to the executable that you have deployed last.
Executable	Identifier of the Control executable that you have selected and that contains the built Participant project to be deployed.	Identifier of the Control executable that contains the built Participant project that you have deployed last.
Controller	Identifier of the topological entity to which you are deploying the built Participant project.	Identifier of the topological entity to which you have deployed the built Participant project last.
Last Built On	Date and time when you successfully built the Control Participant project to be deployed.	Date and time when you successfully built the Control Participant project that you have deployed last.
Last Deployed On	Not applicable	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.
Last Online Change Performed On	Not applicable	Date and time when you last changed the deployed Participant project by using the Refine Online command; otherwise the field is empty.
Last Sync. Performed On⁽¹⁾	Not applicable	Date and time when you last synchronized the standby controller with the primary one by using the Synchronize (Primary→Standby) command.
Start engine after deployment	Select the option if you want to start the controller or emulated simulator when deployment is complete; otherwise the engine remains in stop state. NOTE: The software stops the engine before proceeding with the deployment if the engine is not yet in stop state.	
(1) Displayed only when you deploy a built Participant project to a redundant controller.		
(2) When you deploy to a redundant controller, the dialog box shows separate columns for the primary and the standby engine.		

NOTE: When you deploy for the first time to an engine, the fields of the **Last Deployed Project** section are empty.

Deploying the Control Participant Project

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before deploying to a controller, verify that:

- You have assigned a valid IP address to the controller.
- The IP address corresponds to the physical address printed on the hardware.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For more information, refer to Identifying Engines, page 651.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before deploying to an engine, verify that you have selected the correct project files.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING

UNKNOWN OPERATIONAL STATE OF EQUIPMENT

Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

When deploying to a redundant controller, deployment to the standby controller, page 744 may not complete successfully. In such case, the software displays a message in the notification panel to inform you.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that deployment to the standby controller completed successfully.
- Do not operate the primary controller if deployment to the standby controller does not complete successfully.
- Synchronize the standby controller manually, using the **Sync (Primary→Standby)** command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For information on the **Sync (Primary→Standby)** command, refer to the Execution stage, page 707.

To deploy the Control Participant project, proceed as follows.

Step	Action
1	Verify that the engine can communicate with the PC, page 657 and that it is in Stopped state.
2	In the Topology Explorer , right-click the entity to which you want to deploy the built Participant project and select Deploy Built Project . Result: The Deploy Built Project dialog box opens.
3	Click the menu next to Project and select the Control project.
4	Click the menu next to Executable and select the executable associated to the built Control project that you want to deploy.
5	Click OK . Result: The Confirm Deploy Built Project dialog box opens. NOTE: Click Cancel to close the dialog box without deploying.
6	Verify the information that is displayed.
7	Verify the status of the engine in the Status field.
8	Verify the communication module identification or select one in the IP Address field. NOTE: When deploying to a controller simulator, verify that it is running, page 654 and using the same port number as the one shown in the IP Address field.
9	Select Start engine after deployment if you want the software to start the engine after the deployment is complete.
10	Click OK . Result: The Reconfirm Deploy Built Project dialog box opens. NOTE: Click Cancel to close the Confirm Deploy Built Project dialog box without deploying.
11	Click OK . Result: The software: <ul style="list-style-type: none">• Stops the engine if it is running.• Deploys the Control Participant project.• Displays Completed in the notification panel once it has successfully completed the deployment process to the engine, which is identified by its IP address.<ul style="list-style-type: none">◦ For a redundant controller, the notification panel displays separate messages for the primary and the standby engines by using their respective IP address to identify them. If deployment to either engine does not complete successfully, the software displays a corresponding message in the notification panel.• Updates the related date/time field.• Starts the controller (cold start) or software emulating a simulator if you selected the option. NOTE: Click Cancel to close the Reconfirm Deploy Built Project dialog box without deploying. This reverts to the Confirm Deploy Built Project dialog box.

NOTE: If the deployment process does not complete successfully and the software has stopped the engine before deploying the Participant project, the engine remains in **Stopped** state.

Re-Deploying the Control Participant Project

Overview

The **Re-Deploy Last Project** command for an engine becomes available once you have deployed a project to this engine.

The process of redeploying the Control Participant project is identical to the deployment process, page 672 with the exception that you cannot select a Control project and an associated Control executable for deployment. The software deploys to the controller or station node emulating a simulator a mirror image of the last deployed Participant project, which is the last deployed reference.

To redeploy a built Participant project, the engine that you are deploying to, must be in **Stopped** state before proceeding. If it is not, the software informs you of it, and stops the engine when you proceed with the deployment.

NOTE: If you made changes to the deployed Participant project using the **Refine Online** and/or if you have deployed changes using the **Deploy Changes** commands, then the Participant project that the software redeploys to the controller contains these changes.

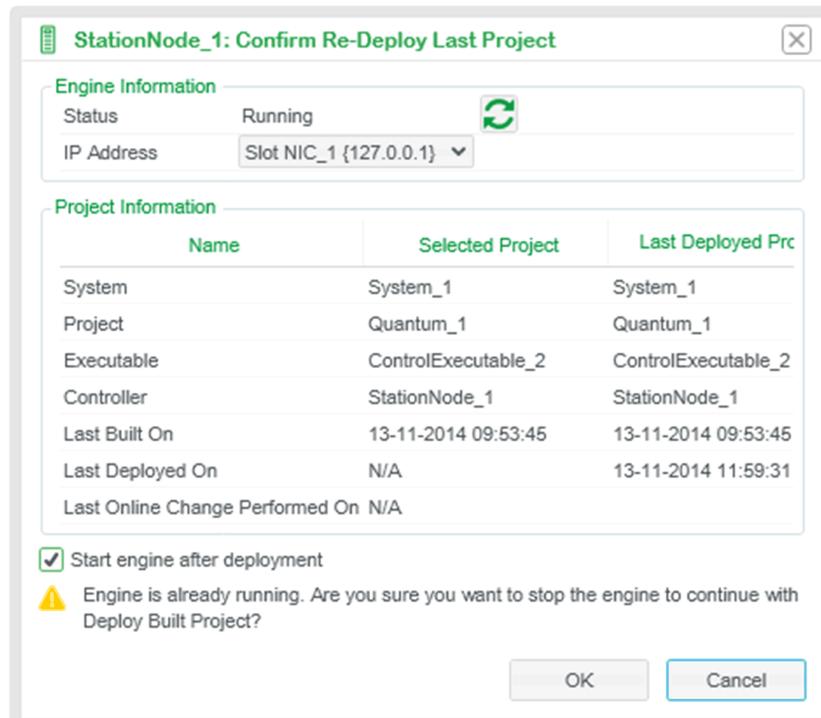
Redeploying to a Redundant Controller

For information that applies specifically to the redeployment to redundant controllers, refer to the topic describing deployment to redundant controllers, page 665.

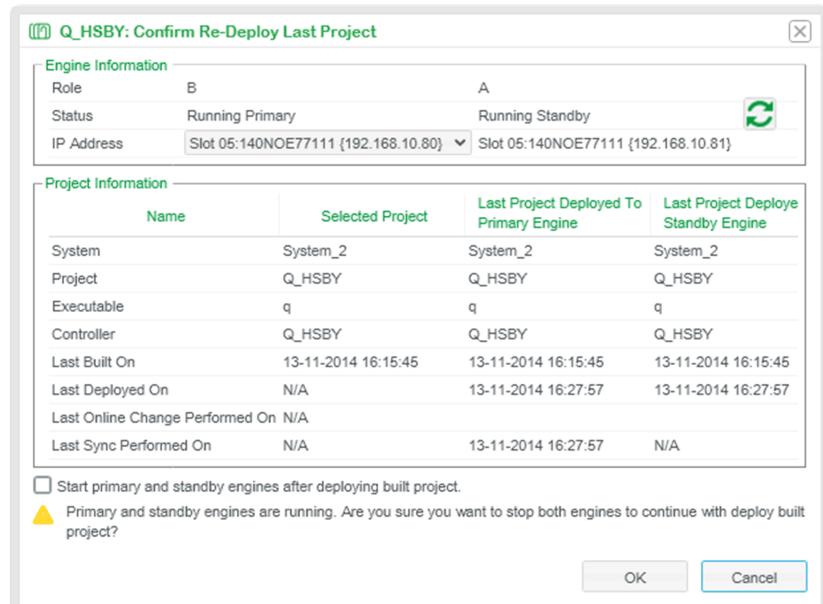
Control Participant Project Information

Before proceeding with the re-deployment of the last deployed reference, the software displays information in the **Confirm Re-Deploy Last Project** dialog box about:

- The engine that you are re-deploying to.
- The project to be re-deployed.
- The Participant project deployed last to the engine.



The figure shows an example of the **Confirm Re-Deploy Last Project** dialog box for a redundant controller.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm Re-Deploy Last Project** dialog box.

Label	Description
Role⁽¹⁾	Role of the engine: <i>A</i> or <i>B</i> . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for deployment. If the status is Not Reachable you cannot proceed with the deployment. In such case, verify the IP address, the connection to the engine, and the status of the engine. NOTE: If the status is Running , a notification is displayed in the dialog box to inform you that the engine will be stopped if you proceed with the deployment.
IP Address	Identifier of the communication module through which the software will deploy the built Participant project and its IP address that is configured in the Topology Explorer . If the topological entity has several communication modules, you can select which one you want to deploy to. Default value: IP address that was selected for the last deployment operation to this engine. NOTE: For controllers of the M580 platform, you cannot redeploy to the CPU module or a NOC communication module by using <i>IP address A</i> . You can only select main IP address, page 683. NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.

(1) Displayed only when you redeploy a built Participant project to a redundant controller.

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Re-Deploy Built Project** dialog box. It also describes the check box.

Name column	Selected Project column	Last Deployed Project column ⁽²⁾
System⁽¹⁾	Identifier of the system that both the engine and the Control project, which you have selected belong to.	
Project⁽¹⁾	Identifier of the Control Project that you have selected.	
Executable⁽¹⁾	Identifier of the Control executable that you have selected and that contains the built Participant project to be deployed.	
Controller⁽¹⁾	Identifier of the controller/station node to which you are deploying the built Participant project.	
Last Built On	Date and time when you successfully built the Control Participant project to be deployed.	Date and time when you successfully built the Control Participant project that you have deployed last.
Last Deployed On	Not applicable	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.
Last Online Change Performed On	Not applicable	Date and time when you last changed the deployed Participant project by using the Refine Online command; otherwise the field is empty.
Last Sync. Performed On⁽³⁾	Date and time when you last synchronized the standby controller with the primary one by using the Synchronize (Primary→Standby) command.	
Start engine after deployment	<p>Select the option if you want to start the engine when deployment is complete; otherwise the engine remains in stop state.</p> <p>NOTE: The software stops the engine before proceeding with the deployment if the engine is not yet in stop state.</p>	
<p>(1) The values of the Selected Project and Last Deployed Project fields are identical.</p> <p>(2) When you redeploy to a redundant controller, the dialog box shows separate columns for the primary and the standby engines.</p> <p>(3) Displayed only when you redeploy to a redundant controller.</p>		

Re-Deploying the Last Deployed Reference

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before deploying to a controller, verify that:

- You have assigned a valid IP address to the controller.
- The IP address corresponds to the physical address printed on the hardware.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For more information, refer to Identifying Engines, page 651.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before deploying to an engine, verify that you have selected the correct project files.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING

UNKNOWN OPERATIONAL STATE OF EQUIPMENT

Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

When redeploying to a redundant controller, re-deployment to the standby controller, page 744 may not complete successfully. In such case, the software displays a message in the notification panel to inform you.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that deployment to the standby controller completed successfully.
- Do not operate the primary controller if deployment to the standby controller does not complete successfully.
- Synchronize the standby controller manually, using the **Sync (Primary→Standby)** command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For information on the **Sync (Primary→Standby)** command, refer to the Execution stage, page 707.

To redeploy the last deployed reference, proceed as follows.

Step	Action
1	Verify that the engine can communicate with the PC, page 657 and that it is in Stopped state. NOTE: When re-deploying to a controller simulator, verify that it is running, page 654 and using the same port number as the one shown in the IP Address field.
2	In the Topology Explorer , right-click the entity to which you want to deploy the built Participant project and select Re-Deploy Last Project . Result: The Confirm Re-Deploy Last Project dialog box opens.
3	Verify the information that is displayed.
4	Verify the status of the engine in the Status field.
5	Select Start engine after deployment if you want the software to start the engine after the redeployment is complete.

Step	Action
6	<p>Click OK.</p> <p>The software displays the Reconfirm Re-Deploy Last Project dialog box.</p> <p>NOTE: Click Cancel to close the Confirm Re-Deploy Last Project dialog box without redeploying.</p>
7	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> • Redeploys the last deployed reference to the engine. • Displays Complete in the notification panel once it has successfully completed the re-deployment process to the engine, which is identified by its IP address. <ul style="list-style-type: none"> ◦ For a redundant controller, the notification panel displays separate messages for the primary and the standby engines by using their respective IP address to identify them. If redeployment to either engine does not complete successfully, the software displays a corresponding message in the notification panel. • Updates the related date/time field. • Starts the controller (cold start) or software emulating a simulator if you selected the option. <p>NOTE: Click Cancel to close the Reconfirm Re-Deploy Last Project dialog box without re-deploying. This reverts to the Confirm Re-Deploy Last Project dialog box.</p>

NOTE: If the redeployment process does not complete successfully and the software has stopped the engine before redeploying the Participant project, the engine remains in **Stopped** state.

Deploying Data to Controllers and Controller Simulators

Overview

The **Deploy Data** command lets you deploy data contained in a backup file, page 733 that has been created by using the **Back Up Data** command.

You can select one file from a list of backup files that have been created for all executables of the same Control Participant project. The selection dialog shows the information that was associated to each backup file at the time of creation, including the type of data that was selected.

By default, backup files are listed in descending order of time stamp (most recent one on top).

NOTE: For more information, refer to *Save/Restore Data Between a File and the PLC* in the help of the Control Participant, page 93.

Deploying Data to Redundant Controllers

For redundant controllers, deployment of data is possible only to the primary controller.

To be able to deploy data to the primary controller, the standby controller must be reachable at the IP address that is configured. It may not be reachable, for example, because it is switched off or the Hot Standby cable is removed.

Only if the **ControllerExist** property, page 509 is set to **Primary** you can deploy data to the primary controller while the standby controller is not reachable.

NOTE: The status of the standby controller is not shown in the **Confirm Deploy Data** dialog box.

Data Backup File Information

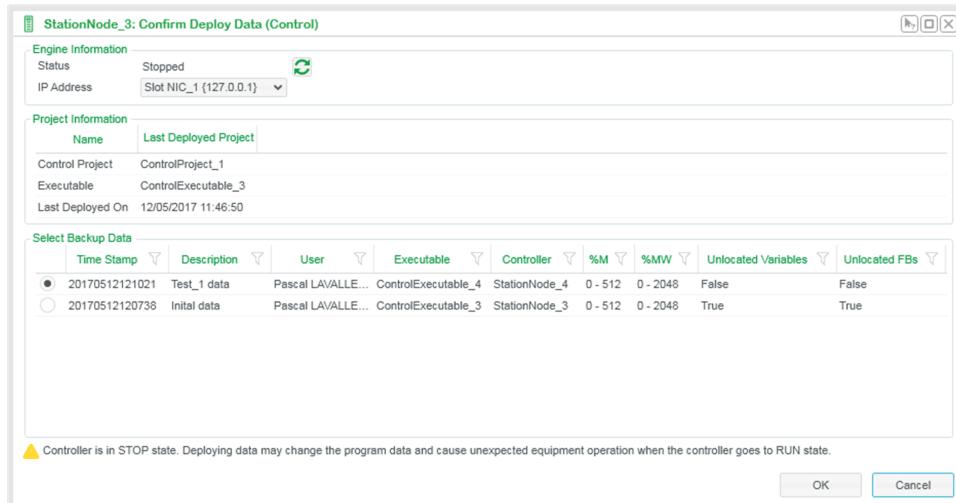
Before proceeding with the deployment of controller data, the software displays information in the **Confirm Deploy Data** dialog box about:

- The deployed Control Participant project.
- The executable that is deployed.
- Executables of this Control project for which a backup file exists.
- The type of data contained in each backup file.
- The backup file identification.

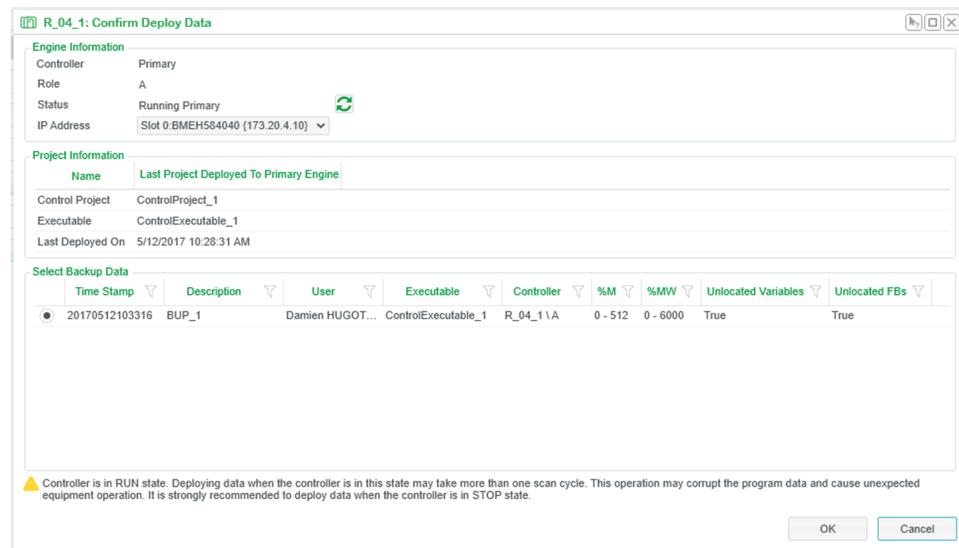
NOTE: If you change the identifier of a controller or executable, or the role of a controller, the identification information of the corresponding backup file is not modified.

You can sort and filter, page 118 the information that is displayed.

The following figure shows an example of the **Confirm Deploy Data** dialog box for a standalone controller.



The following figure shows an example of the **Confirm Deploy Data** dialog box for a redundant controller.



The table describes the information that is displayed in the **Engine Information** section of the **Deploy Data** dialog box.

Label	Description
Controller⁽¹⁾	You can deploy controller data only to the primary controller. Default value: Primary
Role⁽¹⁾	Role of the engine: A or B . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for deployment of data. If the status is Not Reachable , you cannot proceed with the deployment. In such case, verify the IP address, the connection to the engine, and the status of the engine.

Label	Description
IP Address	<p>Identifier of the communication module through which the software will deploy data and its IP address that is configured in the Topology Explorer.</p> <p>If the topological entity has several communication modules, you can select which one you want to use.</p> <p>Default value: IP address that was selected for the last deployment operation to this engine.</p> <p>NOTE: For controllers of the M580 platform, you cannot deploy data to the CPU module or through a NOC communication module by using <i>IP address A</i>. You can only select main IP address, page 683.</p> <p>NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.</p>

(1) Displayed only when you deploy data to a redundant controller.

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Deploy Data** dialog box.

Label	Description
Control Project	Identifier of the Control Participant project associated to the executable that you have deployed last.
Executable	Identifier of the Control executable to which you are deploying data.
Last Deployed On	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.

The table describes the information that is displayed in the **Select Backup Data** section of the **Confirm Deploy Data** dialog box.

Column header	Description
Time Stamp	Date and time of the backup file creation.
Description	Description that was entered when the backup file was created.
User	Name of the user who was logged on when the backup file was created.
Executable	Identifier of the Control executable whose data is contained in the backup file.
Controller	<p>Identifier of the controller that was running the executable whose data was backed up.</p> <p>For redundant controllers, it also indicates the role of the controller that was selected when the backup file was created.</p>
%M	<p>Indicates the memory range of located variables of boolean data type that are contained in the backup file.</p> <p>If no such data was selected during data backup, the value shown is N/A.</p>
%MW	<p>Indicates the memory range of located variables of WORD data type that are contained in the backup file.</p> <p>If no such data was selected during data backup, the value shown is N/A.</p>
Unlocated Variables	<p>True indicates that this type of data was selected during data backup.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • True • False
Unlocated FBs	

Control Project Signature Verification

Before executing the command, the software compares the Control Participant project that is deployed in the engine with the last deployed reference, page 650, which is the Control Participant project residing in the software.

If they are identical, the command is executed. If they differ, the software starts a recovery process first. This process may take several minutes to complete.

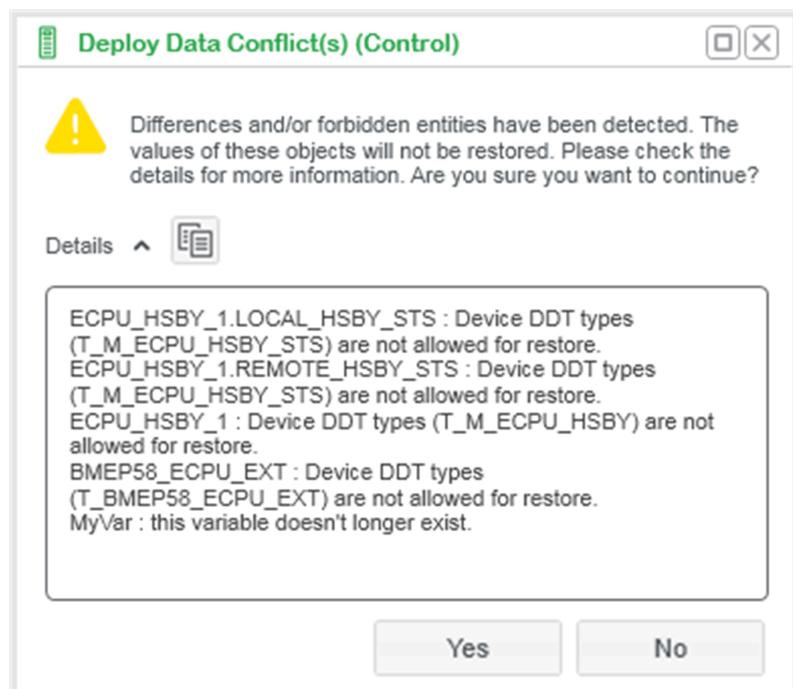
In case the software displays a notification about a signature mismatch, the command cannot be executed. Contact Schneider Electric support, page 104.

Deployed Data Conflicts

Before proceeding with the deployment of data contained in the selected backup file, the software opens a dialog box in case it detects any conflicts in the data to be deployed. It displays the data instances that cannot be deployed.

You have the choice to proceed with the deployment but without the conflicting data instances or to cancel the deployment operation entirely.

The following figure shows an example of the **Deploy Data Conflicts** dialog box.



Deploying Data Contained in a Backup File

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

Deploying data when the controller is in `RUN` state may take several scan cycles and corrupt the program data. This adversely affects the execution of the program.

Further, deploying data to the controller may change program data. This affects the execution of the program when the controller is in `STOP` state and then switched to `RUN` state after the deployment.

WARNING

UNINTENDED EQUIPMENT OPERATION

Before deploying data to the controller:

- Verify that the controller state does not jeopardize data integrity when the data is deployed.
- Verify the impact of the deployment on the execution of the program.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: Schneider Electric recommends deploying data while the controller is in **STOP** state.

To deploy data contained in a backup file to a controller or controller simulator, proceed as follows.

Step	Action
1	<p>Verify that the engine can communicate with the PC, page 657.</p> <p>NOTE: When deploying data to a controller simulator, verify that it is running, page 654 and using the same port number as the one shown in the IP Address field.</p>
2	<p>In the Topology Explorer, right-click the entity to which you want to deploy data and select Deploy Data.</p> <p>Result: The Confirm Deploy Data dialog box opens.</p>
3	<p>In the IP Address field, verify that the communication module identification is correct.</p>
4	<p>Verify the status of the engine in the Status field.</p> <p>NOTE: Deployment is possible only if the state of the engine is indicated as either RUN, STOP (recommended), or WAIT (OFFLINE for Hot Standby controllers of the Quantum platform).</p>
5	<p>Verify the information that is displayed in the Project Information section.</p>
6	<p>Select a backup file.</p>
7	<p>Click OK.</p> <p>Result: The Reconfirm Deploy Data dialog box opens.</p> <p>NOTE: Click Cancel to close the Confirm Deploy Data dialog box without deploying data.</p>
8	<p>Click OK.</p> <p>Result:</p> <ul style="list-style-type: none"> • If the software detects conflicts in the data to be deployed, it opens the Deploy Data Conflicts dialog box, page 680, which lets you proceed with the deployment (Yes) or cancel it (No). • If you select to proceed, the software deploys the data and provides information about the deployment operation in the notification panel (or in a dialog box in case deployment did not succeed). <p>NOTE: Click Cancel to close the Reconfirm Deploy Data dialog box without deploying data.</p>

Deploying Control Project Changes

Overview

Following a deployment, the **Deploy Changes / Undo Online Changes** command allows you to deploy to an engine changes that you have made to the logical Control Participant project that corresponds to the built Participant project that you had deployed to the same engine last.

It lets you also deploy CCOTF-compatible changes, page 505 made to the configuration of a controller.

You can deploy changes pertaining to the following categories:

- Animation tables
- Operator screens
- Sections
- Types
- Variables
- Hardware

In addition, for M580 safety Control projects, page 759:

- Process (non-safe) program interfaces
- Safe program
- Safety password

To deploy other changes, refer to Changes that Cannot be Deployed, page 686.

When you select the **Deploy changes / Undo Online Changes** command, the software compares the deployed Participant project that is currently loaded in the engine with the corresponding built Participant project and identifies:

- Control project and configuration changes that you can deploy without stopping the engine.
- Control project changes that require stopping the engine, page 692. The software displays a dialog box informing you that an engine stop is required in order to deploy such changes.

You can select, which of the identified changes you want to deploy.

The deployment of changes executes the CCOTF transaction and resets the counter.

The software gives you the option to proceed with a cold start of the engine following the deployment of changes that require stopping the engine.

NOTE:

- You can use the **Deploy changes / Undo Online Changes** command to discard changes that you made to the deployed Participant project.
- You need to build the Control project, page 636 before you can deploy changes by using the **Deploy changes / Undo Online Changes** command.
- You cannot view the result of the deployment of changes before the deployment is actually completed.

Control Project Signature Verification

Before executing the command, the software compares the Control Participant project that is deployed in the engine with the last deployed reference, page 650, which is the Control Participant project residing in the software.

If they are identical, the command is executed. If they differ, the software starts a recovery process first. This process may take several minutes to complete.

In case the software displays a notification about a signature mismatch, the command cannot be executed. Contact Schneider Electric support, page 104.

Undoing Online Changes

If you have changed the logic of the deployed Participant project by using the **Refine Online** command, you can use **Deploy changes / Undo Online Changes** to undo these changes. This is possible only if the following conditions are satisfied:

- You have not changed the configuration online.
- You have not yet updated the Control project with these online changes.

The software identifies the changes you made online and displays them in the **Deploy Changes / Undo Online Changes** dialog box. After executing the command, the deployed Control Participant project will be identical to the built Participant project again.

NOTE: If you have changed and built the logical Control Participant project since the deployment, these changes will be applied to the deployed Participant project as well unless you unselect them. It is not possible to unselect CCOTF-compatible hardware changes.

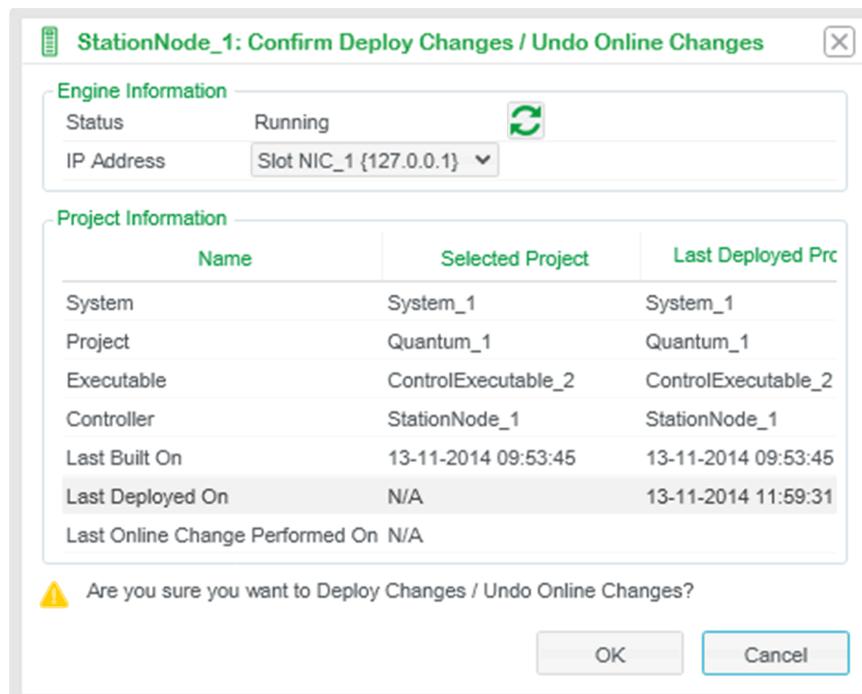
Deploying Changes to a Redundant Controller

For information that applies specifically to the deployment of changes to redundant controllers, refer to the topic describing deployment to redundant controllers, page 665.

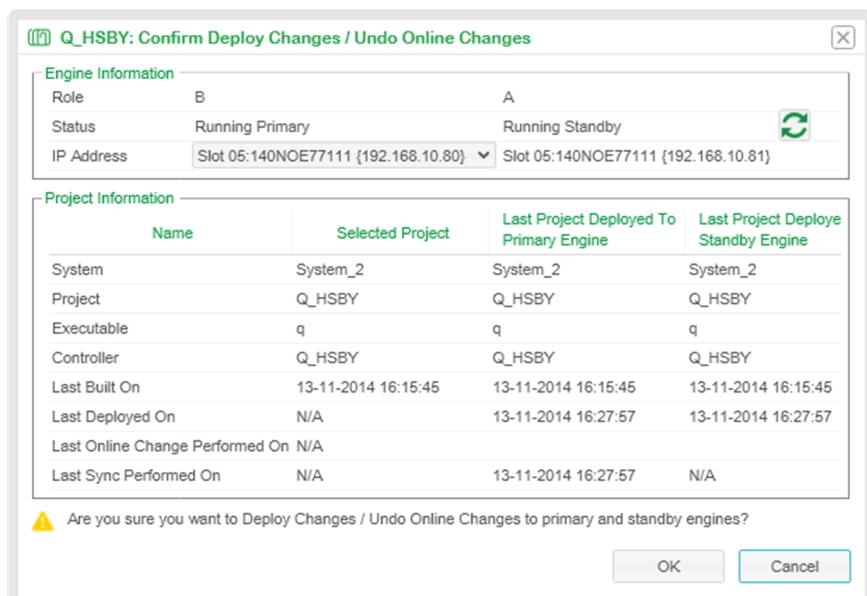
Deployed Project Information

When you select the **Deploy Changes / Undo Online Changes** command, before you can select the changes that you want to deploy, the software displays the **Confirm Deploy Changes / ...** dialog box, which contains information about:

- The engine that you are deploying changes to.
- The Control Participant project that is deployed to the engine.



The figure shows an example of the **Confirm deploy changes / Undo Online Changes** dialog box for a redundant controller.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm deploy changes / Undo Online Changes** dialog box.

Label	Description
Role⁽¹⁾	Role of the engine: <i>A</i> or <i>B</i> . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for deployment. If the status is Not Reachable you cannot proceed with the deployment. In such case, verify the IP address, the connection to the engine, and the status of the engine. NOTE: If the status is Running , a notification is displayed in the dialog box to inform you that the engine will be stopped if you proceed with the deployment.
IP Address	Identifier of the communication module through which the software will deploy the built Participant project and its IP address that is configured in the Topology Explorer . If the topological entity has several communication modules, you can select which one you want to deploy to. Default value: IP address that was selected for the last deployment operation to this engine. NOTE: For controllers of the M580 platform, you cannot deploy changes to the CPU module or a NOC communication module by using <i>IP address A</i> . You can only select main IP address, page 683. NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.
(1) Displayed only when you select the command for a redundant controller.	

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Deploy Changes / Undo Online Changes** dialog box. It also describes the check box.

Name column	Selected Project column	Last Deployed Project column ²
System	Identifier of the system that both the engine and the Control project, which you have selected belong to.	Identifier of the system that both the engine and the Control project, from which you have deployed the Participant project last belong to.
Project	Identifier of the Control Project that you have selected.	Identifier of the Control Project associated to the deployed Participant project that you have deployed last.
Executable	Identifier of the Control executable that you have selected and that contains the built Participant project to be deployed.	Identifier of the Control executable that contains the built Participant project that you have deployed last.
Controller	Identifier of the controller/station node to which you are deploying the built Participant project.	Identifier of the controller/station node to which you have deployed the built Participant project last.
Last Built On	Date and time when you successfully built the Control Participant project to be deployed.	Date and time when you successfully built the Control Participant project that you have deployed last.
Last Deployed On	Not applicable	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.
Last Online Change Performed On	Not applicable	Date and time when you last changed the deployed Participant project by using the Refine Online command; otherwise the field is empty.
Last Sync. Performed On⁽¹⁾	Date and time when you last synchronized the standby controller with the primary one, using the Synchronize (Primary→Standby) command.	

(1) Displayed only when you select the command for a redundant controller.

(2) When you deploy changes to a redundant controller, the dialog box shows separate columns for the primary and the standby engine.

Deploying Changes Other Than FBD Language

The table describes the effect of the **Deploy changes / Undo Online Changes** command on items other than FBD language.

Item	Effect
Non-FBD sections	Details of changes inside non-FBD sections that the software has identified are not shown. Only the name of sections that have been added, removed, or inside which the software has identified changes is displayed. Nevertheless, the changes are applied in the logical Control Participant project.
Animation tables	Details of changes to such items that the software has identified are not shown. Only the name of the animation table and/or operator screen is displayed.
Operator screens	<p>The software displays any animation tables and/or operator screens that the built Participant project contains, including modified and/or new ones. Animation tables and/or operator screens that you have deleted from the Participant project are not shown.</p> <p>You cannot select individual items but only the category for deployment.</p> <p>When selected, animation tables and operator screens of the deployed Participant project are overwritten with the ones contained in the built Participant Project.</p> <p>By default, both categories are not selected.</p> <p>NOTE:</p> <ul style="list-style-type: none"> Overwriting of an existing operator screen creates a copy of the screen in the deployed Participant project. Following deployment, use the Refine Online command to edit the categories and make their contents consistent with the logical Control Participant project. Animation tables and/or operator screens that you have removed from the logical Participant project are not removed from the deployed Participant project by using the command. Use the Refine Online command to delete them.
Hardware	Details of CCOTF- compatible configuration changes are shown. The Hardware category is selected by default (if changes are detected) and is read-only.

Considerations When Deploying Hardware Mapping Changes

Consider the information that is provided on the following pages when using the **Deploy changes / Undo Online Changes** to deploy changes in the hardware mapping affecting:

- IP addresses, page 793
- Signals, page 608

Changes That Cannot Be Deployed

The software only identifies changes that pertain to a category displayed in the **Name** column of the **Deploy Changes / Undo Online Changes** dialog box. Changes that are not identified are not deployed. For more information, refer to Changes that Cannot be Deployed, page 686.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Perform **Deploy Built Project** operation to apply changes that are not identified in the **Deploy Changes / Undo Online Changes** dialog box.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The table indicates the types of changes that are not identified by the software and the command that you need to use to apply them to the deployed Control Participant project directly.

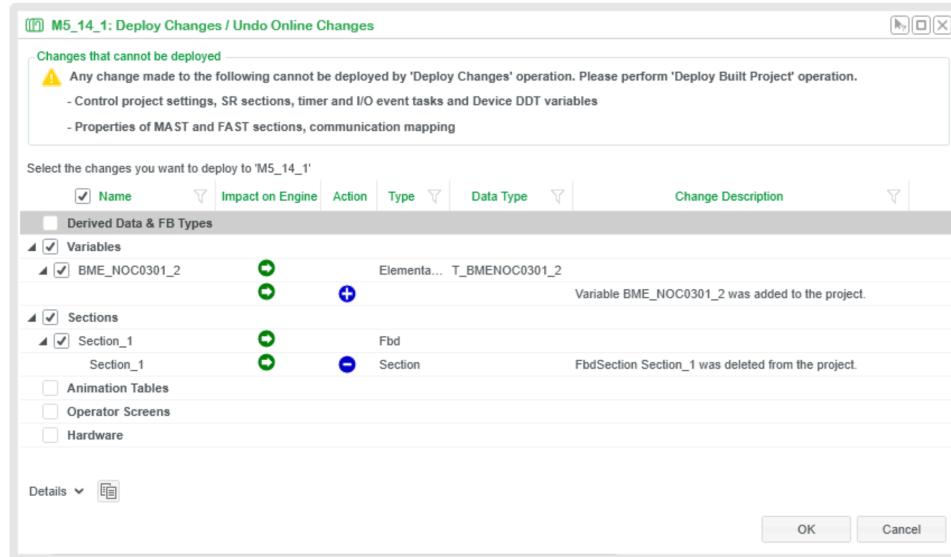
Type of change	Command to use
Control project settings (refer to the Control Participant help, <i>Operating Modes</i>)	Refine Online , page 727 or Deploy Built Project , page 667
SR sections	
Properties of MAST and FAST tasks	
Timer and I/O event tasks	
IODDT variables	Refine Online , page 727
Device DDT variables	Refer to the topic describing the deployment of changes related to device DDT variables, page 687
Changes to the topology, page 499, which you make in the Topology Explorer and that are not CCOTF-compatible.	Deploy Built Project , page 667
Changes to the communication mapping, page 582.	

Identifying Project and Hardware Changes

Before proceeding with the deployment, the software displays the **Deploy Changes / Undo Online Changes** dialog box, which provides information about the differences that it has identified between the built Participant project and the deployed Participant project. It also allows you to select, which changes you want to deploy.

NOTE: When you select a section in order to deploy changes to this section in the deployed Control Participant project, the software overrides the online changes that you have made in this section after the deployment.

The following figure shows an example of the **Deploy Changes / Undo Online Changes** dialog box that the software displays.



Header	Description
Name	Identifier of the items that are changed. Items that are selected are those that the software has identified as changed and that it will deploy. Items are grouped by categories of the Control Participant project. NOTE: A black square in a selection box indicates that at least 1 subitem is unselected.
Impact on Engine	Indicates if the deployment of the item requires stopping the engine, page 692. <ul style="list-style-type: none"> • Green dot with an arrow: You can deploy the item online. • Red dot with a square: The software needs to stop the engine to deploy the item.
Action	<ul style="list-style-type: none"> • Minus icon: The item will be removed from the deployed Participant project. (For example, because the item is removed from the built Participant project or it had been added to the deployed Participant project but not updated in the logical Participant project.) • Plus icon: The item will be added to the deployed Participant project. (for example, because the item is added to the built Participant project or it has been removed from the deployed Participant project during refine online and the change was not updated in the logical Participant project.) • Pencil icon: The item will be modified in the deployed Participant project. NOTE: When you select a section, the software deploys any changes that it has identified for this section. You cannot select/unselect changes individually inside a section.
Type	Indicates the type of the item identified as a change.
Data Type	Indicates the data type that is affected by the change, if applicable.
Description of Change	For FBD sections, variables, types, and hardware, indicates whether the change that the software has identified is an addition, a modification, or a deletion compared to the deployed Participant project.

NOTE: If the software does not detect a change that can be deployed, it displays a notification to inform you.

Deploying Changes Related to Device DDT Variable Creation, Update, or Deletion

The following table describes which information the software displays if a device DDT variable, page 445 is present only in either the built or the deployed Control project.

DDDT variable in the deployed Control project	DDDT variable in the built Control project	Information shown in the Deploy Changes / Undo Online Changes dialog box
Mapped or unmapped	Not present	No information. Use the Refine Online , page 727 or Deploy Built Project , page 667 command to remove the variable in the deployed Control project
Not present	Mapped or unmapped	The software detects the following as candidate for deployment: <ul style="list-style-type: none"> • The device DDT variable • Logic related to the variable • Sections containing the logic NOTE: The variable is unmapped in the deployed Control project.

The following table describes which information the software displays in a situation where:

- A device DDT variable with the same name exists in both the built and the deployed Control project but with a different data type or type version.
- The built Control project contains updated code using the device DDT variable.

DDDT variable in the deployed Control project	DDDT variable in the built Control project	Information shown in the Deploy Changes / Undo Online Changes dialog box
Mapped	Mapped/unmapped	The device DDT variable is not shown as candidate for deployment. Modifications to the DDT variable type are not deployed. Only the changes in code are shown as candidate for deployment. NOTE: <ul style="list-style-type: none"> • A notification is displayed in the Details section of the dialog box. • Deployment of changes may not succeed because not all changes are deployed.
Unmapped	Unmapped	The software detects the following as candidate for deployment: <ul style="list-style-type: none"> • The device DDT variable • Logic related to the variable • Sections containing the logic

The following table describes which information the software displays in a situation where:

- A device DDT variable with the same name exists in both the built and the deployed Control project but with a different data type or type version.
- The device DDT variable is not used in the code of either Control project.

DDDT variable in the deployed Control project	DDDT variable in the built Control project	Information shown in the Deploy Changes / Undo Online Changes dialog box
Mapped	Mapped/unmapped	The device DDT variable is not shown as candidate for deployment. modifications to the DDT variable type are not deployed. NOTE: A notification is displayed in the Details section of the dialog box.
Unmapped	Unmapped	The software detects the device DDT variable as candidate for deployment.

The following table describes which information the software displays in a situation where:

- A device DDT variable that exists in the deployed Control project has been deleted from the built Control project.
- The device DDT variable is either:
 - Used in the code.
 - Not used in the code.

DDDT variable in the deployed Control project	DDDT variable in the built Control project	Information shown in the Deploy Changes / Undo Online Changes dialog box
Mapped	Variable is deleted and references to it in code are removed.	The device DDT variable is not shown as candidate for deployment (removal). Only the changes in code are shown as candidate for deployment.
	Variable is deleted. It is not used in code.	no information
Unmapped	Variable is deleted and references to it in code are removed.	The software detects the following as candidate for deployment: <ul style="list-style-type: none"> • The device DDT variable (removal) • Logic related to the variable • Sections containing the logic
	Variable is deleted. It is not used in code.	The software detects the device DDT variable as candidate for deployment (removal).

Deploying Changes

⚠ WARNING	
UNINTENDED EQUIPMENT OPERATION	
<p>Before deploying to a controller, verify that:</p> <ul style="list-style-type: none"> • You have assigned a valid IP address to the controller. • The IP address corresponds to the physical address printed on the hardware. <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>	

NOTE: For more information, refer to Identifying Engines, page 651.

⚠ WARNING	
UNINTENDED EQUIPMENT OPERATION	
<p>Before deploying to an engine, verify that you have selected the correct project files.</p> <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>	

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING	
UNKNOWN OPERATIONAL STATE OF EQUIPMENT	
<p>Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.</p> <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>	

When you deploy changes to a redundant controller, although deployment to the primary controller succeeded, it is possible that synchronization of the standby controller, page 744 does not complete. In such case, the software displays a message in the notification panel to inform you. If an interruption of the connection to the system server or an unexpected stop of the system server is the cause, the notification panel also contains additional information about steps that you must complete when the connection to the system server is re-established.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that deployment to the standby controller completed successfully.
- Do not operate the primary controller if deployment to the standby controller does not complete successfully.
- Synchronize the standby controller manually, using the **Sync (Primary→Standby)** command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For information on the deployment of changes to redundant controllers, refer to the topic describing deployment to redundant controllers, page 666.

NOTE: For information on the **Sync (Primary→Standby)** command, refer to the Execution stage, page 707.

NOTE: Before deploying a CCOTF-compatible modification, refer to the topics describing the CCOTF functionality in the help of the Control Participant for your specific controller platform and follow the recommendations that they provide.

To deploy changes, proceed as follows.

Step	Action	Result
1	Verify that the engine can communicate with the PC, page 657. NOTE: To deploy changes to a controller simulator, verify that it is running, page 654 and using the same port number as the one shown in the IP Address field.	–
2	In the Topology Explorer , right-click the engine to which you want to deploy changes and select Deploy Changes / Undo Online Changes .	The Confirm Deploy Changes / Undo Online Changes dialog box opens.
3	Verify the information that is displayed.	–
4	Click OK . NOTE: Click Cancel to close the Confirm Deploy Changes / Undo Online Changes dialog box without deploying.	The Deploy Changes / Undo Online Changes dialog box opens.
5	Select the changes that you want to deploy.	–
6	Click OK . NOTE: Click Cancel to close the Deploy Changes / Undo Online Changes dialog box without deploying changes. This reverts to the previous Deploy Changes / Undo Online Changes dialog box.	The Deploy Changes / Undo Online Changes confirmation dialog box opens.

Step	Action	Result
7	If the deployment requires stopping the engine, in the Deploy Changes / Undo Online Changes confirmation dialog box, select Start engine after deploying built project if you want the software to start the engine after the deployment of changes is complete.	-
8	Click OK or Deploy depending on the use case. (Only either button is displayed depending on the deployment use case, without or with engine stop.) NOTE: Click Cancel to close the Deploy Changes / Undo Online Changes dialog box without deploying changes and revert to the selection of changes to deploy.	The software: <ul style="list-style-type: none"> • Stops the engine (if applicable). • Deploys the changes that you have selected to the deployed Control Participant project. • Displays Completed in the notification panel once it has successfully completed the deployment of changes to the engine, which is identified by its IP address. <ul style="list-style-type: none"> ◦ For a redundant controller, the notification panel displays separate messages for the primary and the standby engines by using their respective IP address to identify them. If deployment to either engine does not complete successfully, the software displays a corresponding message in the notification panel. • Updates the related date/time field. • Starts the controller (cold start) or software emulating a simulator if you selected the option (if applicable).

NOTE: In case the deployment of changes does not complete successfully, the software displays a notification to inform you. If the software stopped the engine before deploying Participant project changes, the engine remains in **Stopped** state.

Changes Requiring to Stop the Controller

Summary of Changes

The table indicates if changes that you deploy by using the **Deploy Changes / Undo Online Changes** command, or apply by using the [Refine Online](#), page 727 command require stopping the controller.

Type of change	Requires stopping the controller	
	When deploying, using Deploy Changes command	When applying, using the Refine Online command
General		
Modifying Control Expert project settings.	N/A	Refer to <i>Project Settings</i> for a list of settings that you can change and to <i>Synthesis of Authorized and Unauthorized Modifications</i> in the help of the Control Participant.
Program: Task Sections, SR, SFC Action & Transition		
Modifying/replacing partially the code of an SFC section (SFC chart, transition, action).	Yes	Refer to <i>Synthesis of Authorized and Unauthorized Modifications</i> in the help of the Control Participant.
Deleting SFC chart, transition, action.	Yes	
Adding an SFC section.	No	
Modifying or deleting an LL984 section.	Yes	
Adding an LL984 section.	No	
Modifying, deleting, or adding a section in either of the following languages: <ul style="list-style-type: none">• IL• LD• ST	No	
Used Variables (used even in Animation Tables or Operator Screens)		
Modifying the <i>Retain</i> or <i>Exchange On STBY</i> attribute of a used variable.	Yes	Yes
Modifying the time stamp attribute of any used variable.	Yes	Refer also to <i>Synthesis of Authorized and Unauthorized Modifications</i> in the help of the Control Participant.
Modifying the structure of a used DDT.	Yes	
Used DFBs		
Modifying the name of an input, output, or input/output pin (formal parameter) of a used DFB.	Yes	Refer also to <i>Synthesis of Authorized and Unauthorized Modifications</i> in the help of the Control Participant.
Adding an input, output, or input/output pin (formal parameter) to a used DFB.	Yes	
Deleting an input, output, or input/output pin (formal parameter) from a used DFB.	Yes	

Supervision Project Deployment Stage

Overview

This section describes the methods that you can use to deploy built Supervision Participant projects to computers (for example, operation servers, operator stations), which are connected to the Ethernet network. These computers are modeled by the corresponding station nodes in the topology of the system.

It also describes how to deploy only the changes that you have made following a first deployment.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Preparing for Deployment

Overview

The software lets you deploy a built Supervision Participant project and related files to the station nodes, page 516 that you have defined in the topology of the system and mapped, page 622 to the executable that is associated to the Supervision project.

You can deploy either way:

- In the **Project Explorer**: From the executable of the Supervision project. This deploys a selection of files to the station nodes that are mapped to the executable in the service mapping.
- In the **Topology Explorer**: From a station node that is mapped to the executable in the service mapping. This deploys a selection of files to this station node only.

The software detects roles of station nodes depending on the services, page 520 that are configured in the **Topology Explorer** and considers a station node to be:

- An I/O server when configured with Supervision and **OFS** services.
- An alarm, trend, or report server, or a Supervision client when configured with the Supervision service only.

The software deploys to the computer that has an IP address matching the IP address of the station node configured in the **Topology Explorer**.

For information on the required Supervision software, refer to the topic describing software requirements (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Target Computer Setup

To prepare for the deployment, you need to:

- Verify that the configuration of the station nodes, page 500 in the topology of the system matches the configuration of the target computers (for example, verify that the station node containing the **OFS** service has the same IP address as the computer that runs the OPC Factory Server or OPC UA Server Expert).

If a station node is configured with several NICs, each having an IP address:

- When deploying from the executable, the first IP address that is reachable from the system server computer is used.
- When deploying from the station node, you can select which IP address to use among those that are reachable from the system server computer.
- Create shared folders on each computer to which you deploy the Supervision project.
- Verify that the system server can access these shared folders; otherwise, the deployment cannot complete successfully.

For information on how to set up target computers, refer to the topic describing how to prepare for deployment (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

NOTE: When sharing a folder, restrict access by giving permissions only to the user who needs to access the folder. In this case, the user who is logged on to the system server computer.

Deploying the Built Supervision Participant Project from the Executable

Overview

By using the **Deploy Built Project** command, you can transfer selected files of the built Supervision Participant project to the operation server, and/or operator stations that are connected to the Ethernet network. The deployment is performed to the station nodes that you have mapped to the executable of the Supervision project in the service mapping and which represent these computers.

After a first deployment, you need to restore the deployed Supervision Participant project on each target computer.

Information related to subsequent deployments from the same executable is provided in the topic describing how to deploy the built Supervision Participant project again, page 696.

Deployed Files

You can deploy the following files:

- The Supervision master project file (.ctz), for example, *Supervision_1_Executable_1.ctz*, which includes the Supervision computer setup file (*citect.ini*) (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).
- Attached user files, page 497. These are located inside the master project (.ctz) on the target computer.
- OFSOPC/OPC UA configuration files (.xml), page 643: Deployed only to station nodes mapped to the *IOServer* service of the Supervision project. If there are several configuration files, each file is deployed to its respective station node only.
- The included projects, page 496.

You can select not to deploy the configuration file and/or the included project in the **Deployment File Selection** dialog box, page 700, which is accessible during the deployment process.

NOTE: Supervision project files that are deployed are already compiled by using the Supervision Participant. To verify the version, open a Supervision Participant window (for example, by refining a Supervision project), and open the **About** dialog box.

Deploying Out Of Date Participant Projects

You can deploy a Supervision Participant project whose status is **Out Of Date**.

Before deploying a built Supervision Participant project, verify its **Build State**, page 645. Only Supervision Participant projects whose status is **Built** contain the latest changes that you have made, page 645 to the system.

Aborting Deployment Tasks

After confirming the command by clicking **OK** in the **Deploy Built Project** or **Deployment File Selection** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task.

Files that have already been transferred to station nodes at the moment you abort the task are not removed from the target locations. Open the shared folders on each station node to confirm which files have been transferred.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are deploying only a small project, the task may complete before you are able to click the icon.

Deploying the Supervision Files

To deploy selected files pertaining to the built Supervision Participant project to the target computers from the executable, proceed as follows.

Step	Action
1	Verify that the target computers are prepared, page 694.
2	Right-click the executable that you want to deploy and select Deploy Built Project . Result: The Deploy Built Project dialog box opens, which gives you the possibility to modify the selection of deployed files.
3	Click: <ul style="list-style-type: none"> • Yes to open the Deployment File Selection dialog box, page 700; proceed to step 4. • No to deploy only the master Supervision project file (.ctz) but no OFSOPC/OPC UA configuration file and no included projects. • Cancel to cancel the deployment. Result: If you clicked No , the software: <ul style="list-style-type: none"> • Deploys the files to the appropriate folders on each target computer. • Displays deployment information in the notification panel for each target computer, including the IP addresses that were used. • Opens the Deploy Built Project dialog box, which contains a summary of the deployment process per target computer.
4	In the Deployment File Selection dialog box, select the files that you want to deploy and click OK . Result: The software: <ul style="list-style-type: none"> • Deploys the files to the appropriate folders on each target computer. • Displays deployment information in the notification panel for each target computer, including the IP addresses that were used. • Opens the Deploy Built Project dialog box, which contains a summary of the deployment process per target computer. NOTE: Click Cancel to cancel the deployment.

Restoring the Supervision Project

After a first deployment, you need to restore the deployed Supervision Participant project on each target computer.

Restoring the project creates a Supervision project folder on the target computer that has the same name as the deployed Supervision master project file (.ctz).

For information on restoring the deployed Supervision project, refer to topic describing how to restore Supervision projects (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

Deploying the Built Supervision Participant Project Again

To perform a subsequent deployment of the complete Supervision project from the same executable after you have modified and built it, follow the procedure describing how to deploy the project files, page 696.

The software replaces the existing files in the *SoCoDeploy*, *SoCoOfsDeploy*, and/or *SoCoOpcUaDeploy* shared folders (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide) on the target computers with the version that is contained in the Supervision project that you are deploying again.

Then, to apply the changes, the steps that you need to complete on each of the target computer vary, depending on the changes that you have deployed (for example, updating the OFSOPC/OPC UA configuration data manually). Refer to the topic describing how to execute the deployed Supervision project (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

NOTE: If you had already deployed attached user files, page 497 and you redeploy the same Supervision project after removing one or more of user files, these files remain in the *SoCoDeploy* shared folder where they had been deployed initially. You need to remove them manually from the shared folder.

Deploying the Built Supervision Participant Project from the Station Node

Overview

The **Deploy Built Project** command that is available from the station node provides an alternative deployment method. By selecting it, you can transfer selected files of the built Supervision Participant project to a specific computer connected to the Ethernet network (for example, the operation server or an operator station). The deployment is performed from the station node that represents this computer. The station node needs to be mapped (service mapping, page 622) to the executable of the Supervision project that you are deploying.

After a first deployment, you need to restore the deployed Supervision Participant project on the target computer.

Information related to subsequent deployments of the same Supervision project is provided in the topic describing how to deploy the built Supervision Participant project again, page 696.

Deployed Files

You can deploy the following files:

- The Supervision master project (.ctz file, for example, *Supervision_1_Executable_1.ctz*), which includes the Supervision cocomputer setup file (*citect.ini*) (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).
- Attached user files, page 497. These are located inside the master project (.ctz) on the target computer.
- The OFSOPC/OPC UA configuration files (.xml), page 643 that are related to the station node: Deployed only to station nodes mapped to the *IOServer* service of the Supervision project.
- The included projects, page 496.

You can select not to deploy the configuration files and/or the included project in the **Deployment File Selection** dialog box, page 700, which is accessible during the deployment process.

NOTE: Supervision project files that are deployed are already compiled by using the Supervision Participant. To verify the version, open a Supervision Participant window (for example, by refining a Supervision project), and open the **About** dialog box.

Selecting Participant Project and Control Executable

When you select the **Deploy Built Project** command for a station node, the software displays the **Deploy Built Project** dialog box, which allows you to select:

- **Project:** Any Supervision project that exists in the system whose executable is mapped to at least one service, page 622
- **Executable:** For each project, any built or out of date Supervision executable that is mapped to the station node (service mapping).

Built Participant Project Build Status

The software informs you when the built Supervision Participant project that you want to deploy is marked as being out of date. Nevertheless, you can proceed with the deployment.

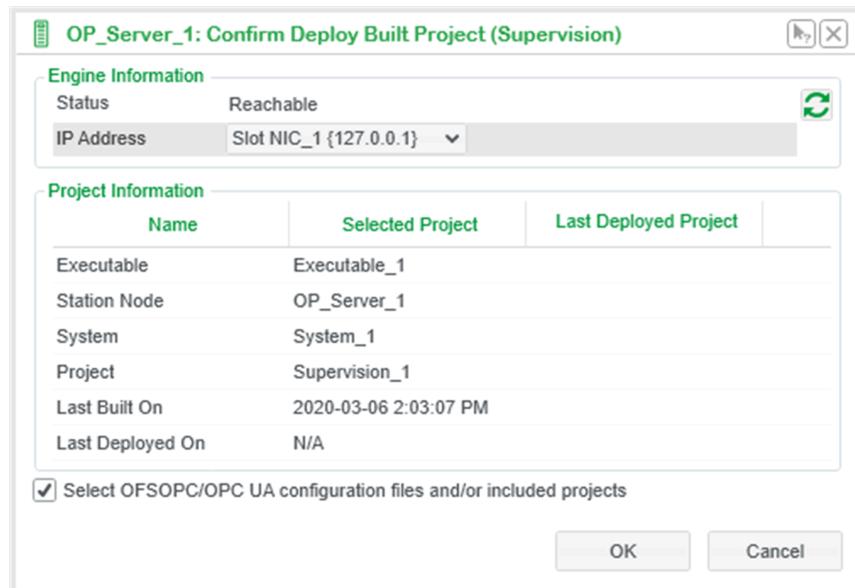
Before deploying a built Supervision Participant project, verify that its **Build State**, page 645 is up to date and/or that it contains the latest changes that you have made, page 645 to the system.

Supervision Participant Project Information

Before proceeding with the deployment of a built Participant project, the software displays information in the **Confirm Deploy Built Project** dialog box about:

- The station node that you are deploying to.
- The built Participant project to be deployed.
- The Participant project that was deployed last to the engine.

The following figure shows an example of the **Confirm Deploy Built Project** dialog box.



The tables describe the information that is displayed in the **Confirm Deploy Built Project** dialog box.

Engine Information section	Description
Status	Status of the station node that is selected for deployment. If the status is Not Reachable , you cannot proceed with the deployment. In such case, verify the IP address and the connection to the station node. NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.
IP Address	Identifier of the NIC through which the software will deploy the built Participant project and its IP address that is configured in the Topology Explorer . If the station node has several NICs, you can select which one you want to use for deployment.

Project Information section	Selected Project column	Last Deployed Project column
Executable	Identifier of the Supervision executable that you have selected and that contains the built Participant project to be deployed.	Identifier of the Supervision executable that contains the built Participant project that you have deployed last.
Station Node	Identifier of the topological entity to which you are deploying the built Participant project.	Identifier of the topological entity to which you have deployed the built Participant project last.
System	Identifier of the system that both the station node and the Supervision project, which you have selected belong to.	Identifier of the system that both the engine and the Supervision project, from which you have deployed the Participant project last belong to.
Project	Identifier of the Supervision Participant project that you have selected.	Identifier of the Supervision Participant project associated to the executable that you have deployed last.
Last Built On	Date and time when you successfully built the Supervision Participant project to be deployed.	Date and time when you successfully built the Supervision Participant project that you have deployed last.

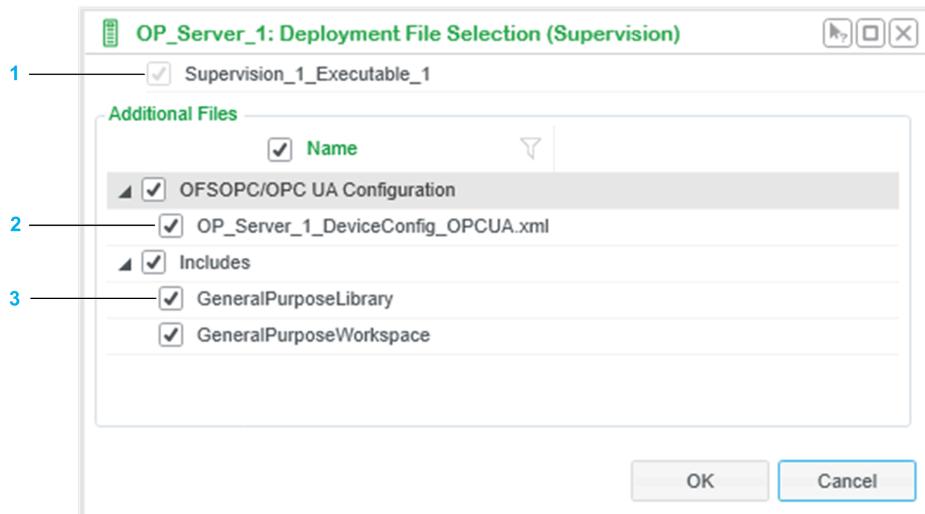
Project Information section	Selected Project column	Last Deployed Project column
Last Deployed On	Not applicable	Date and time when you last deployed the Participant project from the station node or from the executable associated to the Supervision project.
Select OFSOPC/OPC UA configuration files and/or included projects	<p>Check box.</p> <p>When selected, opens the Deployment File Selection dialog box, where you can select the OFSOPC/OPC UA configuration files and included projects to be deployed in addition to the Supervision master project file (.ctz).</p> <p>When cleared, the dialog box does not open and only the Supervision master project file (.ctz) is deployed.</p>	

Check box	Description
Select OFSOPC/OPC UA configuration files and/or included projects	<p>When selected, opens the Deployment File Selection dialog box, where you can select the OFSOPC/OPC UA configuration files and included projects to be deployed in addition to the Supervision master project file (.ctz).</p> <p>When cleared, the dialog box does not open and only the Supervision master project file (.ctz) is deployed.</p>

NOTE: When you deploy for the first time to a station node, the fields of the **Last Deployed Project** section are empty.

Selecting the Files to Deploy

The following figure shows an example of the **Deployment File Selection** dialog box, which opens when you select the **Select OFSOPC/OPC UA configuration files and/or included projects** check box and click **OK** in the **Confirm Deploy Built Project** dialog box.



Item	Description
1	Supervision master project file (.ctz), which is selected by default. You cannot clear the check box.
2	OFSOPC or OPC UA configuration files, page 643 that are related to the station node and generated depending on the selected protocol for the tag container. Configuration files can be selected only for station nodes that are mapped to an I/O server.
3	Included projects, page 496 that are associated to the master project.

Aborting Deployment Tasks

After confirming the command by clicking **OK** in the **Reconfirm Deploy Built Project** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task.

Files that have already been transferred to the station node at the moment you abort the task are not removed from the target locations. Open the shared folders on the station node to confirm which files have been transferred.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are deploying only a small project, the task may complete before you are able to click the icon.

Deploying the Supervision Files

To deploy selected files pertaining to the built Supervision Participant project from a station node, proceed as follows.

Step	Action
1	In the Topology Explorer , identify the station node that represents the computer you want to deploy to.
2	Verify that the target computer is prepared, page 694.
3	Right-click the station node and click Supervision > Deploy Build Project . Result: The Deploy Built Project dialog box opens.
4	Click the Project list and select the Supervision project.
5	Click the Executable list and select the executable associated to the built Supervision project that you want to deploy.
6	Click OK . Result: The Confirm Deploy Built Project dialog box opens. NOTE: Click Cancel to close the dialog box without deploying.
7	Verify the status of the engine in the Status field.
8	Verify the NIC identification and IP address or select one in the IP Address field.
9	Verify the information that is displayed in the Project Information section of the Confirm Deploy Built Project dialog box.
10	To modify the selection of files to deploy, select the Select OFS configuration files and/or included projects check box, page 699 and click OK ; otherwise, to exclude configuration files and included projects from the deployed files, clear the check box, click OK , and proceed to step 12. Result: The Deployment File Selection dialog box opens.
11	In the Deployment File Selection dialog box, select the files you want to deploy to the station node and click OK . Result: The Reconfirm Deploy Built Project dialog box opens. NOTE: Click Cancel to cancel your file selection and revert to the Confirm Deploy Built Project dialog box.
12	Click OK in the Reconfirm Deploy Built Project dialog box. Result: The software: <ul style="list-style-type: none"> • Deploys the selected files to the appropriate folders on the target computer. • Displays Completed in the notification panel and provides deployment information. • Updates the related date/time field. NOTE: Click Cancel to revert to the Confirm Deploy Built Project dialog box.

Restoring the Supervision Project

After a first deployment, you need to restore the deployed Supervision Participant project on the target computer.

Restoring the project creates a Supervision project folder on the target computer that has the same name as the deployed Supervision master project file (.ctz).

For information on restoring the deployed Supervision project, refer to topic describing how to restore Supervision projects (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

Deploying the Built Supervision Participant Project Again

To perform a subsequent deployment of the complete Supervision project after you have modified and built it, follow the procedure describing how to deploy the project files, page 701.

The software replaces the existing files in the *SoCoDeploy* and *SoCoOfsDeploy* shared folders (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide) on the target computer with the version that is contained in the Supervision project that you are deploying again.

Then, to apply the changes, the steps that you need to complete on the target computer vary, depending on the changes that you have deployed (for example, updating the OFSOPC/OPC UA configuration data manually). Refer to the topic describing how to execute the deployed Supervision project (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

NOTE: If you had already deployed attached user files, page 497 and you redeploy the same Supervision project after removing one or more of user files, these files remain in the *SoCoDeploy* shared folder where they had been deployed initially. You need to remove them manually from the shared folder.

Deploying Supervision Participant Project Changes

Overview

By using the **Deploy Changes** command, you can deploy the changes that you have made to an already deployed Supervision Participant project given these changes are supported by the command.

You can deploy changes either from the executable or the station node independently from the method that you have used to perform the previous deployment.

You can select to deploy only the Supervision master project file (.ctz) or also additional files.

Prerequisites

The following conditions must be satisfied to be able to deploy changes by using the **Deploy Changes** command.

- You must have deployed the complete Supervision Participant project to the station node and restored it by using the Supervision software.
- The Supervision project folder (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide) that was created by restoring the deployed Supervision Participant project must be present on the target computer.
- The changes that can be deployed incrementally are only those that can be applied by using the **Build** command:
 - A maximum of 5 changes involving Supervision pages.
 - A maximum of 30 changes involving data facets.
 - Modifications of parameters of the **Message** category of Supervision elements

Refer to the topic describing the build stage for details, page 647.

NOTE: Using the **Build All** command disables the **Deploy Changes** command.

Deployed Files

The files that you can deploy are the same as for the deployment of the complete Supervision Participant project, page 698.

You can change the selection during the deployment process in the **Deployment File Selection** dialog box, page 700.

Aborting Deployment Tasks

After confirming the command by clicking **OK** in the **Reconfirm Deploy Changes** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task.

Files that have already been transferred to station nodes at the moment you abort the task are not removed from the target locations. Open the shared folders on each station node to confirm which files have been transferred.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are deploying only few changes, the task may complete before you are able to click the icon.

Deploying Changes from the Executable

To deploy changes to the Supervision Participant project to the target computers from the executable, proceed as follows.

Step	Action
1	Verify that the target computers are prepared, page 694.
2	In the Project Explorer , expand the Supervision project that you want to deploy from and open the Executables folder to show existing executables.
3	Right-click the executable that you want to deploy and select Deploy Changes . Result: The Deploy Changes dialog box opens, which gives you the possibility to modify the selection of files.
4	Click: <ul style="list-style-type: none">• Yes to open the Deployment File Selection dialog box, page 700; proceed to step 5.• No to deploy only the master Supervision project file (.ctz) but no OFSOPC/OPC UA configuration file and no included project.• Cancel to cancel the deployment. Result: If you clicked No , the software: <ul style="list-style-type: none">• Deploys the master Supervision project file to the appropriate folders on each target computer.• Displays deployment information in the notification panel for each target computer, including the IP addresses that were used.• Opens the Deploy Changes dialog box, which contains a summary of the deployment process per target computer.
5	In the Deployment File Selection dialog box, select the files that you want to deploy and click OK . Result: The software: <ul style="list-style-type: none">• Deploys the files to the appropriate folders on each target computer.• Displays deployment information in the notification panel for each target computer, including the IP addresses that were used.• Opens the Deploy Changes dialog box, which contains a summary of the deployment process per target computer. NOTE: Click Cancel to close the dialog box without deploying.

Deploying Changes from the Station Node

To deploy changes to the Supervision Participant project from a station node, proceed as follows.

Step	Action
1	In the Topology Explorer , identify the station node that represents the computer you want to deploy to.
2	Verify that the target computer is prepared, page 694.
3	Right-click the station node and click Supervision > Deploy Changes . Result: The Confirm Deploy Changes dialog box opens. The information of the last deployed executable and project is indicated in the Executable and Project fields of the Project Information section.
4	Verify the status of the engine in the Status field.
5	Verify the NIC identification and IP address or select one in the IP Address field.
6	If you want to deploy other files in addition to the master Supervision project, select the Select OFSOPC/OPC UA configuration files and/or included projects check box, page 699 and click OK ; otherwise, to exclude configuration files and included projects from the deployed files, clear the check box, click OK , and proceed to step 8. Result: The Deployment File Selection dialog box opens. NOTE: Click Cancel to close the dialog box without deploying.

Step	Action
7	<p>In the Deployment File Selection dialog box, select the files you want to deploy to the station node and click OK.</p> <p>Result: The Reconfirm Deploy Changes dialog box opens.</p> <p>NOTE: Click Cancel to cancel your file selection and revert to the Confirm Deploy Changes dialog box.</p>
8	<p>Click OK in the Reconfirm Deploy Changes dialog box.</p> <p>Result: The software:</p> <ul style="list-style-type: none">Deploys the selected files to the appropriate folders on the target computer.Displays Completed in the notification panel and provides deployment information.Updates the related date/time field. <p>NOTE: Click Cancel to revert to the Confirm Deploy Changes dialog box.</p>

Execution Stage

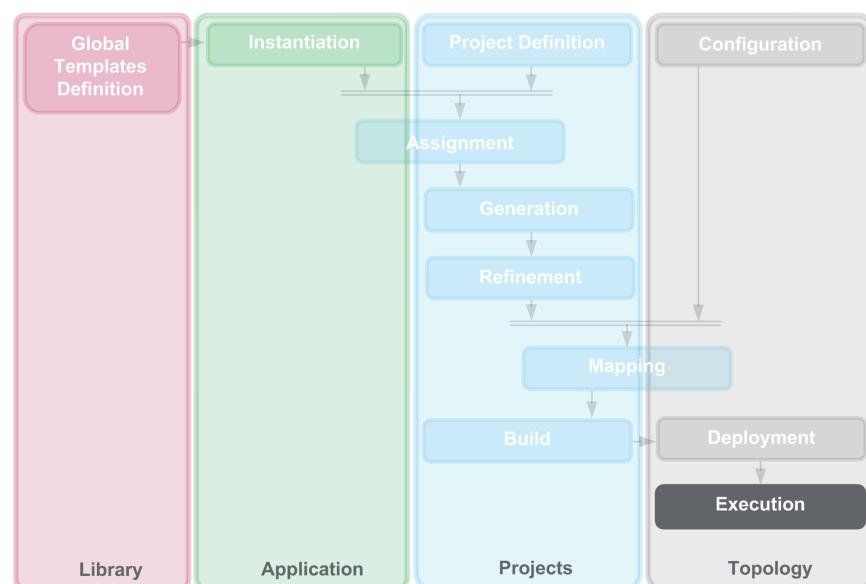
What's in This Chapter

Control Project Execution Stage	707
Supervision Project Execution Stage	757

Overview

This chapter describes how to execute and/or modify the Control and Supervision projects that you have deployed to engines of the system.

The following figure shows the position of the **Execution** stage within the system engineering life cycle.



Refer to the Execution stage, page 56 for a description of the purpose of this stage.

Control Project Execution Stage

Overview

This section describes how to use the **Topology Explorer** to change the state of controllers, modify deployed Control Participant projects, and how to transfer changes to the associated logical Participant project in the software.

NOTE: When one or more users work on components of the same system simultaneously (for example, Participant projects, the application, project containers, topological entities), locking mechanisms may restrict certain concurrent actions, page 70. In such case, check the **Notification Panel**, page 73 for details.

Control Participant Project Execution Process

Overview

The software allows you to execute a deployed Control Participant project, make changes to the deployed Participant project online, and update the logical Control Participant source project with these changes.

For this purpose, following the deployment of a built Control Participant project, the following commands become available:

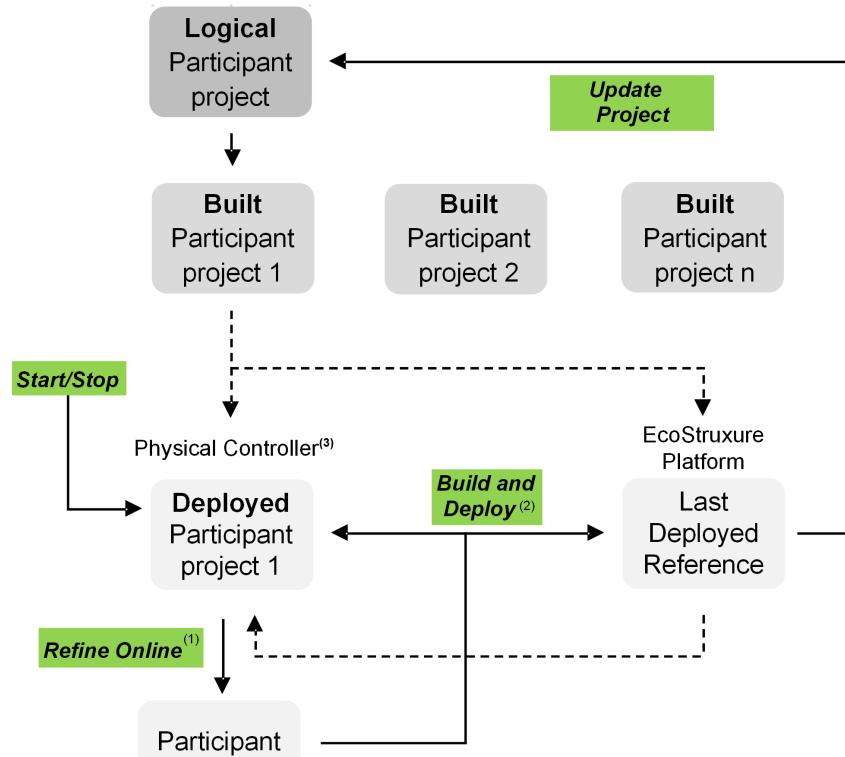
- **Start**
- **Stop**
- **Refine Online**
- **Sync (Primary→Standby)**
- **Update Project**

The commands are available for the following engines to which you have deployed a Control Participant project:

- Controllers.
- Station nodes running software emulating a simulator.

Execution Flowchart

The following figure illustrates the different steps that are part of the execution of the deployed Control Participant project and their associated commands.



--- Represents commands pertaining to the deployment stage, page 650.

(1) Includes the section import functionality, page 738.

(2) The functionality is a sub-command of the **Refine Online** command within the Control Participant.

(3) Or emulated simulator (Control service of a station node).

Prerequisites

The table describes the actions that are prerequisite for using a command.

Command	Actions
Start / Stop	Deploy the built Control Participant project.
Refine Online	Deploy the built Control Participant project.
Sync (Primary→Standby)	<ul style="list-style-type: none"> Deploy the built Control Participant project. Either of the following use cases may apply: <ul style="list-style-type: none"> Deployment to, or synchronization of the standby engine did not complete successfully following execution of the Deploy, Re-Deploy, Deploy Changes, or Refine Online command. Deployment was done to the primary engine only because the <i>ControllerExist</i> parameter is set to <i>Primary</i>. You need to set the parameter to <i>PrimaryAndStandby</i>, verify that the standby engine is reachable, and that both engines are connected by a Hot Standby cable link. Verify that the status of the primary engine is Running Primary.
Update Project	<ul style="list-style-type: none"> Deploy the built Control Participant project. Change the deployed Control Participant project by using the Refine Online command. <p>NOTE: Certain changes cannot be transferred, page 686 to the logical Control Participant project by using the command.</p>

For the other actions that are required, for example, mapping of the Control executable to an engine, refer to the system engineering life cycle that is described in this manual.

Control Project Mismatch Verification

Before executing the **Start/Stop**, **Sync (Primary→Standby)**, or **Refine Online** command, the software compares the Control Participant project that is deployed in the engine with the [last deployed reference](#), page 708, which is the Control Participant project residing in the software.

If they are identical, the command is executed. If they differ, the software starts a recovery process first. This process may take several minutes to complete.

In case the software displays a notification about a signature mismatch, the command cannot be executed. Contact Schneider Electric support.

Client/Server Connection

In case of an unexpected stop of the system server, the notification panel of engineering clients that were connected to the system server before the communication interruption may contain information about operations that did not complete successfully because of this interruption. It may also indicate corrective actions.

After an interruption of the client/server connection, the states of both the engineering client and the system server may have become inconsistent if an execution operation was in progress.

NOTICE

DATA CORRUPTION

After an interruption of the client/server connection, verify that the last operation executed by the software was completed successfully, and if necessary, repeat the last operation.

Failure to follow these instructions can result in equipment damage.

NOTE: For more information on client and server behavior in case of a communication interruption, refer to the topic describing client/server connection, page 64.

Start/Stop Commands

The software can send a start or stop command to the controller or software emulating a simulator.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING

UNKNOWN OPERATIONAL STATE OF EQUIPMENT

Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Execution With Redundant Controllers

Overview

This topic describes aspects that are specific to the execution of Control Participant projects with redundant controllers, which are modeled in the topology of the system.

The same commands that are used on standalone configurations are also used for redundant controllers.

Detailed information on the commands that are available as part of the execution stage are described later in this chapter.

For a description of the sequence of actions that Control performs on the engines, refer to the topic describing main redundant controller scenarios, page 741.

NOTE: For controllers of the M580 platform, you cannot perform execution tasks on the CPU module or a NOC communication module by using *IP address A* and *IP address B*. You can only select main IP address, page 683.

ControllerExist Property

Verify that the **ControllerExist** parameter is set correctly, page 665.

Engine Status for Redundant Controllers

Refer to the information that is provided in the topic describing deployment to redundant controllers, page 665.

Starting/Stopping a Redundant Controller

For redundant controllers, the **Start/Stop** commands act on the primary and the standby engines, provided that both are reachable.

When you select the **Start** or **Stop** command for a redundant controller, page 744, composed of one primary and one standby engine to which IP addresses *IP*, and *IP + 1* are assigned respectively, the software displays the **Confirm <Command>** dialog box (where *<Command>* represents either **Start** or **Stop**), which provides information on both engines.

NOTE: When you select the **Start** command and the status of the standby engine is **Stop Offline** (or **Stop** for controllers of the **M580** platform), the primary engine starts and synchronizes the standby engine so that both engines start.

Redundant Controller Stop Sequence

When you stop the redundant controller and both engines are running, the software stops in the following order:

- The standby engine
- The primary engine

Refining Online with a Redundant Controller

To proceed with the deployment, the primary engine must be in either **Running Primary** or **Stop Offline** (or **Stop** for controllers of the **M580** platform) state.

The software executes the command on the project that is deployed on the primary engine.

When you deploy online changes that do not require stopping the engine, and the primary engine is in **Running Primary** state:

- The software enables the configuration mismatch feature of the controller so that the standby engine does not enter the offline operating mode when changes are deployed to the primary engine.
- The software deploys to the primary engine first, and upon successful completion of the deployment, automatically synchronizes the standby engine.
- Then, the software disables the configuration mismatch feature.

When you deploy changes to a redundant controller, although deployment to the primary controller succeeded, it is possible that synchronization of the standby controller, page 744 does not complete. In such case, the software displays a message in the notification panel to inform you. If an interruption of the connection to the system server or an unexpected stop of the system server is the cause, the notification panel also contains additional information about steps that you must complete when the connection to the system server is re-established. This is the case, for example, when the software cannot disable the configuration mismatch feature. If this feature is enabled and a switchover occurs, the standby controller operates with a program that does not contain the changes that you have deployed to the primary controller.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that deployment to the standby controller completed successfully.
- Do not operate the primary controller if deployment to the standby controller does not complete successfully.
- Synchronize the standby controller manually, using the **Sync (Primary→Standby)** command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: If the primary engine is not in **Running Primary** state, synchronization of the standby engine cannot complete successfully. In such case, starting, page 713 the primary engine synchronizes the standby engine.

Redundant Controller Synchronization Timeout

After you select the **Build and Deploy** command (see EcoStruxure™ Process Expert, Control Participant Services, User Guide), upon successful deployment to the primary engine, the deployed Control Participant project is stored in the contents repository as the last deployed reference, page 708.

It avoids project mismatch, page 709 and allows you to synchronize the standby controller, using the **Sync (Primary→Standby)** command in case the synchronization timeout is elapsed, page 719.

Starting and Stopping Controllers

Overview

Following the deployment of a built Control Participant project to an engine, you can change the state of the engine by using the following commands:

- **Start**
- **Stop**

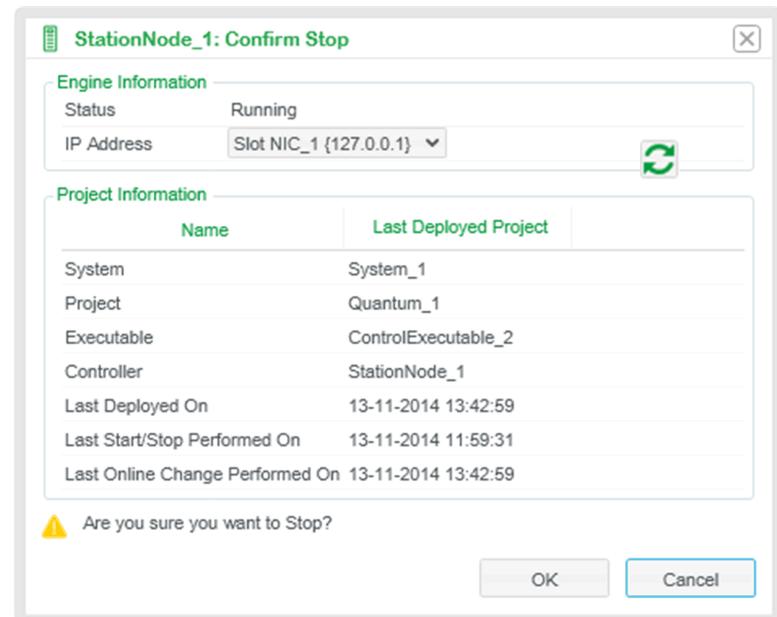
The commands act on the physical controller or on the software emulating a simulator that is installed on the computer represented by the engine in the topology of the system. The IP address of the controller or of the computer on the Ethernet network needs to match with the address that is assigned to the corresponding engine.

Confirming Execution

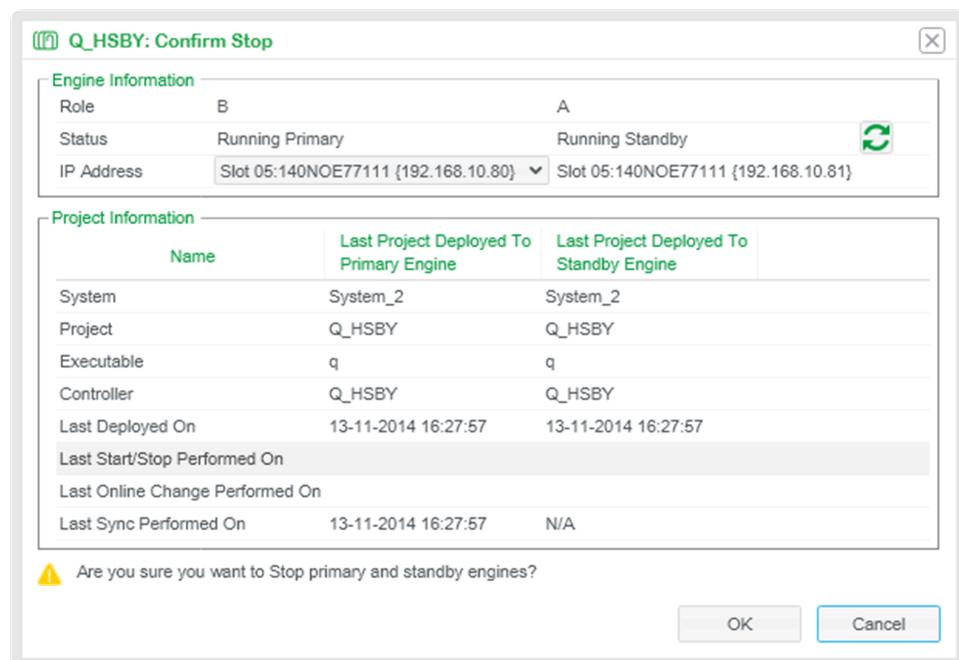
When you select the **Start** or **Stop** command for a controller or controller simulator, before proceeding with the execution of the command, the software displays the **Confirm <Command>** dialog box (where *<Command>* represents either **Start** or **Stop**), which contains information about:

- The engine that you want to start or stop.
- The Control Participant project that is deployed to the engine.

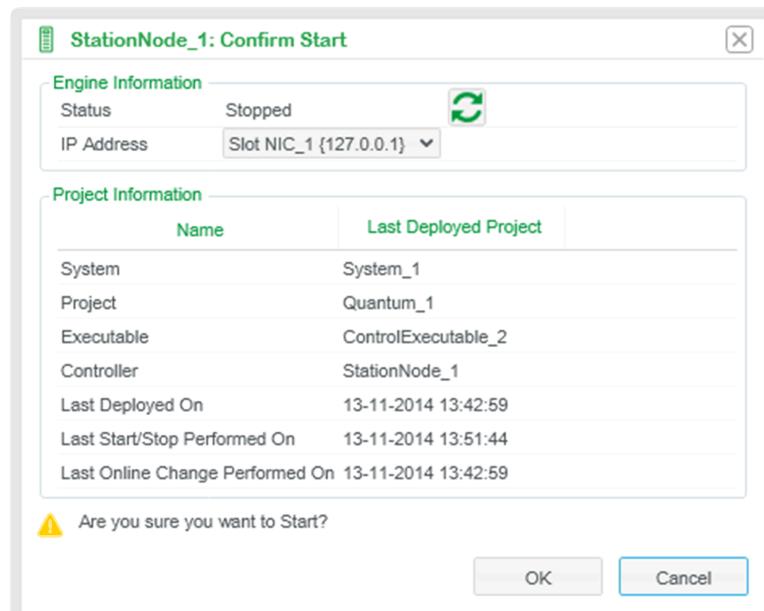
The following figure shows an example of the **Confirm Stop** dialog box that opens when you select the **Stop** command for a controller that is running.



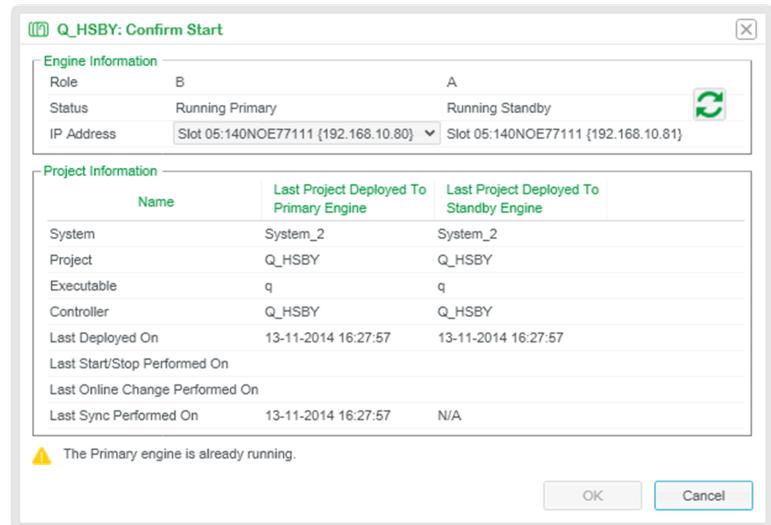
The following figure shows an example of the **Confirm Stop** dialog box that opens when you select the **Stop** command for a redundant controller that is running.



The following figure shows an example of the **Confirm Start** dialog box that opens when you select the **start** command for a controller that is stopped.



The following figure shows an example of the **Confirm Start** dialog box that opens when you select the **start** command for a redundant controller that is running. The command cannot be executed because the engine is already running.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm Start** and **Confirm Stop** dialog boxes.

Label	Description
Role⁽¹⁾	Role of the engine: <i>A</i> or <i>B</i> . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for execution. If the status is Not Reachable you cannot proceed with the command. In such case, verify the IP address, the connection to the engine, and the status of the engine.
IP Address	Identifier of the communication module through which the software sends the command and its IP address that is configured in the Topology Explorer . If the topological entity has several communication modules, you can select through which one you want to start or stop. NOTE: For controllers of the M580 platform, you cannot start or stop an engine by using <i>IP address A</i> of the CPU module or a NOC communication module. You can only select main IP address, page 683. NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.

⁽¹⁾ Displayed only when you select the command for a redundant controller.

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Start** and **Confirm Stop** dialog boxes.

Name column	Last Deployed Project column ⁽²⁾
System	Identifier of the system that both the engine and the Control project, from which you have deployed the Participant project last belong to.
Project	Identifier of the Control Participant project associated to the deployed Participant project that you have deployed last.
Executable	Identifier of the Control executable that contains the built Participant project that you have deployed last.
Controller	Identifier of the topological entity to which you have deployed the built Participant project last.

Name column	Last Deployed Project column ⁽²⁾
Last Deployed On	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.
Last Start/Stop Performed On	Date and time when you last started or stopped the engine.
Last Online Change Performed On	Date and time when you last changed the deployed Participant project by using the Refine Online command; otherwise the field is empty.
Last Sync. Performed On⁽¹⁾	Date and time when you last synchronized the standby engine with the primary engine by using the Synchronize (Primary→Standby) command; otherwise the field is empty.
(1) Displayed only when you select the command for a redundant controller.	
(2) When you start or stop a redundant controller, the dialog box shows separate columns for the primary and the standby engine.	

NOTE: If the software detects that the engine is already running or stopped, it displays information in the dialog box and you cannot execute the command.

Starting an Engine

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before starting or stopping a controller, verify that:

- You have assigned a valid IP address to the controller.
- The IP address corresponds to the physical address printed on the hardware.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For more information, refer to Identifying Engines, page 651.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING

UNKNOWN OPERATIONAL STATE OF EQUIPMENT

Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

When starting a redundant controller, synchronization of the standby controller may not complete successfully. In such case, the software displays a message in the notification panel to inform you.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that deployment to the standby controller completed successfully.
- Do not operate the primary controller if deployment to the standby controller does not complete successfully.
- Synchronize the standby controller manually, using the **Sync (Primary→Standby)** command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For information on the **Sync (Primary→Standby)** command, refer to the topic describing synchronization of redundant controllers, page 719.

To start the engine, proceed as follows.

Step	Action	Result
1	Verify that the engine is connected to the computer, page 657.	–
2	In the Topology Explorer , right-click the engine and select Start .	The Confirm Start dialog box opens.
3	Verify the information that is displayed.	–
4	Click OK . NOTE: Click Cancel to close the Confirm Start dialog box without starting the engine.	The Reconfirm Start dialog box opens.
5	Click OK . NOTE: Click Cancel to close the Reconfirm Start dialog box without starting the engine. This reverts to the Confirm Start dialog box.	<p>The software:</p> <ul style="list-style-type: none"> • Starts the engine. • Displays Completed in the notification panel once it has successfully completed the command for the engine, which is identified by its IP address. <ul style="list-style-type: none"> ◦ For a redundant controller, the information pane displays separate messages for the primary and the standby engines by using their respective IP address to identify them. • Updates the Last Start/Stop Performed On field.

Stopping an Engine

WARNING

UNINTENDED EQUIPMENT OPERATION

Before starting or stopping a controller, verify that:

- You have assigned a valid IP address to the controller.
- The IP address corresponds to the physical address printed on the hardware.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For more information, refer to Identifying Engines, page 651.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING

UNKNOWN OPERATIONAL STATE OF EQUIPMENT

Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To stop the engine, proceed as follows.

Step	Action	Result
1	Verify that the engine is connected to the computer, page 657.	–
2	In the Topology Explorer , right-click the engine that you want to stop and select Stop .	The Confirm Stop dialog box opens.
3	Verify the information that is displayed.	–
4	Click OK . NOTE: Click Cancel to close the Confirm Stop dialog box without stopping the engine.	The Reconfirm Stop dialog box opens.
5	Click OK . NOTE: Click Cancel to close the Reconfirm Stop dialog box without stopping the engine. This reverts to the Confirm Stop dialog box.	<p>The software:</p> <ul style="list-style-type: none"> • Stops the engine to stop the execution of the deployed Control participant project. • Displays Completed in the notification panel once it has successfully completed the command for the engine, which is identified by its IP address. <ul style="list-style-type: none"> ◦ For a redundant controller, the information pane displays separate messages for the primary and the standby engines by using their respective IP address to identify them. • Updates the Last Start/Stop Performed On field.

Synchronizing Redundant Controllers

Overview

The **Sync (Primary→Standby)** command allows you to manually transfer the deployed Control Participant project from the primary engine to the secondary redundant engine in case of an application program, configuration, or data mismatch.

Manual synchronization is only possible when the primary engine is running. The state of the standby engine does not matter.

The software verifies if the application program or configuration of the primary and the secondary redundant engines are identical and proceeds with the transfer only if it detects a difference.

It also verifies that there is no project mismatch, page 709.

The command is available only for entities of the topology that are configured as redundant controllers.

Refer also to Redundant Controller Deployment and Execution Scenarios, page 744.

Hot Standby Cable Link

If the Hot Standby cable link between the controllers is broken, synchronization does not complete successfully, and a message is shown to inform you of the detected error.

Synchronization Timeout

During synchronization of the standby engine, the software compares the signatures of both the primary and the standby engines. When they are identical, it considers that the synchronization completed successfully.

After 3 min, if the comparison does not detect identical signatures, the software displays a dialog box to inform you that synchronization did not complete successfully and cancels the comparison process. However, the synchronization itself is not canceled and may still complete after the timeout.

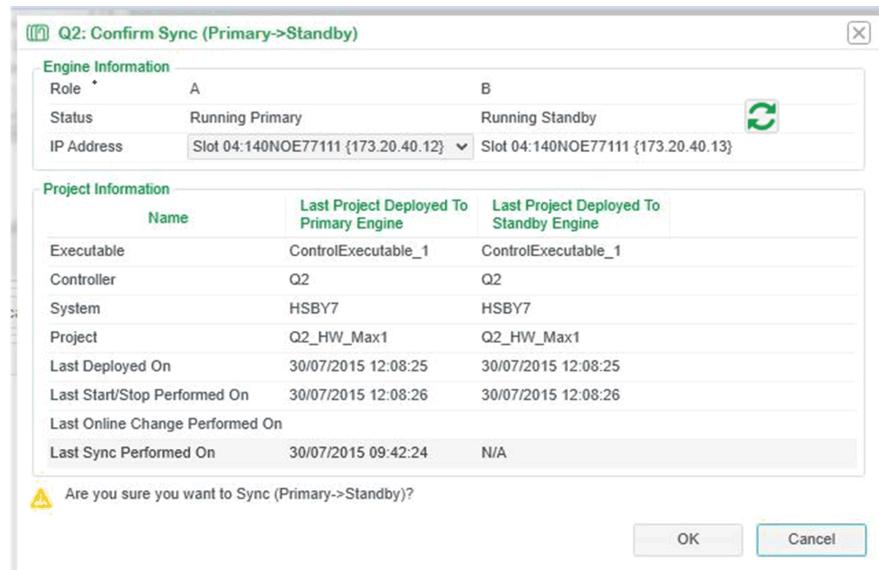
You can retry synchronizing by using the **Sync (Primary→Standby)** command.

Confirming the Synchronization

Before proceeding with the execution of the command, the software displays the **Confirm Sync (Primary→Standby)** dialog box, which contains information about:

- The current state of the primary and standby engines.
- The deployed Control Participant project to transfer to the standby controller.

The following figure shows an example of the **Confirm Sync (Primary→Standby)** dialog box that opens when you select the **Sync (Primary→Standby)** command with the primary controller in **Running Primary** state.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm Sync (Primary→Standby)** dialog box.

Label	Description
Role	Role of the engine: <i>A</i> or <i>B</i> . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for synchronization. If the status is Not Reachable you cannot proceed with the command. In such case, verify the IP address, the connection to the engine, and the status of the engine.
IP Address	Identifier of the communication module through which the software sends the command and its IP address that is configured in the Topology Explorer . If the topological entity has several communication modules, you can select through which one you want to synchronize. NOTE: For controllers of the M580 platform, you cannot synchronize an engine by using <i>IP address A</i> of the CPU module or a NOC communication module. You can only select main IP address, page 683.

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Sync (Primary→Standby)** dialog box.

Name column	Last Deployed Project To Primary Engine / Standby Engines columns
System	Identifier of the system that both the engine and the Control project, from which you have deployed the Participant project last belong to.
Project	Identifier of the Control Participant project associated to the deployed Participant project that you have deployed last.
Executable	Identifier of the Control executable that contains the built Participant project that you have deployed last.
Controller	Identifier of the topological entity to which you have deployed the built Participant project last.
Last Deployed On	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.

Name column	Last Deployed Project To Primary Engine / Standby Engines columns
Last Start/Stop Performed On	Date and time when you last started or stopped the engine.
Last Online Change Performed On	Date and time when you last changed the deployed Participant project by using the Refine Online command; otherwise the field is empty.
Last Sync. Performed On	Date and time when you last synchronized the standby engine with the primary engine by using the Synchronize (Primary→Standby) command; otherwise the field is empty.

NOTE: If the software detects that the engine is stopped, the **Confirm Sync (Primary→Standby)** dialog box opens to inform you that the synchronization cannot be completed.

Synchronizing the Standby Controller

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before starting or stopping a controller, verify that:

- You have assigned a valid IP address to the controller.
- The IP address corresponds to the physical address printed on the hardware.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For more information, refer to Identifying Engines, page 651.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING

UNKNOWN OPERATIONAL STATE OF EQUIPMENT

Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Synchronization of the standby controller, page 744 may not complete, for example, if the primary or the standby engine is reserved. In such case, a message appears in the notification panel to inform you.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that deployment to the standby controller completed successfully.
- Do not operate the primary controller if deployment to the standby controller does not complete successfully.
- Synchronize the standby controller manually, using the **Sync (Primary→Standby)** command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To synchronize the standby engine with the primary engine, proceed as follows.

Step	Action
1	Verify that the primary engine is connected to the PC, page 657 and its status is Running Primary .
2	In the Topology Explorer , right-click the redundant controller that you want to synchronize and select Sync (Primary→Standby) . Result: The Confirm Sync (Primary→Standby) dialog box opens.
3	Verify the information that is displayed.
4	Click OK . NOTE: Click Cancel to close the Confirm Sync (Primary→Standby) dialog box without synchronizing.
5	The Reconfirm Sync (Primary→Standby) dialog box opens.
6	Click OK . NOTE: Click Cancel to close the Reconfirm Sync (Primary→Standby) dialog box without synchronizing. This reverts to the Confirm Sync (Primary→Standby) dialog box .
7	Result: If the software detects no difference in the application program or configuration between the primary and the standby engines, it does not proceed with the transfer, and displays the Sync Operation Result dialog box to inform you of the result. Otherwise, the software: <ul style="list-style-type: none">• Transfers the deployed Control Participant project from the primary engine to the standby engine.• Displays Completed in the notification panel once it has successfully completed the command for the engines, which are identified by their respective IP address.• Updates the related date/time field.

About Changes Made Online

Overview

This topic provides an overview of the workflow of the system engineering life cycle with a focus on changes that you make during online refinement.

After refining the deployed project online, the logical and the deployed Control Participant projects are different.

If you refine (offline) the logical Control Participant project later on and deploy these changes, the operation may undo your online refinements. You have the possibility to preserve online refinements by transferring them to the logical Control Participant project. However, some online refinements cannot be transferred or can be transferred only partially.

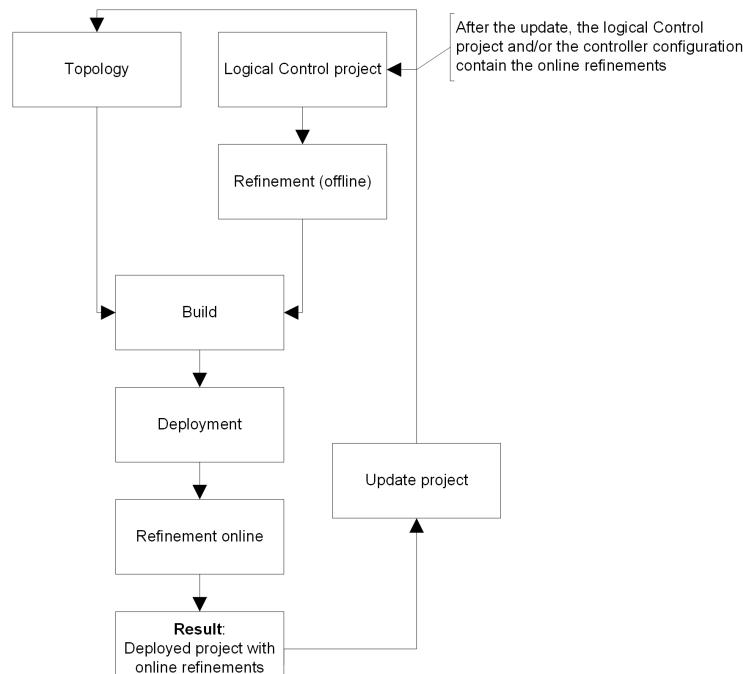
The table indicates which workflow chart provides the information depending on your objective.

Objective	Refer to
Making online refinements persistent in the logical Control Participant project and controller configuration by updating the project.	Workflow 1
Preserving online refinements that cannot be transferred to the logical Control Participant project	Workflow 2
Updating the logical Control Participant project with references to DDDT variables coming from the topology.	Workflow 3

NOTE: For details on using the various commands that allow you to refine online, page 727, deploy changes, page 682, and update the logical Control project, page 748, refer to the corresponding topics in the deployment and execution sections.

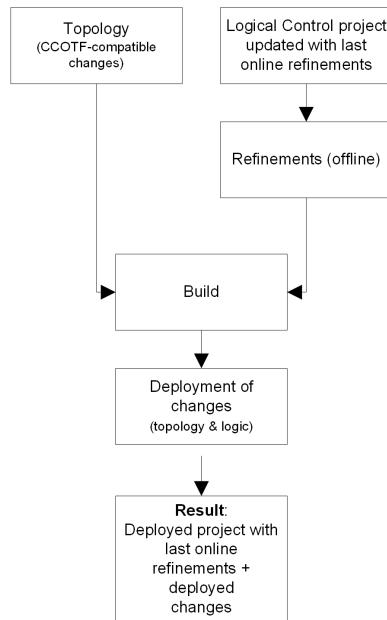
Workflow 1

The figure illustrates the typical workflow to make online refinements persistent by transferring them to the logical Control Participant project.



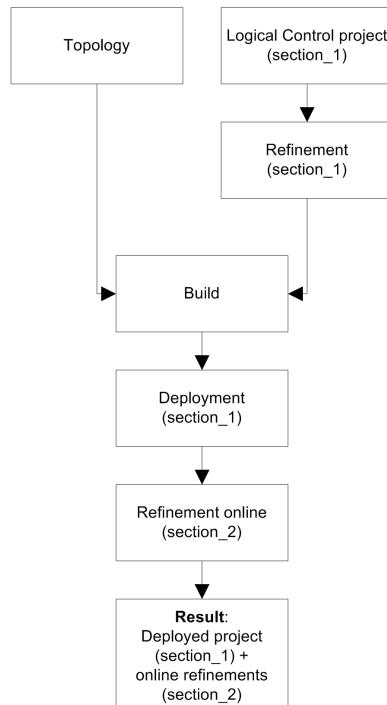
NOTE: Not updating the logical Control Participant project allows you to undo online refinements by deploying changes. The software detects the online refinements as change compared to the logical Control project. However, using the **Deploy Changes** command to undo online changes, page 683 is not possible if you have made changes to the configuration online.

This allows you later on to deploy additional changes to the sections that you had modified during refinement online while preserving online refinements. These changes can come from the topology, and/or the application, and/or from refinements of the logical Control Participant project.

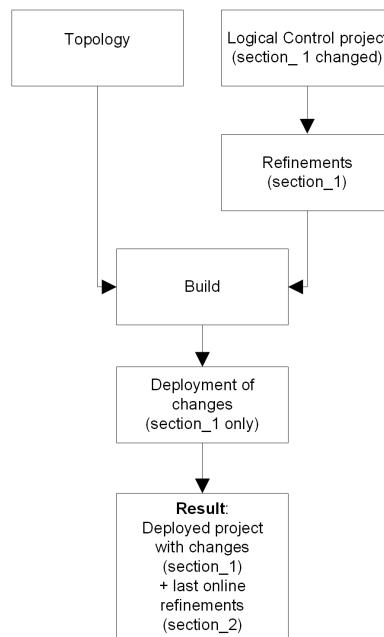


Workflow 2

When you make refinements online that you cannot or do not want to transfer to the Control Participant project, you can make these refinements in a new, separate section (section_2 in this example). You can unselect this section when deploying changes later on.



Making refinements online in a separate section (section_2 in this example) allows you to preserve them when you deploy changes (to topology, application, or in refinement (offline)) by unselecting the section that contains the online refinements (section_2).



Workflow 3

After you update the logical Control project with the following online refinements, the Control Participant reports a detected error during project analysis, page 388 because the DDDT variable does not exist in the logical Control project:

- You add a variable that references a DDDT variable, which exists only in the topology, page 507.
- You add an instance of an existing DFB type, which references a DDDT variable, which exists only in the topology.

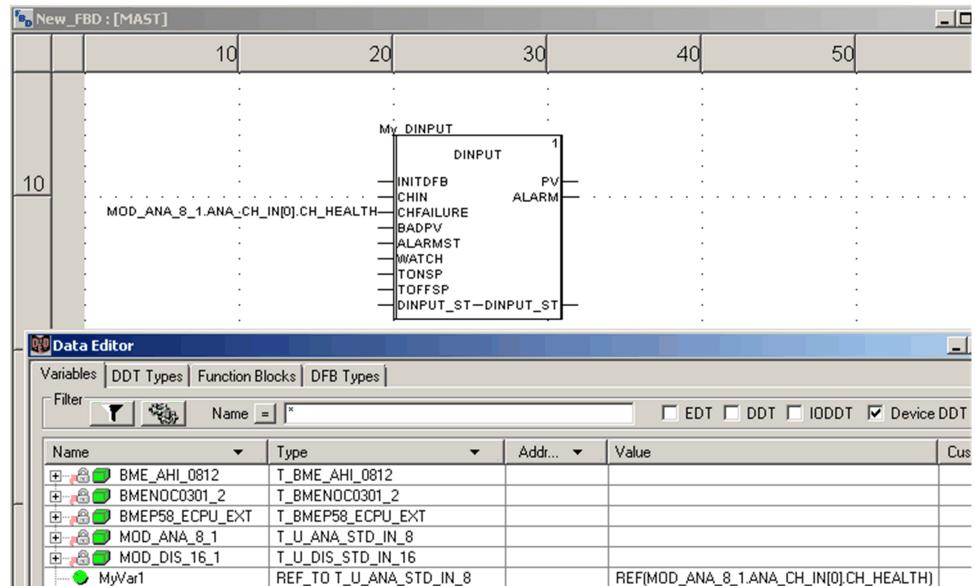
The solution is:

- When you add a variable that references a DDDT variable:
 - Do not use the variable in code that is generated by assigned facets or modified during refinement (offline) so that a subsequent deployment of changes does not undo online refinement.
 - Use the variable in a separate section, page 724, which you can unselect during deployment of changes.
- When you add an instance of an existing DFB type, which references a DDDT variable, build the Control project. This integrates variables coming from the topology into the built Control Participant project.

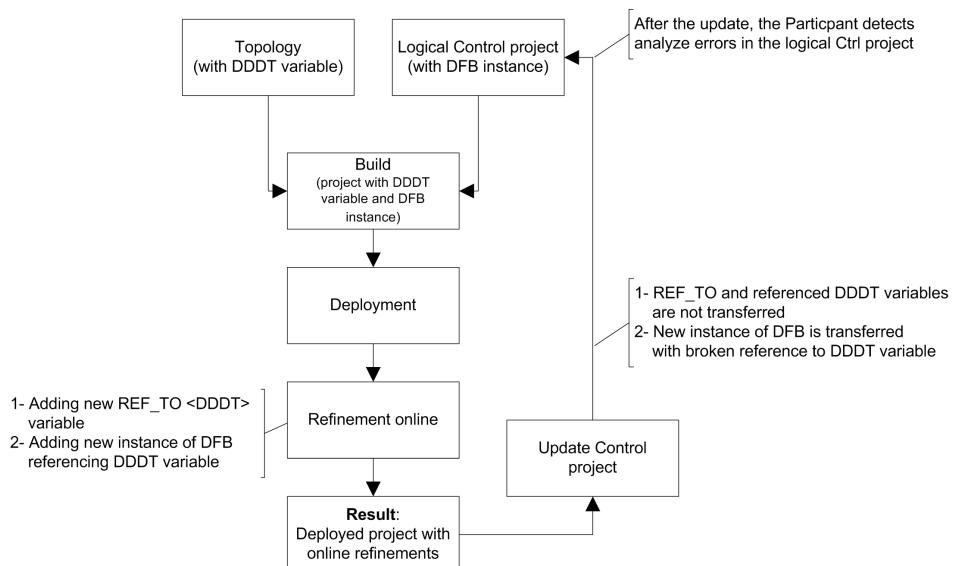
In this example, variable *MyVar1* is created during refinement online. It references the *MOD_ANA_8_1* DDDT variable of a HART analog input module, which comes from the topology. You cannot transfer *MyVar1* nor the *MOD_ANA_8_1* DDDT variable to the logical Control project.

Variables						
Filter						
Name	Type	Addr...	Value	Cus...	Constant	
BME_AHI_0812	T_BME_AHI_0812					
BMENOC0301_2	T_BMENOC0301_2					
BMEP58_ECPU_EXT	T_BMEP58_ECPU_EXT					
MOD_ANA_8_1	T_U_ANA_STD_IN_8					
MOD_DIS_16_1	T_U_DIS_STD_IN_16					
MyVar1	REF_TO T_U_ANA_STD_IN_8		REF(MOD_ANA_8_1.ANA_CH_IN[0].CH_HEALTH)			

In this example, the *My_DINPUT* DFB instance is created during refinement online from the DINPUT DFB type, which exists in the logical Control project. *My_DINPUT* references the *MOD_ANA_8_1* DDDT variable of a HART analog input module, which comes from the topology. You can transfer the *My_DINPUT* DFB instance to the logical Control project but the reference to the DDDT variable is broken.



The figure shows how online refinements that reference DDDT variables coming from the topology affect the deployment and execution workflow.



NOTE: You can create the DDDT variable in the logical Control project by using the wizard, page 445.

Making Changes Online

Overview

The **Refine Online** command opens the deployed Control Participant project in the Control Participant, allowing you to change the project when the engine is running or stopped.

You may be able to import sections, page 738 of the logical Control Participant project that have been modified and generated since their last deployment.

Online refinement requires using the Build and Deploy subcommand, which is described in Refining the Deployed Control Participant Project Online (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

When you close the Control Participant after deploying changes online, the software lets you update the associated logical Control Participant project right away with the changes you have made online. Until you do so, an online refinement indicator, page 500 is displayed in the corresponding topological entity folder. It indicates that the logical and the deployed Control Participant projects are different.

NOTE: To be applied, certain changes that you make online require stopping the controller, page 692.

Refining Online with a Station Node

When you select the **Refine Online** command for a station node (controller simulator), a message may appear asking you to deploy, page 650 the built Control Participant project if the previously deployed project is not loaded anymore into the controller simulator that is running on this station node.

Preserving Changes Made Online

After selecting the **Refine Online** command, if deployment of the following online changes does not complete successfully, then these changes are not discarded:

- Deployment of changes that do not require stopping the engine.
- Deployment of changes that require stopping the engine.

In such case, select the **Refine Online** command again and continue your work.

NOTE: If the deployment of online changes requires disconnecting the engine, the changes are discarded.

Updating the Project with Changes Made Online

Once you have successfully deployed changes made online and close the Control Participant, the software prompts you to update the associated logical Control Participant project and the controller configuration with these changes. You are not prompted in case you have made changes that require stopping the controller.

If you accept, the **Update Project** dialog box, page 748 opens. From here on, you follow the same process as when you select the **Update Project** command. The same restrictions apply.

If you do not accept, the online refinement icon, page 500  is displayed in the topological entity folder. It indicates that changes made online need to be applied to the associated logical Control Participant project and/or the controller configuration to make it consistent with the one that is deployed to the engine. The icon is displayed independently of the type of change that you make.

The icon is displayed and the software prompts you to perform the update until you complete the update process.

Refer to the topic describing how to update logical Control Participant projects, page 748.

NOTE: If you have changed the configuration online, you must perform the update to be able to use the **Deploy changes / Undo Online Changes** command.

NOTE: The icon is also displayed if you accept to perform the update but cancel the operation, if you update the project only partially, or if the update does not complete successfully.

Making Changes Online in Separate FBD Sections

When possible, it is good practice to make changes online in separate FBD sections that you create during online refinement. This is because you cannot transfer to the logical Control Participant project certain changes. It also applies when you do not update the logical Control Participant project after making changes online.

It helps you preserve changes that you have made online when you deploy changes, page 683 after building the logical Control Participant project.

Viewing the Application Structure When Making Changes Online

When you modify a Control Participant project online, the **Application Browser** displays FBD sections based on the location of instances whose Control facets are assigned to these sections.

It shows the relationship between the folder structure of the application and the sections of the Control Participant project. This is useful, for example, when you want to modify the logic related to a specific area or subdivision of the application. You can open sections from the pane.

The pane only shows FBD and non-FBD sections of the Control Participant project that are deployed at the moment you select the **Refine Online** command.

For a general description of the contents of the pane, refer to the topic describing the application structure view, page 371.

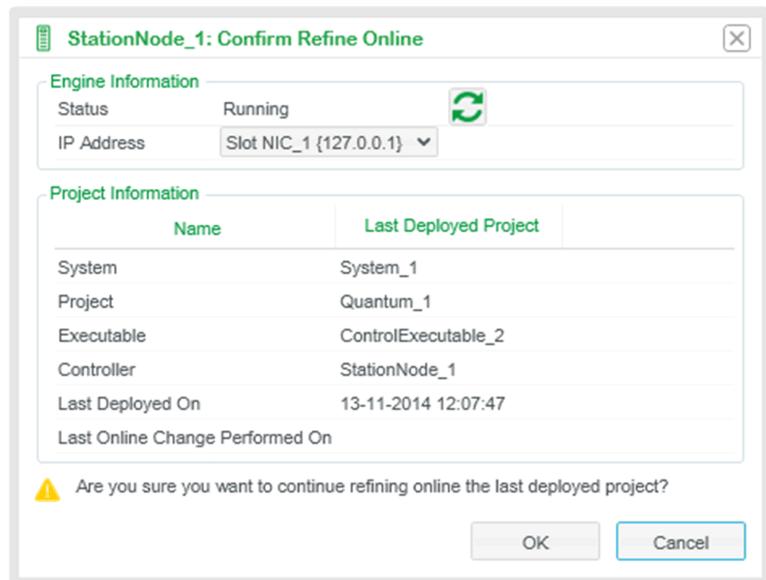
For information on how to use the pane, refer to the topic describing the refinement of Control Participant projects online (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

For general information on how to work with panes, refer to the topic describing the engineering client workspace, page 111.

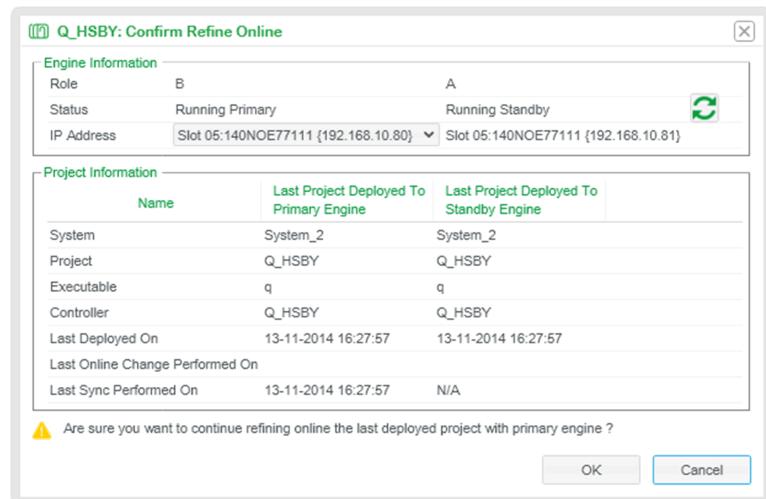
Confirming the Command

When you select the **Refine Online** command, before you can proceed with the online editing, the **Confirm Refine Online** dialog box opens. It contains information about:

- The engine whose program you want to refine online.
- The Control Participant project that is deployed to the engine.



The figure shows an example of the **Confirm Refine Online** dialog box for a redundant controller.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm Refine Online** dialog box.

Label	Description
Role⁽¹⁾	Role of the engine: <i>A</i> or <i>B</i> . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for online refinement. If the status is Not Reachable or Stop Offline⁽²⁾ you cannot proceed with the command. In such case, verify the IP address, the connection to the engine, or the status of the engine.
IP Address	Identifier of the communication module through which the software sends the command and its IP address that is configured in the Topology Explorer . If the engine has several communication modules, you can select one. NOTE: For controllers of the M580 platform, you cannot connect to an engine by using <i>IP address A</i> of the CPU module or a NOC communication module. You can only select main IP address, page 683. NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.

(1) Displayed only when you select the command for a redundant controller.

(2) Stop for controllers of the M580 platform.

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Refine Online** dialog box.

Name column	Last Deployed Project column ⁽²⁾
System	Identifier of the system that both the engine and the Control project, from which you have deployed the Participant project last belong to.
Project	Identifier of the Control Participant project associated to the deployed Participant project that you have deployed last.
Executable	Identifier of the Control executable that contains the built Participant project that you have deployed last.
Controller	Identifier of the topological entity to which you have deployed the built Participant project last.
Last Deployed On	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.
Last Online Change Performed On	Date and time when you last changed the deployed Participant project by using the Refine Online command; otherwise the field is empty.
Last Sync. Performed On⁽¹⁾	Date and time when you last synchronized the standby engine with the primary engine by using the Synchronize (Primary→Standby) command; otherwise the field is empty.

(1) Displayed only when you select the command for a redundant controller.
(2) When you select the command for a redundant controller, the dialog box shows separate columns for the primary and the standby engines.

Making Changes Online

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before deploying to an engine, verify that you have selected the correct project files.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Before starting or stopping a controller, verify that:

- You have assigned a valid IP address to the controller.
- The IP address corresponds to the physical address printed on the hardware.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: For more information, refer to Identifying Engines, page 651.

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

⚠ WARNING
UNKNOWN OPERATIONAL STATE OF EQUIPMENT
Before starting or stopping a controller, always positively confirm that there is no critical operation in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

When you deploy changes to a redundant controller, although deployment to the primary controller succeeded, it is possible that synchronization of the standby controller, page 744 does not complete. In such case, a message appears in the notification panel to inform you.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
<ul style="list-style-type: none"> Verify that deployment to the standby controller completed successfully. Do not operate the primary controller if deployment to the standby controller does not complete successfully. Synchronize the standby controller manually, using the Sync (Primary→Standby) command and verify that the synchronization completed successfully.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

When you make changes by online refinement, the associated logical and built Control Participant projects do not contain these changes. To preserve these changes during a next deployment, you need to integrate them into the built Control Participant project first by updating the logical Control Participant project, page 748, the controller configuration, and building the project; otherwise, the changes are discarded.

⚠ WARNING
LOSS OF DATA
After building and deploying changes made during online refinement, update the associated logical Control Participant project.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To edit the deployed Control Participant project online, proceed as follows.

Step	Action	Result
1	Verify that the engine is connected to the PC, page 657 and that it is running or stopped.	–
2	In the Topology Explorer , right-click the engine and select Refine Online .	The Confirm Refine Online dialog box opens.
3	Verify the information that is displayed.	–
4	Click OK . NOTE: Click Cancel to close the Confirm Refine Online dialog box without editing the Participant project.	The Reconfirm Refine Online dialog box opens.
5	Click OK . NOTE: Click Cancel to close the Reconfirm Refine Online dialog box without editing the Participant project. This reverts to the Confirm Refine Online dialog box.	The Control Participant opens extracted, page 116 and maximized.

Step	Action	Result
6	Proceed with changes or import of sections in the deployed Control project inside the Control Participant UI (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).	–
7	Close the Control Participant window.	The Refine Online dialog box opens prompting you to update the associated logical Control Participant project with the changes you have made online if applying these changes did not require stopping the controller. NOTE: Not updating the associated logical Control Participant project may result in losing online refinement changes during next deployments.
8	Click Yes . NOTE: Click No to close the Refine Online dialog box without updating the logical Control Participant project and display an online refinement indicator, page 500 in the associated topological entity folder.	The Update Project dialog box opens, which allows you to view and select changes made online that you can transfer to the associated logical Control Participant project and controller configuration.
9	Follow the procedure to update logical Control Participant projects, page 748 and controller configurations.	–

NOTE: If you close the Control Participant window without making changes online, the **Refine Online** online dialog box may still open, which prompts you to update the associated Control project. This is the case if you had made changes online previously, which you have not yet applied to the associated logical Control Participant project. The online refinement icon is also displayed in the associated topological entity folder to indicates this.

Backing Up Controller Data

Overview

The **Back Up Data** command lets you back up and save to file (.DTX) the following controller data:

- Located data of boolean type (%M).
- Located data of WORD type (%MW).
- Unlocated variables.
- Unlocated function block instances (DFB and EFB).

The functionality is available once you have deployed a built Control Participant project to a controller or station node (controller simulator) of the system.

Data backup files:

- Are associated to information such as a time stamp, which lets you select the correct file to deploy data again, page 677 and manage backup files, page 286.
- Retain your backup settings making it easy to reuse the same settings for subsequent backup operations on the same deployed executable.
- Can be exported and imported, page 265 with the Control Participant project.

NOTE: For more information, refer to *Save/Restore Data Between a File and the PLC* in the help of the Control Participant, page 93.

Backing Up Data with Redundant Controllers

For redundant controllers, back up of data is possible only from the primary controller.

To be able to back up data from the primary controller, the standby controller must be reachable at the IP address that is configured. It may not be reachable, for example, because it is switched off or the Hot Standby cable is removed.

Only if the **ControllerExist** property, page 509 is set to **Primary** you can back up data from the primary controller while the standby controller is not reachable.

NOTE: The status of the standby controller is not shown in the **Confirm Back Up Data** dialog box.

Backup File Identification

When the software creates a backup file, it associates the following information to the file:

- A date/time stamp.
- A description entered by the user.
- The name of the user who is logged on to the client.
- Identifiers of the controller, the Control Participant project, and the executable.
- The type of data that is backed up.

This information is shown for existing backup files when you select the **Deploy Data** command.

NOTE:

- When you delete a Control Participant project or a controller, its data backup files are deleted as well.
- If you change the identifier of a controller or executable, or the role of a controller, the identification information of the corresponding backup file is not modified.

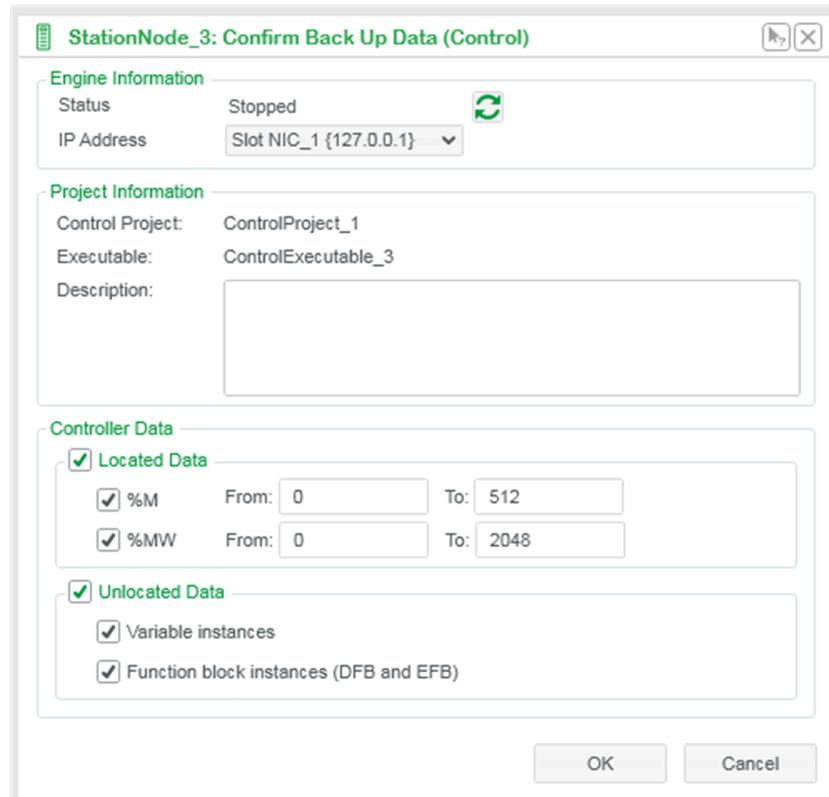
Controller and Controller Data Information

Before proceeding with the backup of controller data, the software displays information in the **Confirm Back Up Data** dialog box about:

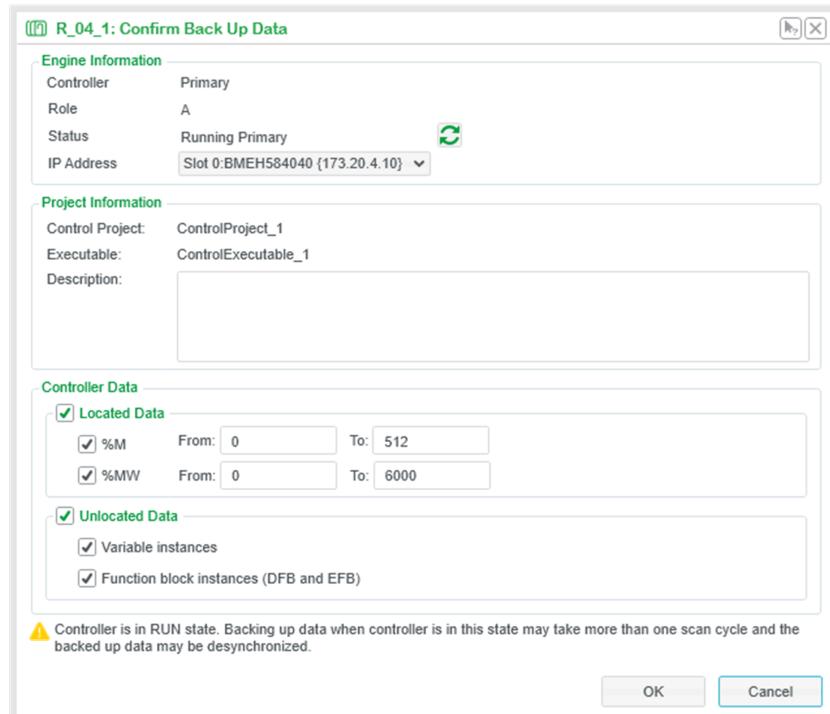
- The executable whose data you want to back up.
- The associated Control Participant project.
- The engine to which the executable is deployed.

This dialog box lets you also select the data that you want to back up.

The following figure shows an example of the **Confirm Back Up Data** dialog box for a standalone controller.



The following figure shows an example of the **Confirm Back Up Data** dialog box for a redundant controller.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm Back Up Data** dialog box.

Label	Description
Controller⁽¹⁾	You can perform the backup only from the primary controller. Default value: Primary
Role⁽¹⁾	Role of the selected controller: <i>A</i> or <i>B</i> . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected for backup. If the status is Not Reachable , you cannot proceed with the deployment. In such case, verify the IP address, the connection to the engine, and the status of the engine. NOTE: If the status is Running , a notification is displayed in the dialog box to inform you about a possible desynchronization of the backed up data.
IP Address	Identifier of the communication module through which the software will back up data and its IP address that is configured in the Topology Explorer . If the topological entity has several communication modules, you can select which one you want to use. Default value: IP address that was selected for the last deployment operation to this engine. NOTE: For controllers of the M580 platform, you cannot back up data from the CPU module or through a NOC communication module by using <i>IP address A</i> . You can only select main IP address, page 683. NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.

(1) Displayed only when you back up data from a redundant controller.

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Back Up Data** dialog box.

Label	Description
Control Project	Identifier of the Control Participant project associated to the executable that you have deployed last.
Executable	Identifier of the Control executable that contains the built Participant project that you have deployed last.
Description	You need to enter a description for the backup file by using free form text to be able to proceed with the backup.

The table describes the information that is displayed in the **Controller Data** section of the **Confirm Back Up Data** dialog box.

Section	Description
Located Data	<p>Select the check boxes depending on the type of located variables that you want to back up to file:</p> <ul style="list-style-type: none"> • %M: When selected, backs up located boolean variables. You need to provide the memory range to be backed up. Default range: 0 to maximum configured memory size -1. • %MW: When selected, backs up located variables of WORD data type. You need to provide the memory range to be backed up. Default range: 0 to maximum configured memory size -1. <p>NOTE: The maximum configured memory size depends on the memory configuration of located data of the CPU module of the controller to which the Control project is deployed.</p> <p>Selecting/clearing the Located Data check box also selects/clears the other check boxes in this section.</p> <p>Default value for first data backup for an executable: True (selected). If you have already performed a data backup operation for the same executable, your last settings are retained. The settings are reset when you restore a system or import the topology.</p>
Unlocated Data	<p>Select the check boxes depending on the type of located variables that you want to back up to file:</p> <ul style="list-style-type: none"> • Variable instances: When selected, backs up unlocated variable instances. • Function block instances (DFB and EFB): When selected, backs up unlocated function block instances. <p>Selecting/clearing the Unlocated Data check box also selects/clears the other check boxes in this section.</p> <p>Default value for first data backup for an executable: True (selected). If you have already performed a data backup operation for the same executable, your last settings are retained. The settings are reset when you restore a system or import the topology.</p>

Control Project Signature Verification

Before executing the command, the software compares the Control Participant project that is deployed in the engine with the last deployed reference, page 650, which is the Control Participant project residing in the software.

If they are identical, the command is executed. If they differ, the software starts a recovery process first. This process may take several minutes to complete.

In case the software displays a notification about a signature mismatch, the command cannot be executed. Contact Schneider Electric support, page 104.

Backing Up Controller Data

Unintended situations can occur if the operations that are in progress are not known when acting on a controller.

Backing up data when the controller is in **RUN** state may take several scan cycles and the backed up data may be desynchronized. This adversely affects program execution when the backed up data is deployed to the controller.

⚠ WARNING	
UNINTENDED EQUIPMENT OPERATION	
<p>Before backing up data from the controller, verify that the controller state does not jeopardize data integrity when the data is backed up.</p> <p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p>	
<p>NOTE: Schneider Electric recommends backing up data while the controller is in STOP state.</p>	

To back up the controller data, proceed as follows.

Step	Action
1	Verify that the engine can communicate with the PC, page 657.
2	In the Topology Explorer , right-click the entity whose data you want to back up and select the Back Up Data command. Result: The Confirm Back Up Data dialog box opens.
3	In the IP Address field, verify that the communication module/controller simulator identification is correct.
4	Verify the status of the engine in the Status field. NOTE: Backup is possible only if the status of the engine is indicated as either RUN , STOP (recommended), or WAIT (OFFLINE for Hot Standby controllers of the Quantum platform).
5	Verify the information that is displayed in the Project Information section.
6	Enter a description for the backup file.
7	Configure the data that you want to back up in the Controller Data section.
8	Click OK . Result: The Reconfirm Back Up Data dialog box opens. NOTE: Click Cancel to close the Confirm Back Up Data dialog box without backing up data.
9	Click OK . Result: <ul style="list-style-type: none"> • The software backs up the selected controller data and creates the corresponding backup file. • It provides information about the backup operation in the notification panel (or in a dialog box in case the operation did not succeed). NOTE: Click Cancel to close the Reconfirm Back Up Data dialog box without backing up data.

Importing Sections from the Logical Control Participant Project

Overview

It is possible to import FBD sections directly from the logical Control Participant project to help speed up the implementation of changes during commissioning. This is achieved by skipping the build and deployment stages of the system engineering life cycle.

The import functionality allows you to import into the deployed Control project, FBD sections of the corresponding logical Control Participant project, which you have modified since they were deployed last. The modifications need to be generated, page 416 to be imported.

The time stamp of the last generation of a section, page 336 is used as reference.

Changes made to a section by using the **Refine** command can be imported only if you have generated the section afterwards.

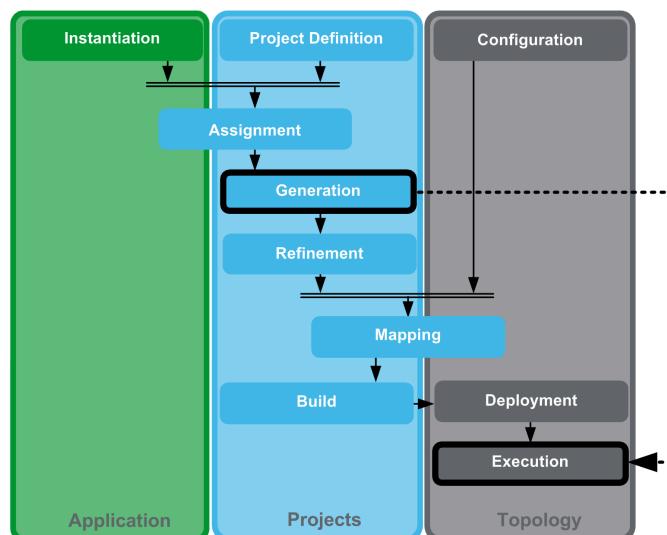
Refining a section by using the **Refine** command after generating it disables the possibility to import it.

Variables and types that are not used anymore after an import are not removed.

NOTE: If the section that you import has the same name as a section in the deployed Control project, it replaces the existing section. As a result, changes you had made in the section are overwritten.

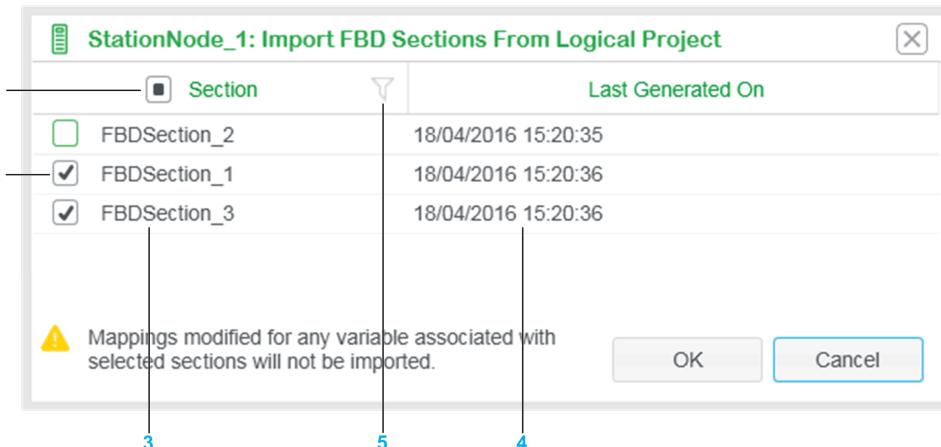
Section Import Functionality in the System Engineering Life Cycle

In the following figure, the dotted line shows the scope of the section import functionality in the context of the system engineering life cycle. Modifications that you make after generating a section and that pertain to the refinement, mapping, and build stages cannot be imported.



Section Import Window

The figure shows an example of the section import window, which opens when you click the section import button in the toolbar of the Control Participant window during online refinement.



Item	Description
1	Check box to select or clear all the sections displayed in the import window. A black square inside the check box indicates that at least one section is not selected.
2	Check box to select or clear sections individually.
3	Identifier of sections of the corresponding logical Control project, which have been modified and generated since the build Control project was last deployed to the engine. A section appears only if the change is supported, page 739 by the section import functionality. Sections are listed in ascending Order (Containers pane of the Assignment Editor).
4	Value of the <i>Generated On</i> property of the section.
5	Filter, page 118 for the Section column.

Attributes Related to HAL and Network Variables

When you import a section that contains a variable generated by a HAL template (for example, a signal conditioning template) or peer to peer communication templates (for example, \$DOUTBool_UL and \$DINBool_UL, which generate a variable with P2P as custom attribute), and the section and variable already exist in the deployed Control project, the *Address* and *Value* attributes of the variable are not modified by the import operation.

Online Refinement Indicator

When you import a section, build, and deploy the change, the online refinement icon, page 500 is displayed and you are prompted to update the associated logical Control project. However, if you have not made any other change apart from importing the section, the logical Control project is already up-to-date. Accepting to update the logical Control project displays a message informing you that no changes are detected.

Changes to Sections That Can Be Imported During Online Refinement

The table indicates whether a section can be imported during online refinement depending on the change that you have made. It is assumed that the section is generated after making the change (if required to apply the change).

Change description	Allows importing the section	Comment
Assigning a facet to the section.	Yes	–
Unassigning a facet from the section.	Yes	–
Actions resulting in a change of the status of a facet.	Yes	Except unlinking and relinking a facet.
Regenerating the section (independently if changes were made in the section before regenerating).	Yes	–
Renaming the section.	Yes	The renamed section is imported as a new section. The section with the original name that existed remains.
Changing the order of sections.	Yes	–
Creating a new section without assigning facets.	Yes	–
Renaming a variable (for example, by changing the identifier of the instance referencing the facet that creates the variable). The variable and section already exist in the deployed Control project.	Yes	The renamed variable is created as a new variable. The variable with the original name that existed remains.
Creating a section and making a change by using the Refine command (for example, adding a function block). Then, making a change in the section that can be imported (for example, assigning a facet).	Yes	The section is imported and also contains the change you had made through refinement.
Making a change in a section that can be imported (for example, assigning a facet) while a section with the same name already exists in the deployed Control project.	Yes	The section in the deployed Control project is replaced by the imported section.
Changing the order of facets inside the section.	No	–
Deleting a section to which facets are assigned and that is generated.	No	You cannot remove a section from the deployed Control project by using the section import functionality.
Making a change in the section that can be imported (for example, assigning a facet), then making a change in the section by using the Refine command (for example, adding a function block or modifying a variable used in this section).	No	–
Making changes in a section that can be imported and deploying these changes by using the Deploy Changes / Undo Online Changes command.	No	The changes are already applied to the deployed Control Participant project. If you have made changes that can be imported in more than one section but have not selected all these sections during deployment of changes, the sections that you have not selected appear as candidates for import.
1. Deploying a section. 2. In the Project Explorer making a change in the section that can be imported (for example, assigning additional facets). 3. Making a change in this section in the deployed Control project by using the Refine Online command and applying it by clicking the button to build and deploy (for example, removing a DFB).	No	You have deployed a change to a section that you want to import after generating this section.

Importing Sections from the Logical Control Participant Project

Refer to the topic describing how to refine the deployed Control project online (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

Main Redundant Controller Scenarios

Overview

This topic outlines the sequence of actions performed by the software when executing commands of the context menu of a redundant controller entity, which is composed of one primary and one standby engine. IP addresses IP and $IP + 1$ are assigned respectively to each engine.

Other scenarios are described in the topic documenting redundant Controller reservation and connection scenarios, page 744.

Deployment and Re-Deployment Sequence of Actions

Various deployment and re-deployment scenarios are considered depending on the status of the engines of the redundant controller before and after successful deployment.

Deployment is considered to require stopping engines.

Scenario 1: Both engines are stopped. No start after deployment.

	Actions
1	Deployment to $IP + 1$.
2	Deployment to IP .

Scenario 2: Both engines are stopped. Engines are started after deployment.

	Actions
1	Deployment to IP .
2	Start of IP .
3	Synchronization of $IP + 1$.
4	A successful synchronization starts $IP + 1$.

Scenario 3: Both engines are running. No start after deployment.

	Actions
1	Stop of $IP + 1$.
2	Deployment to $IP + 1$.
3	Stop of IP .
4	Deployment to IP .

Scenario 4: Both engines are running. Engines are started after deployment.

	Actions
1	Stop of $IP + 1$.
2	Stop of IP .
3	Deployment to IP .
4	Start of IP .
5	Synchronization of $IP + 1$.
6	A successful synchronization starts $IP + 1$.

Deployment of Changes and Online Refinement Sequence of Actions

Various scenarios for deploy changes and refine online are considered depending on:

- The status of the redundant controller before and after successful deployment.
- The need to stop the controller.
- The need to disconnect the engine.
- The status of the redundant controller after successful deployment.

Scenario 5: *IP* is stopped. Disconnection required. No start after deployment.

	Actions
1	Disconnection of <i>IP</i> .
2	Project is rebuilt.
3	Deployment as per deployment scenarios 1 or 2.

Scenario 6: *IP* is stopped. Disconnection not required. Engines started after deployment.

	Actions
1	Deployment as per deployment scenario 2.

Scenario 7: *IP* is stopped. Disconnection not required. No start after deployment.

	Actions
1	Changes are built in <i>IP</i> .

Scenario 8: *IP* is stopped. Disconnection not required. No start after deployment.

	Actions
1	Changes are built in <i>IP</i> .
2	Deployment to <i>IP + 1</i> .

Scenario 9: *IP* is running. Engine stop not required. Disconnection not required.

	Actions
1	Changes are built in <i>IP</i> .
2	Synchronization of <i>IP + 1</i> .

Scenario 10: *IP* is running. Engine stop required. Disconnection not required. Engines started after deployment

	Actions
1	Deployment as per deployment scenario 4.

Scenario 11: *IP* is running. Disconnection not required. Engine stop required. No start after deployment

	Actions
1	Stop of <i>IP + 1</i> .
2	Stop of <i>IP</i> .
3	Changes are built in <i>IP</i> .

Scenario 12: *IP* is running. Disconnection required. Engine stop not required. No start after deployment.

	Actions
1	Disconnection of <i>IP</i> .
2	Project is rebuilt.
3	Deployment as per deployment scenarios 3 or 4.

Redundant Controller Reservation and Connection Scenarios

Overview

This topic describes the actions that the software performs on the primary and standby engines for each deployment and execution command, depending on:

- The reservation state of the engine.
- The status of the connection to the engine.
- The status of the engine.

It is considered that the redundant controller is composed of a primary and a standby engine to which IP addresses IP and $IP + 1$ are assigned respectively.

For the Deploy Built Project and Re-Deploy Last Project Commands

The table describes the actions that the software performs on the primary and standby engines when you select the **Deploy Built Project** and **Re-Deploy Last Project** commands.

Engine External- ly Re- served	Controller IP Connected (Controller State)	Controller $IP + 1$ Connected (Controller State)	Action Executed by the software on Engine		Comment
			Primary	Stand- by	
primary	true	indifferent	no action		Primary engine is reserved.
standby	true (stopped)	true	deployment	no action	Standby engine is reserved.
standby	true (running)	true	no action		Standby engine is reserved and cannot be stopped.
none	false	indifferent	no action		Unable to connect to controller IP .
none	true (running)	false	no action		Unable to connect to controller $IP + 1$.
none	true (stopped)	false	deployment	no action	Unable to connect to controller $IP + 1$.
none	true	true	deployment to both		If the engines need to be stopped for the deployment, the software stops the primary engine even if the standby engine is already stopped, given that: <ul style="list-style-type: none"> • none of the engines is reserved • the connection status for both engines is true

For the Deploy Changes/Undo Online Changes Command

The table describes the actions that the software performs on the primary and standby engines when you select the **Deploy Changes/Undo Online Changes** command.

Engine External- ly Re- served	Controller <i>IP</i> Connected (Controller State)	Controller <i>IP + 1</i> Connected (Controller State)	Action Executed by the software on Engine		Comment
			Primary	Stand- by	
primary	true	indifferent	no action		Primary engine is reserved.
none	false	indifferent	no action		Unable to connect to controller <i>IP</i> .
standby	true	indifferent	no action		Standby engine is reserved.
none	true (stopped)	indifferent	deployment of changes	no action	No synchronization of standby controller because primary controller is stopped.
none	true (running)	true	deployment of changes		If the engines need to be stopped for the deployment of changes, the software stops the primary engine even if the standby engine is already stopped, given that: <ul style="list-style-type: none"> • none of the engines is reserved • the connection status for both engines is true

For the Start Command Only

The table describes the actions that the software performs on the primary and standby engines when you select the **Start** command.

Engine External- ly Re- served	Controller <i>IP</i> Connected (Controller State)	Controller <i>IP + 1</i> Connected (Controller State)	Action Executed by the software on Engine		Comment
			Primary	Stand- by	
primary	true (running)	indifferent	no action		Primary engine is already running.
primary	true (stopped)	indifferent	no action		Primary engine is reserved.

Refer to For the Start and Stop Command, page 746 for additional scenarios.

For the Stop Command Only

The table describes the actions that the software performs on the primary and standby engines when you select the **Stop** command.

Engine External- ly Re- served	Controller <i>IP</i> Connected (Controller State)	Controller <i>IP + 1</i> Connected (Controller State)	Action Executed by the software on Engine		Comment
			Primary	Stand- by	
primary	true (stopped)	indifferent	no action		Primary engine is already stopped.
primary	true (running)	indifferent	no action		Primary engine is reserved.

Engine External- ly Re- served	Controller IP Connected (Controller State)	Controller IP + 1 Connected (Controller State)	Action Executed by the software on Engine		Comment
none	true (running)	true (stopped)	Stop	no action	
standby	true (running)	true (running)	no action		Standby engine is reserved and cannot be stopped. Stopping primary engine causes switchover.
none	true (running)	false	no action		Unable to connect to controller IP + 1.
none	true (running)	true (running)	stop		The software stops in the following order: <ul style="list-style-type: none">• the standby engine• the primary engine

Refer to For the Start and Stop Command, page 746 for additional scenarios.

For the Start and Stop Commands

The table describes the actions that the software performs on the primary and standby engines when you select the **Start** or **Stop** command.

Engine External- ly Re- served	Controller IP Connected (Controller State)	Controller IP + 1 Connected (Controller State)	Action Executed by the software on Engine		Comment
			Primary	Stand- by	
standby	true	true	no action		Standby engine is reserved.
none	false	indifferent	no action		Unable to connect to controller IP.
none	true	false	no action		Unable to connect to controller IP + 1.

For the Refine Online Command

The table describes the actions that the software performs on the primary and standby engines when you select the **Refine Online** command.

Engine External- ly Re- served	Controller IP Connected (Controller State)	Controller IP + 1 Connected (Controller State)	Action Executed by the software on Engine		Comment
			Primary	Stand- by	
primary	true	indifferent	no action		primary engine is reserved.
standby	not required	true	no action		standby engine is reserved.
Standby-Later ⁽¹⁾	true	indifferent	no action		standby engine is reserved.
Standby-Later ⁽²⁾	true	indifferent	no action or deployment	no action	No synchronization of standby because reserved.

Engine External- ly Re- served	Controller <i>IP</i> Connected (Controller State)	Controller <i>IP + 1</i> Connected (Controller State)	Action Executed by the software on Engine		Comment
					Changes may have been deployed to primary engine.
none	false	indifferent	no action		Unable to connect to controller <i>IP</i> .
none	true	true	deployment of project changes		If the engines need to be stopped for the deployment of online changes, the software stops the primary engine even if the standby engine is already stopped, given that: <ul style="list-style-type: none"> • none of the engines is reserved • the connection status for both engines is true

(1) The engine is reserved after the software opens the Control Participant window in the **Topology Explorer**.
(2) The engine is reserved after the software executes the Build & Deploy, page 708 subcommand.

For the Sync (Primary→Standby) Command

The table describes the actions that the software performs on the primary and standby engines when you select the **Sync (Primary→Standby)** command.

Engine External- ly Re- served	Controller <i>IP</i> Connected (Controller State)	Controller <i>IP + 1</i> Connected (Controller State)	Action Executed by the software on Engine		Comment
			Primary	Standby	
primary	true	indifferent	no action		primary engine is reserved.
standby	true	true	no action		standby engine is reserved.
none	false	indifferent	no action		Unable to connect to controller <i>IP</i> .
none	true	false	no action	synchron- ize standby	–

Updating the Logical Control Participant Project and Controller Configuration

Overview

The **Update Project** command allows you to apply to the logical Control Participant project and the controller configuration changes that you have made online, page 723. You must have made the changes by using the **Refine Online**, page 727 command.

An indication that online changes need to be applied is given by an online refinement icon, page 500 that is displayed in the associated topological entity folder.

You can apply changes pertaining to the following categories:

- Animation tables
- Operator screens
- Sections
- Types
- Variables
- Hardware

Applicable restrictions are described in this topic.

In addition, for M580 safety Control projects, page 759:

- Process (non-safe) program interfaces
- Safe program

When you select the **Update Project** command, the software compares the deployed Participant project that is currently loaded in the engine with the corresponding logical Participant project and controller configuration and identifies differences.

The software allows you to select, which of the identified changes you want to apply.

You can request the software to perform a consistency check before updating the logical Participant project.

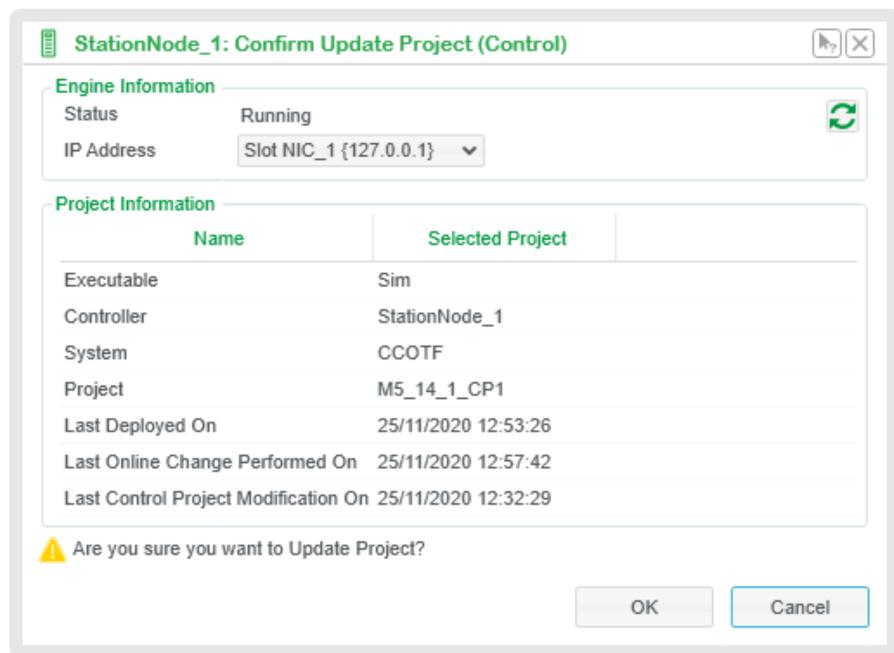
If you changed the logical Participant project since the deployment, using the **Update Project** command may override some of these changes.

NOTE: You need to build the Participant project, page 629 to which you have applied changes by using the **Update Project** command to be able to deploy these changes.

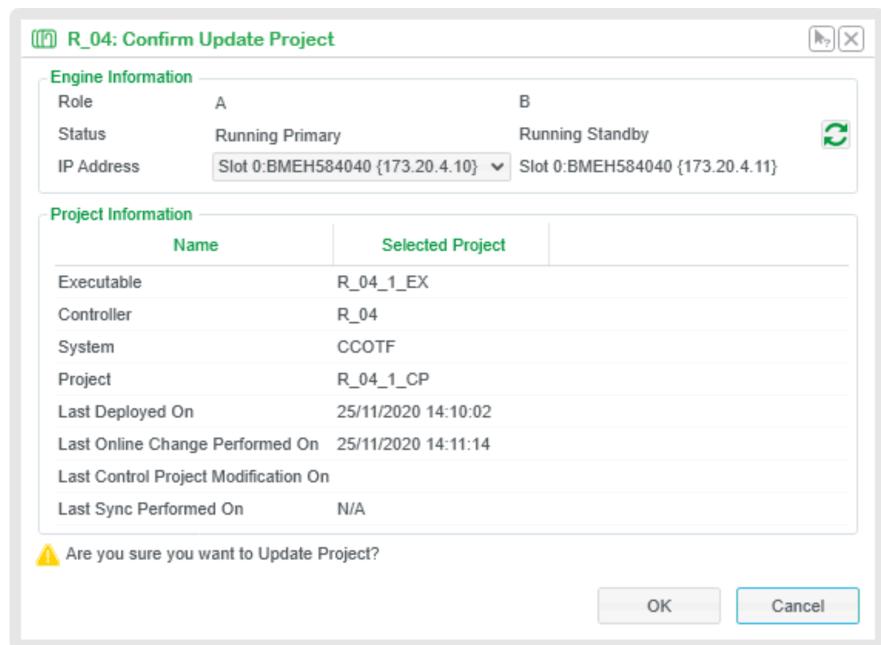
Confirming the Command

When you select the **Update Project** command, before you can proceed with the selection of changes, the **Confirm Update Project** dialog box opens. It contains information about:

- The engine from which you are updating the Control Participant project.
- The Control Participant project that is deployed to the engine.



The following figure shows an example of the **Confirm Update Project** dialog box for a redundant controller.



The table describes the information that is displayed in the **Engine Information** section of the **Confirm Update Project** dialog box.

Label	Description
Role⁽¹⁾	Role of the engine: <i>A</i> or <i>B</i> . If the software cannot communicate with the engine, the role is indicated as Not Detected .
Status	Status of the engine that is selected to update the Control Participant project from. If the status is Not Reachable you cannot proceed with the command. In such case, verify the IP address, the connection to the engine, and the status of the engine.

Label	Description
IP Address	<p>Identifier of the communication module through which the software sends the command and its IP address that is configured in the Topology Explorer. If the engine has several communication modules, you can select one.</p> <p>NOTE: For a station node, it is the combination of its NIC and the port that is configured for the Control service that is mapped to the executable.</p>
(1) Displayed only when you select the command for a redundant controller.	

NOTE: To refresh the engine status and attempt to connect to it, click the reconnect button, page 657.

The table describes the information that is displayed in the **Project Information** section of the **Confirm Update Project** dialog box.

Name column	Selected Project column
System	Identifier of the system that both the engine and the Control project belong to.
Project	Identifier of the Control Participant project associated to the deployed Participant project.
Executable	Identifier of the Control executable that contains the built Participant project.
Controller	Identifier of the topological entity from which you are updating the logical Control Participant project.
Last Deployed On	Date and time when you last deployed the Participant project to the engine or you last completed the Deploy Changes / Undo Online Changes command, independently if there were changes to be deployed or not.
Last Online Change Performed On	Date and time when you last changed the deployed Participant project by using the Refine Online command; otherwise the field is empty.
Last Control Project Modified On	Date and time when you last modified the logical Control Participant project by using the Generate or Refine command.
Last Sync. Performed On⁽¹⁾	Date and time when you last synchronized the standby engine with the primary engine by using the Synchronize (Primary→Standby) command; otherwise the field is empty.
(1) Displayed only when you select the command for a redundant controller.	

NOTE: The **Confirm Update Project** dialog box is not displayed if you choose to update the logical Control Participant project when prompted after completing a refine online operation, page 727.

Updating with Changes Other Than FBD Language

The table describes the effect of the **Update Project** command on items other than FBD language.

Item	Effect
Non-FBD sections	<p>Details of changes inside non-FBD sections that the software has detected are not shown. Only the name of the section is displayed.</p> <p>When selected, such sections are either updated, added, or removed from the logical Control Participant project.</p> <p>NOTE: If you update the logical Control Participant project with code in LL984 language added during online refinement and you had not selected the Ladder Logic 984 (LL984) setting in the Control Participant before deploying, the validity icon, page 324 is displayed after the update.</p> <p>To be able to build the logical Control Participant project select the Ladder Logic 984 (LL984) setting in project containing the configuration of the controller (see EcoStruxure™ Process Expert, Control Participant Services, User Guide) that is mapped to the executable of this Control project.</p>
Animation tables	<p>The software does not detect changes to such items.</p> <p>The software displays any animation tables and/or operator screens that the deployed Participant project contains, including modified and/or new ones. Animation tables and/or operator screens that you have deleted from the deployed Participant project are not shown.</p>
Operator screens	<p>You cannot select individual items but only the category for transfer to the logical Participant project.</p> <p>When you select a category, any animation tables or operator screens of the logical Control Participant project are overwritten with the ones contained in the deployed Participant Project.</p> <p>By default, both categories are not selected.</p> <p>NOTE:</p> <ul style="list-style-type: none"> Overwriting of an existing operator screen creates a copy of the screen in the logical Participant project. Following the update, use the Refine command to edit the categories and make their contents consistent with the deployed Control Participant project. Animation tables and/or operator screens that you have removed from the deployed Participant project are not removed from the logical Participant project by using the command. Use the Refine command to delete them.
Hardware	<p>Details of configuration are not shown.</p> <p>The Hardware category is selected by default (if changes are detected) and is read-only.</p> <p>If the controller configuration is modified because of the update operation, mappings may be deleted.</p> <p>NOTE: If you have changed the configuration online, performing the update is required to be able to use the Deploy changes / Undo Online Changes command.</p>

Updating with Changes to Variables

The table describes the behavior of the software when you update a logical Control Participant project, depending on the changes that you have made to variables contained in the deployed Control Participant project.

Variables in the Deployed Control Project	Behavior
Variables with HAL , page 639 or P2P , page 456 as Custom attribute.	No changes to their attributes are updated in the logical Control Participant project.
Variables without HAL , page 639 as Custom attribute.	Any attributes of such variables that you add or modify during refinement online are updated in the logical Control Participant project, except for the Address or Value attributes if it corresponds to or references a topological address. NOTE: For more information on variables using the Value attribute for topological addressing, refer to the topic describing REF_TO type variables (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).
Device DDT variables	These variables (and any of their attributes) coming from the configuration are not transferred to the logical Control Participant project. Therefore, if you add code during refinement online, page 723 that uses this type of variable, you can update the logical Control Participant project with that code but not with the variable itself. This causes the code to become invalid. NOTE: You can create the DDDT variable in the logical Control project by using the wizard, page 445.

Updating with Changes to Sections Containing HAL DFBs

You cannot transfer to the logical Control Participant project any changes that you have made or code that you have added to sections containing HAL DFB instances, page 639.

Changes That Cannot Be Transferred to Control Projects

You cannot transfer to the logical Control Participant project changes pertaining to any of the following categories:

- Control project settings (refer to the Control Participant help, *Operating Modes*).
- SR sections.
- Properties of MAST and FAST tasks.
- Properties of MAST and FAST sections⁽¹⁾.
- Timer and I/O event tasks.
- *Alias* of attribute of variables for Modicon X80 I/O modules (reference to the Device DDT variable).

(1) Except for changes in the **Comment** tab.

Identifying Project and Hardware Changes

Before proceeding with the update, the software displays the **Update Project** dialog box, which provides information about the differences that it has detected between the deployed Participant project and the logical Participant project. It also allows you to select, which changes you want to apply to the logical Control Participant project.

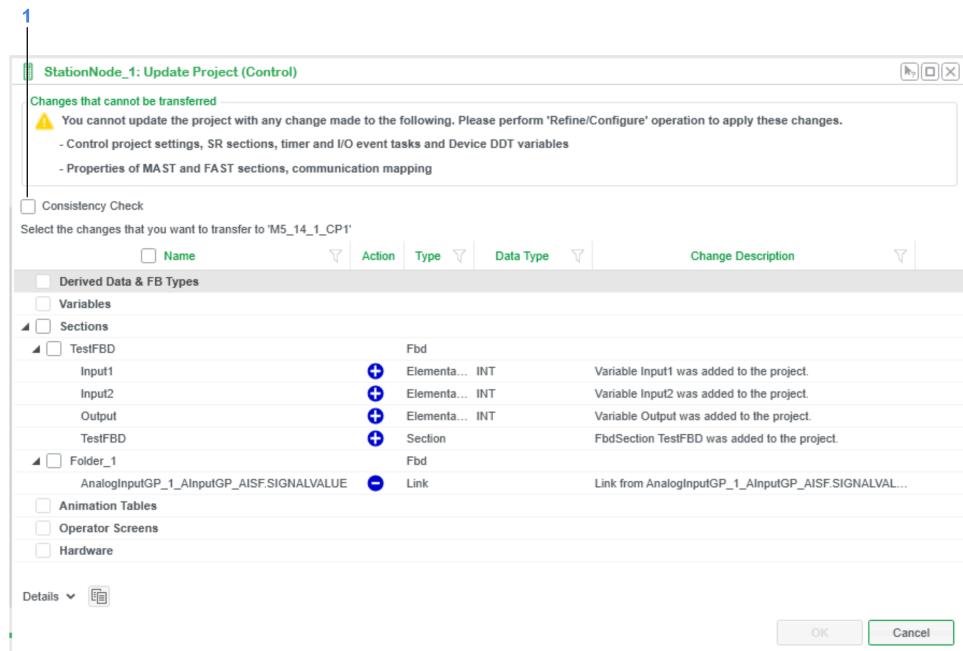
The software can only update the logical Participant project with changes that pertain to a category displayed in the **Name** column of the **Update Project** dialog box. Changes that are not detected or that are detected but that cannot be selected are not transferred.

When the software detects no changes, it does not display the **Update Project** dialog box but instead, shows a message to inform you that it has found no changes. This is the case if, for example, only a timer event was added during online refinement.

NOTE: When you select a section in order to apply changes to this section in the logical Control Participant project, the software overrides the refinements that you made in this section after the deployment. See also the topic that describes the impact of a Control project update, page 754 on sections that you have created after deployment.

NOTE: Restrictions may apply when updating variables, page 751.

The following figure shows an example of the **Update Project** dialog box.



1 Check box to have a consistency check, page 754 performed on the selected items before proceeding with the update. By default, the option is not selected.

Header	Description
Check boxes	Select items that you want to update in the logical Control Participant project. A green square inside the check box in the column header indicates that, at least, 1 item that the software has detected as a difference is still unselected. NOTE: You cannot select sections that contain HAL DFB instances, page 639.
Name	Identifier of the item that the software has detected as a difference. Items are grouped by categories of the Control Participant project.
Action	Indicates the action that the software will carry out for the item: <ul style="list-style-type: none"> Plus icon: The item is added to the logical Participant project/controller configuration. (for example, because the item has been added during online refinement or removed from the logical Participant project after the last deployment.) Minus icon: The item is removed from the logical Participant project/controller configuration. (for example, because the item has been removed during online refinement or added to the logical Participant project after the last deployment.) Pencil icon: The item is modified compared to the logical Participant project. NOTE: When you select a section, the software applies any changes that it has identified for this section. You cannot select/unselect changes individually inside a section.
Type	Type of the item.
DataType	Data type of the item, if applicable.
Change Description	For FBD sections, variables and types, indicates whether the change that the software has identified is an addition, a modification, or a deletion compared to the logical Participant project.

NOTE: If the software does not detect a difference between the deployed Participant project and the corresponding logical Participant project in a category of the **Update Project** dialog box, no entry appears under that category.

Checking Consistency

Select the corresponding check box in the **Update Project** dialog box for the software to perform a consistency check, page 449 before proceeding with the update of the logical Participant project.

In such case, the **Check Consistency** dialog box opens. It shows the following for the changes that you have selected in the **Update Project** dialog box:

- The sections that contain constituents that are inconsistent.
- For each section, the facets that are assigned to the section and generate the corresponding constituents.
- For each facet, a description of the inconsistency.

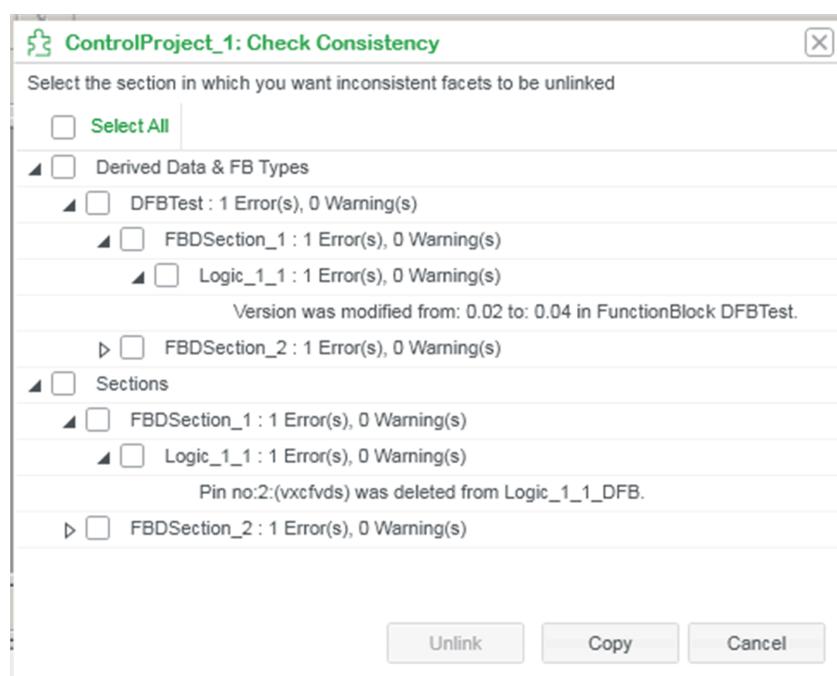
Select the check box next to a facet to unlink, page 411 it.

If the check box of a facet is cleared, the generation status of the facet remains **Inconsistent**, page 411.

During a subsequent generation, page 417 of the updated logical Participant project, the software:

- Does not modify constituents of facets that are unlinked.
- Attempts to restore constituents of facets that are inconsistent.

By default, no items are selected.



NOTE: If the software does not detect inconsistencies, the **Check Consistency** dialog box does not open.

Updating After Creating Sections in the Logical Control Project Without Deploying Them

After updating the logical Control project, a section is moved to the end of sections in this project (highest order value) if the following conditions are fulfilled:

- The logical Control project is deployed.
- You create a section either in the **Project Explorer** or during refinement (offline) post deployment.

- You do not deploy this section.
- The section that you have created occupies the same position as a section in the deployed Control project or you select the section in the **Update Project** dialog box (delete).

It can also have an impact on facets that are assigned to this section.

The following example shows how a section created in the logical Control project after deployment can have the same position as a section in the deployed Control project.

Example: The logical Control project contains two sections. It is deployed.

Logical Control project (offline)	Deployed Control project (online)
Section_1 / order 0	Section_1
Section_2 / order 1	Section_2

Then, section_3 is created in between Section_1 and Section_2. It now has the same position as Section_2 in the deployed Control project.

Logical Control project (offline)	Deployed Control project (online)
Section_1 / order 0	Section_1
Section_3 / order 1	Section_2
Section_2 / order 2	

The table describes the possible scenarios where updating the logical Control project has an impact on the position of a section that you have created in it post deployment and/or on the status of the facets assigned to this section.

Assignment and generation status	The position of the section conflicts with the position of a section in the deployed project	Proposed action during Control project update	User selection	Result in logical Control project after the update
Facets are assigned to the section but none are generated.	Yes	None. The section created in the logical project is not detected as a difference because it is not generated.	Not applicable.	The section that was created in the logical project is moved to the end of sections (highest order value) ⁽¹⁾⁽²⁾ .
Facets are assigned to the section and generated.	No	Delete.	The section is selected for deletion.	<ul style="list-style-type: none"> The section created in the logical project is moved to the end of sections (highest order value)⁽²⁾. The generation status of facets is changed to Non Generated.
	Yes	Delete.	The section is selected for deletion.	<ul style="list-style-type: none"> The section created in the logical project is moved to the end of sections (highest order value)⁽²⁾. The generation status of facets is changed to Non Generated.
		Delete.	The section is not selected (default).	<ul style="list-style-type: none"> The section created in the logical project is moved to the end of sections (highest order value)⁽¹⁾. The generation status of facets is unchanged.

(1) After the update is complete, you may be able to preserve the change that you have made in the logical Control project by moving the section back to its original position and propagating the change in the system engineering life cycle.

(2) After the update is complete, to remove the section from the logical Control project, right-click it in the **Containers** pane of the **Assignment Editor** and select **Delete**.

NOTE: In the above scenarios, it is implied that other changes that you may have made during refinement online are applied to the logical Control project (for example, adding code to a section or changing the order of sections).

NOTE: The behavior is the same when you update the logical Control project if you create a section in the **Project Explorer**, generate, and deploy it and then delete it during refinement online (given the section is not locked, page 442).

Aborting Project Update Tasks

After confirming the update by clicking **OK** in the **Update Project** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The task is not updated.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are updating the Control Participant project with only a few changes, the task may complete before you are able to click the icon.

Updating the Project and Controller Configuration

To update the logical Control Participant project and/or the controller configuration, proceed as follows.

Step	Action
1	Verify that the engine is connected to the PC, page 657.
2	In the Topology Explorer , right-click the controller or station node containing the Participant project that you have refined online and select Update Project . Result: The Confirm Update Project dialog box opens. NOTE: For a station node, first, in the Update Project dialog box, you can select a Control project from a list of projects containing online changes (if applicable).
3	Verify the information that is displayed and click OK . Result: The Update Project dialog box opens. NOTE: Click Cancel to close the Confirm Update Project dialog box without applying the changes.
4	Select the changes that you want to update the logical Control Participant project with. NOTE: Click Cancel to close the Update Project dialog box without updating the Control Participant project.
5	Select the Check Consistency check box if you want the software to perform a consistency check; otherwise proceed to step 8.
6	Click OK . Result: The Check Consistency dialog box opens if you selected the option and inconsistencies are detected.
7	In the Check Consistency dialog box, select facets that you want to unlink. NOTE: Click Cancel to close the Check Consistency dialog box without unlinking facets and revert to the Update Project dialog box.
8	Click Unlink or OK . Result: The software: <ul style="list-style-type: none">• Applies the changes that you have selected.• Changes the status of the associated executable to Out Of Date.• Displays Completed in the information pane once it has successfully completed the update process.• Updates the related date/time field.• Removes the online refinement icon, page 500  from the associated topological entity only if you had selected to update the Control Participant project with all the detected changes of the following categories; otherwise the indicator remains.<ul style="list-style-type: none">◦ Derived Data & FB Types◦ Variables & FB Instances◦ Sections

Supervision Project Execution Stage

Executing the Deployed Supervision Participant Project

Overview

You need to execute the deployed Supervision Participant project locally, on each PC to which you have deployed the Supervision project by using the appropriate Supervision runtime.

For information on the required software, refer to the topic describing software requirements (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Executing a Deployed Supervision Project

For information on executing the deployed Supervision Participant project on a PC, refer to Executing the Deployed Supervision Project (see EcoStruxure™ Process Expert, Supervision Participant Services, User Guide).

Engineering M580 Safety Systems

What's in This Chapter

Using Modicon M580 Safety Controllers.....	759
M580 Safety Systems in the Instantiation Stage.....	760
M580 Safety Systems in the Project Definition Stage	761
M580 Safety Systems in the Assignment Stage	766
M580 Safety Systems in the Generation Stage	767
M580 Safety Systems in the Refinement Stage	767
M580 Safety Systems in the Configuration Stage.....	769
M580 Safety Systems in the Mapping Stage.....	774
M580 Safety Systems in the Build Stage	774
M580 Safety Systems in the Deployment Stage.....	775
M580 Systems in the Execution Stage	776
Using Supervision with M580 Safety Systems	777

This chapter describes the aspects that are specific to using an M580 safety configuration and its associated Control project (the *M580 safety system*) when engineering a system in EcoStruxure Process Expert. The information is grouped by stages of the system engineering life cycle.

Using Modicon M580 Safety Controllers

Certifications

The M580 safety controller (PAC) is certified by TÜV Rheinland Group for use in applications up to and including safety integrity level 3 (SIL3).

For additional information, refer to the topic describing certifications in the *Modicon M580 Safety Manual*, which you can find in the help of the Control Participant, page 93.

Control Participant of EcoStruxure Process Expert

EcoStruxure Process Expert 2020 R2 uses Control Expert XL Safety V15.0 as Control Participant.

The safety configuration and safety program of an M580 safety system are created and modified exclusively by the Control Participant.

Only the process (non-safety) areas and the non-safety modules of the M580 safety system are managed by EcoStruxure Process Expert.

Engineering Systems Containing M580 Safety Controllers

When you engineer an EcoStruxure Process Expert system that contains an M580 safety controller, in addition to following the normal system engineering life cycle, perform the following activities:

- Verify that the reference and version of the Control Participant is certified by the TÜV.

You can look up the TÜV-certified software tools and versions in the *Revision List*, which is in the appendices of the *M580 Functional Safety Controller System* certificate at www.certipedia.com or www.fs-products.com.

- Verify the integrity of the Control Participant (self-test).
- Use and monitor the safe signature and the safety application ID (SAId).

For details, refer to the topic describing building an M580 safety project in the *Modicon M580 Safety Manual*.

- Follow the safe application life cycle and apply the safety rules as described in the *Modicon M580 Safety Manual*, which you can find in the help of the Control Participant, page 93.

Information related to the following topics that are described in the *Modicon M580 Safety Manual* does not apply or applies only partially when working with the Control Participant.

Topic	Difference when using EcoStruxure Process Expert
<i>Usage of Process Namespace</i> project setting (Refer to the topic describing data separation in an M580 safety project)	The project setting, page 767 is not supported.
Access rights (Refer to the topic describing workstation security management)	User profiles and access rights for the Control Participant are managed by EcoStruxure Process Expert and cannot be modified. Users who can log in to the engineering client have full access to the functionality of the Control Participant.
Maintenance mode (Refer to the topic describing workstation security management)	You can switch the M580 safety controller to maintenance mode by using the button in the toolbar of the Participant window when you use the Refine Online command.
Building an M580 safety Control project (Refer to the topic describing the operation of an M580 safety system)	Use a build command in the context menu of the executable of the Control project in the Project Explorer .
Safe signature (Refer to the topic describing building an M580 safety project)	Using a build command, page 774 in the context menu of the executable of a safety Control project generates a new SourceSafeSignature if you have made an applicable configuration or value change.
Safe area password (Refer to the topic describing application security management)	The password protection, page 770 is managed in the Topology Explorer instead by using the Safety password.
Transferring M580 safety projects (Refer to the topic describing modifications to Control Expert for the M580 safety system)	The command to transfer the project from the controller to the Control Participant is disabled. To transfer the project to the controller, use only the deployment commands in the Topology Explorer .
CCOTF (Refer to the topic describing modifications to Control Expert for the M580 safety system)	CCOTF operations can be performed also when you configure, page 505 an M580 safety controller (offline).
Saving & restoring data between a file and the PAC (Refer to the topic describing the modifications to Control Expert for the M580 safety system)	The commands are disabled in the Control Participant. Use the Back Up Data and Deploy Data commands in the Topology Explorer instead.

Additional Information

For more information on M580 safety systems, refer to the *Modicon M580 Safety System Planning Guide*, which you can find in the help of the Control Participant, page 93.

M580 Safety Systems in the Instantiation Stage

This topic describes the aspects of the instantiation that are specific to using M580 safety systems. For general information, refer to the topic describing the instantiation stage.

Systems Explorer

You can use M580 safety controllers and their projects in any EcoStruxure Process Expert system.

A system can contain both safety and non-safety controllers and Control projects.

Instantiation Stage

No safety-specific templates are provided to create instances in the application of a system that uses an M580 safety controller. The safety data and logic can be created only by refining the safety Control project.

Instances that you create in the application can only be used to generate the following:

- The process data and logic in a safety Control project.
- Supervision project data.

M580 Safety Systems in the Project Definition Stage

This topic describes the aspects of the Control project definition that are specific to using M580 safety systems. For general information, refer to the topic describing the Control project definition stage.

Creating M580 Safety Control Projects

M580 Safety Control Projects

In a system, M580 safety Control projects can coexist with Control projects of any other controller family.

M580 Safety Control Project Content

An M580 safety Control project can contain the following:

- A safety program (**SAFE**)
- A non-safety program (**PROCESS**)
- Global data

The logic and data that is part of each program is isolated from the other.

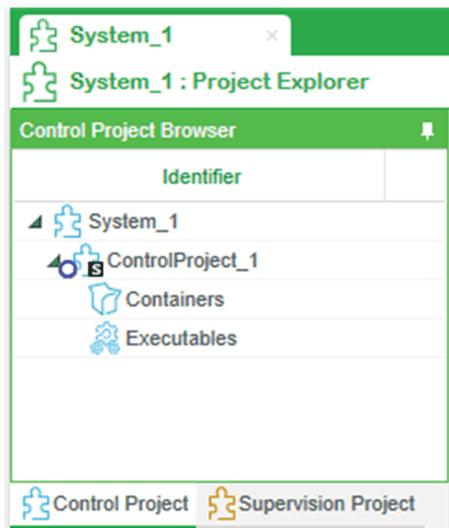
Creating M580 Safety Control Projects

To create an M580 safety Control project, right-click the system root folder in the **Project Browser** and select **M580 Safety**.

Creating such a project also creates an M580 safety controller with a default configuration in the **Topology Explorer**, which is associated to the project.

Conversely, if you have created an M580 safety controller, a blank safety Control project has been created in the **Project Explorer**.

The following figure shows an M580 safety Control project as it appears in the **Project Browser**.



Creating Sections, Logic, and Data

The following table indicates how to create the various types of program components of a safety Control project and where you can view them.

Type of logic and data	Ways to create	Viewing logic and data
Safety sections, logic, and data (includes animation tables)	By refining the Control Participant project only.	Visible only in the refinement window (offline and online) of the Control Participant.
Process sections, logic, and data	By following the normal system engineering life cycle. (Non-FBD sections and logic requires refining the Control Participant project.)	FBD and non-FBD sections as well as facets assigned to FBD sections are visible in the Assignment Editor .
Global data	By refining the Control Participant project only.	Visible only in the refinement window (offline and online) of the Control Participant.

Deleting M580 Safety Control Projects

Deleting a safety Control project also deletes the associated M580 safety controller entity in the **Topology Explorer**, and vice versa.

Managing Network Variables

The **Manage Network Variables** dialog box, page 459 does not let you manage variables of the safety program.

Supervision Projects

The Supervision projects that you create are independent of the M580 safety system.

You can create instances to be assigned to Supervision projects to monitor safety Control data, page 777.

Exporting and Importing M580 Safety Control Projects

This topic describes the aspects of the Control project export and import that are specific to M580 safety Control projects. For general information about project export and import operations, refer to the topic describing how to manage Control projects, page 265 of a system.

Exporting M580 Safety Control Projects

The **Export** command of the **Project Explorer** lets you export one or more safety Control projects and create a non-editable export file (.sbk).

You can also export at the same time Control projects of the other controller families and Supervision projects.

In the **Export** window, the **Safe Program** section of safety Control projects is selected by default and read-only. The safety program is exported as a whole.

The associated safety configuration is not exported.

Importing M580 Safety Control Projects

The **Import** command of the **Project Explorer** lets you import the M580 safety Control projects that are contained in a project export file (.sbk).

The table describes the result when you import an M580 safety Control project in different scenarios.

Target system	Result after import in Project Explorer	Impact on topology after import
An M580 safety Control project with the same identifier does not exist.	<p>The safety Control project is imported.</p> <p>If you select to import the safety program, it is imported as a whole.</p>	<p>An M580 safety controller entity is created with the default configuration⁽¹⁾.</p> <p>NOTE: If you want to import a safety Control project that is associated to an existing M580 safety controller in the Topology Explorer, before importing the safety project, rename the existing safety Control project, page 764 to have the same identifier as the one you will be importing.</p>
An M580 safety Control project with the same identifier already exists.	<p>If you select to import the safety program, it is imported as a whole.</p> <p>The import rules, page 271 that the software applies with regard to safety sections are the same as those for process sections.</p>	The existing M580 safety controller is not impacted and no additional one is created.
A Control project with the same identifier but of the M580 (<i>non-safety</i>) or another controller family already exists.	<p>You cannot import the M580 safety Control project.</p> <p>In the Import window, it is tagged with a red dot⁽²⁾.</p>	None.

(1) Schneider Electric recommends verifying that the Global Templates library of the target system contains the necessary topological safety templates (Foundation library) before importing. Otherwise, you will not be able to save your changes when you configure the associated safety controller in the **Topology Explorer**.

(2) The same also applies when you import any Control project and an M580 safety Control project with the same identifier already exists in the target system.

NOTE: During import of a safety Control project, the associated M580 safety controller is locked.

Importing a Control Expert Project

The **Import Control Project** command of the **Project Explorer** lets you create a new M580 safety Control project by using the data of a compatible Control Expert project export file (.stu or .sbk).

The following are created:

- An M580 safety Control project in the **Project Explorer**.
- An associated M580 safety controller with the same configuration and project settings as in the export file in the **Topology Explorer**.

The name of the imported file becomes the identifier of the safety Control project that is created. The associated M580 safety controller is created with the default identifier.

Safety Password Management When Importing

When you import one or more M580 safety Control projects, before the import operation starts, the **Verify Safety Password** window may open prompting you to enter a safety password for each one.

The import operation starts only once all required safety passwords have been entered.

NOTE: The safety password that is referred to is the one that has been set in the **Topology Explorer** for the associated M580 safety controller, page 770.

The following table describes which safety password you need to enter when prompted during the import of an M580 safety Control project. This applies even if you select not to import the safety program.

Target system	Safety password required	Impact on safety password after import
An M580 safety Control project with the same identifier does not exist.	The safety password of the safety Control project that you are importing (if set).	The safety password settings are the same as that of the imported safety Control project.
An M580 safety Control project with the same identifier already exists.	The safety password of the existing safety Control project in the target system (if set).	<p>The safety password and password settings are the same as that of the existing safety Control project in the target system.</p> <p>For example, if the existing safety Control project has no safety password but the safety Control project that you are importing has a safety password, after the import, no safety password is set.</p>

The following table describes which safety password you need to enter when prompted during the import of a Control Expert M580 safety Control project.

Target system	Safety password required	Impact on safety password after import
Irrelevant	The safety password of the safety Control project that you are importing (if set).	<p>The safety password settings are the same as that of the imported safety Control project.</p> <p>You can manage the safety password from the controller entity in the Topology Explorer.</p>

Importing a Safety Control Project Associated to an Existing M580 Safety Controller

By using an example, this topic illustrates the required workflow when you want to import an M580 safety Control project that is associated to an existing M580 safety controller or one that you will be importing. The workflow with M580 safety systems differs slightly because the import of a safety controller/project creates the counterpart safety project/controller.

Starting Point

You have exported the following to two separate export files (.sbk):

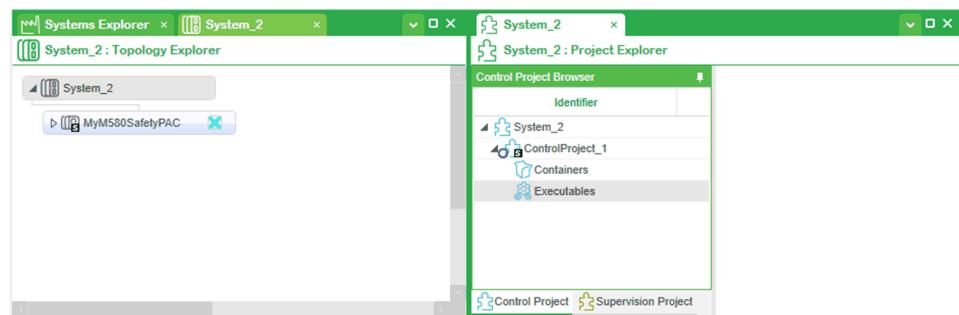
- The configuration of an M580 safety controller *MyM580SafetyPAC* from the **Topology Explorer**
- The associated M580 safety Control project *MyM580SafetyProject* from the **Project Explorer**

The executable of *MyM580SafetyProject* is mapped to *MyM580SafetyPAC* in the service mapping and both are exported.

In this example, the objective is to import into a different system the M580 safety controller and its project so that they are still associated and the executable of the project mapped to the controller.

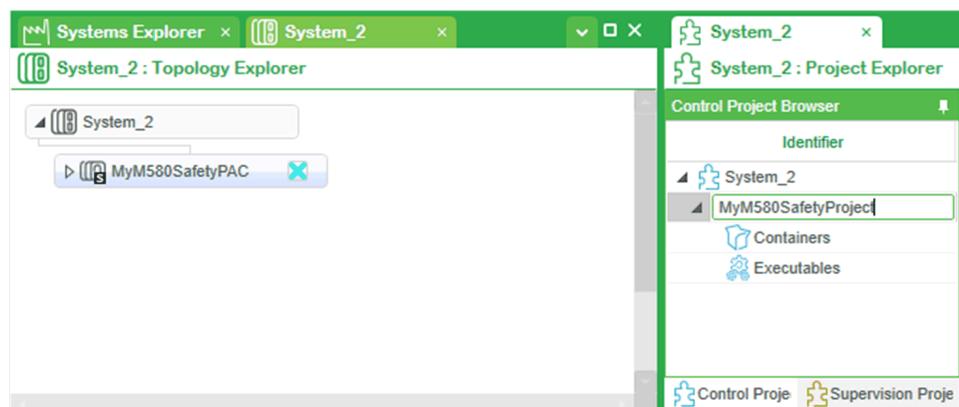
Importing the M580 Safety Controller

After the import of *MyM580SafetyPAC*, a blank safety Control project is created. The project is associated to the controller but has the default identifier.



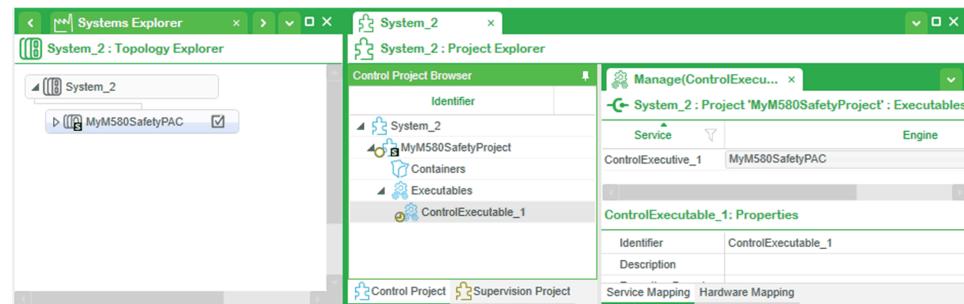
Renaming the Associated M580 Safety Control Project

Rename the associated safety Control project so that it has the same identifier as the project that you are going to import.



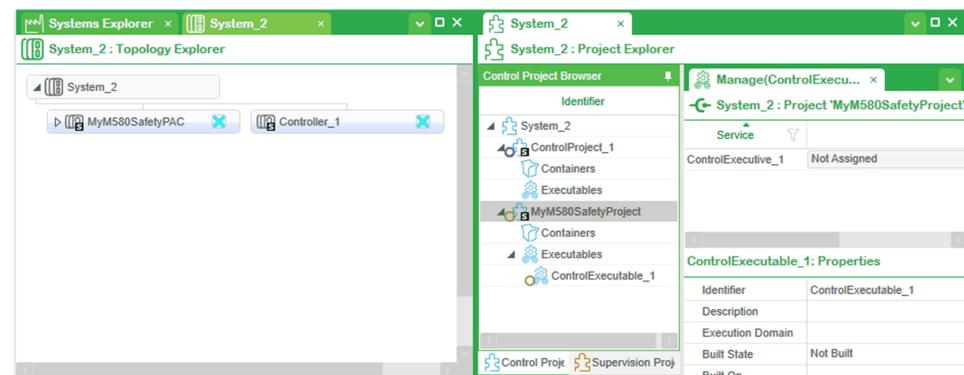
Importing the Associated M580 Safety Control Project

When you import *MyM580SafetyProject* and select its executable and service mapping, it is merged with the safety Control project that you have renamed. No new associated safety controller is created. Instead, the service mapping with *MyM580SafetyPAC* is recreated.



Importing the Associated Project Without Renaming

The following figure shows the result after importing *MyM580SafetyProject* if you do not rename *ControlProject_1*. Two pairs of controller/Control project are created and *MyM580SafetyPAC* is not associated nor mapped to *MyM580SafetyProject*.



M580 Safety Systems in the Assignment Stage

This topic describes the aspects of the Control project assignment that are specific to using M580 safety systems. For general information, refer to the topic describing the assignment stage.

Creating and Viewing Sections

You can only create FBD sections of the process program by using the **Assignment Editor**.

None of the sections of the safety program that you create in refinement appear in the **Assignment Editor**.

Assigning Facets of Instances

You can only assign facets of instances to FBD sections of the process program by using the **Assignment Editor**.

M580 Safety Systems in the Generation Stage

This topic describes the aspects of the Control project generation that are specific to using M580 safety systems. For general information, refer to the topic describing the generation stage.

Generating M580 Safety Control Projects

The generation process only generates logic and data in the process program of an M580 safety Control project in the same way as it is done for other Control projects.

NOTE: While the safety Control project is being generated, it is locked. This also locks the associated safety controller.

Regenerating M580 Safety Control Projects

Regenerating a safety Control project does not discard changes that you have performed in the safety program.

The process program is impacted in the same way as the logic and data of non-safety Control projects.

Impact of Generation on the Associated M580 Safety Controller

When you generate a safety Control project and you have not yet configured the associated M580 safety controller, the safety controller entity changes to a configured state in the **Topology Explorer**. That is, you can expand the node and see the default controller configuration.

M580 Safety Systems in the Refinement Stage

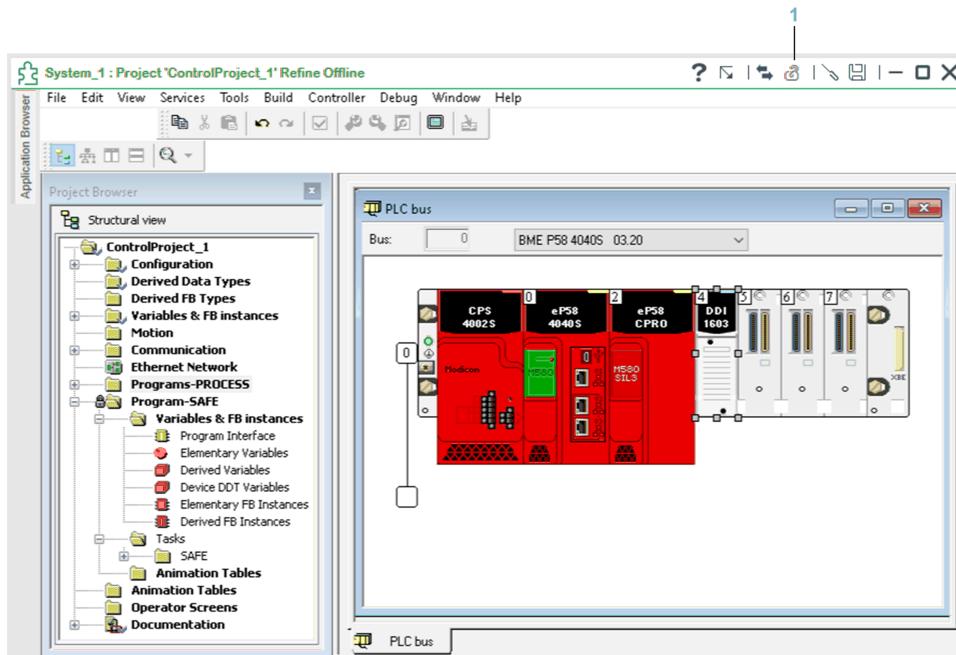
This topic describes the aspects of the offline Control project refinement that are specific to using M580 safety systems. For general information, refer to the topic describing the refinement stage.

Refining M580 Safety Control Projects Offline

Use the **Refine** command in the **Project Explorer** to open the project in the Control Participant. This lets you refine both the safety and the process program.

If a safety password, page 770 is set for the controller, before you can modify the safety program, you need to deactivate the safe area password protection in the toolbar of the Participant window and enter the safety password. Safe area password protection is activated automatically when you close the Participant window.

The following figure shows an example of the Control Participant window that opens when you refine the safety Control project while safety password protection is enabled.



1 Button to deactivate/activate the safe area password protection.

NOTE: When you refine a safety Control project, you can view and access the configuration of the associated controller. However, to modify the configuration, follow the normal system engineering life cycle. That is, use the **Configure** command in the **Topology Explorer**.

NOTE: While the Control Participant refinement window is open for the M580 safety project, the associated safety controller is locked.

Impact of Refinement on the Associated M580 Safety Controller

When you refine a safety Control project and you have not yet configured the associated M580 safety controller, the safety controller entity changes to a configured state in the **Topology Explorer**. That is, you can expand the node and see the default controller configuration.

Control Participant Project Settings

The *Usage of Process Namespace* project setting of the Control Participant is not supported and it must remain disabled. Otherwise, you cannot save changes that you have made during Control project refinement.

This has an impact on how the operator screen can reach variables of the Process and Global namespaces because you cannot use the PROCESS prefix anymore in case two variables are declared with the same name in these namespaces.

For details, refer to the topic describing how to display data on operator screens (*Animation Tables and Operator Screens*) in the *Modicon M580 Safety Manual*.

Performing Consistency Checks

When you perform a consistency check, page 449, only the logic of the process program is checked.

Device DDT Variable Wizard

In the toolbar of the Control Participant window, the button that opens the device DDT variable wizard, page 445 is not available because any device DDT variables

that are created for the hardware that is configured in the associated safety controller are already present and can be used.

Managing Network Variables

The **Manage Network Variables** dialog box, page 459 does not let you manage variables of the safety program.

M580 Safety Systems in the Configuration Stage

This topic describes the aspects of the configuration that are specific to using M580 safety systems. For general information, refer to the topic describing the configuration stage, page 499.

Creating and Configuring M580 Safety Controllers

Integrated Architectures

For a list of M580 safety CPU modules that can be configured, refer to the topic describing integrated architectures (see EcoStruxure™ Process Expert, Control Participant Services, User Guide).

For details on the supported modules, refer to the *Modicon M580 Safety Manual*.

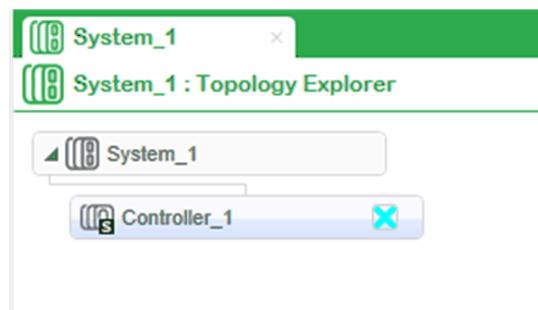
Creating M580 Safety Controller Entities

To create an M580 safety controller, right-click the system root folder or a topology folder in the **Topology Explorer** and select **M580 Safety**.

Creating a safety controller also creates a blank M580 safety Control project in the **Project Explorer**, which is associated to the controller.

Conversely, if you have created an M580 safety Control project, a default safety controller entity has been created in the topology.

The following figures shows an M580 safety controller entity as it appears in the **Topology Explorer**.



Deleting M580 Safety Controller Entities

Deleting a safety controller also deletes the associated M580 safety Control project in the **Project Explorer**, and vice versa.

Configuring an M580 Safety Controller

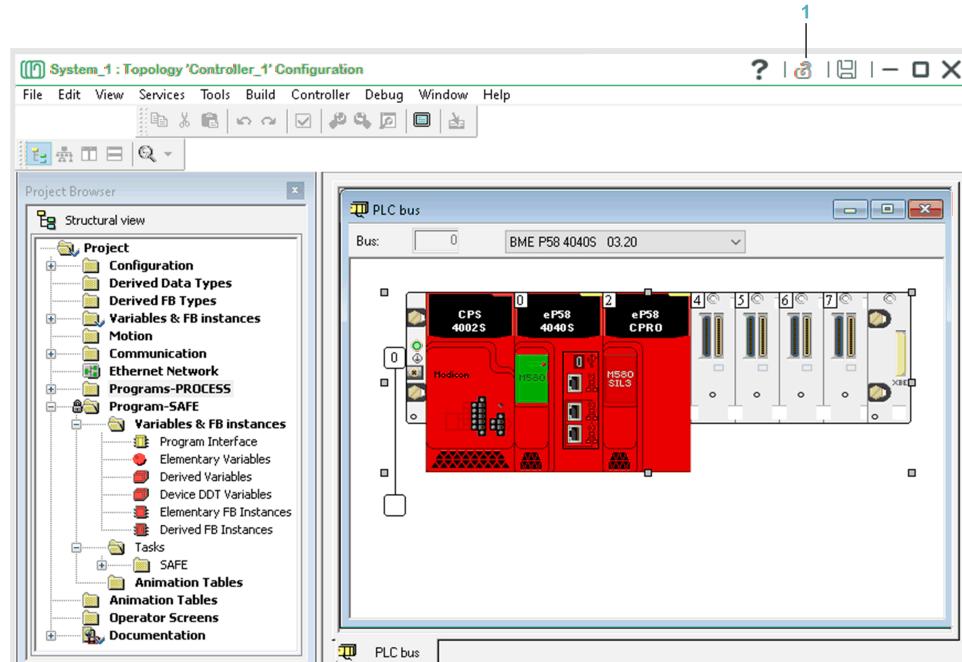
Use the **Configure** command to open the Control Participant. This lets you add, change, and configure the hardware of the safety controller:

- The safety CPU and coprocessor modules
- Safety power supply modules

- Safety I/O modules
- Non-safety modules
- Racks

If the safety password, page 770 is enabled for the controller, before you can configure safety modules, you need to disable the safety protection in the toolbar of the Participant window and enter the safety password. Safety protection is enabled automatically when you close the Participant window.

The following figure shows an example of the Control Participant window that opens when you configure an M580 safety controller while safety password protection is enabled.



1 Button to disable safety protection.

NOTE: When you configure an M580 safety controller, you can view and access both the safe and the process programs of the associated safety Control project. However, to create or modify any data and logic, use the **Project Explorer**.

NOTE: While the Control Participant configuration window is open for the M580 safety controller, the associated safety Control project is locked.

Analyzing Configuration Changes

By default, when you save the configuration of an M580 safety controller, the Control Participant does not perform an analysis. To have the configuration analyzed, enable *Analyze Project* in the settings, page 260 of the M580 safety Control project. This also analyzes the program areas of the Control project.

Managing the Safety Password

Overview

The safe area password protection function of the Control Participant for M580 safety controllers is managed from the **Topology Explorer** by using the **Safety** password function instead. The corresponding property is therefore disabled inside the Control Participant.

The **Safety** password is not available for a station node running the controller simulator.

For more information about the safe area password function, refer to the topic describing *Application Security Management* in the *Modicon M580 Safety Manual*.

Scope of the Safety Password

The following table summarizes for which actions you need to enter the safety password when safety password protection is enabled.

Action	Description
Configuration of the M580 safety controller.	Unlocking safety protection to be able to configure safety modules.
Offline refinement of the M580 safety Control project.	Unlocking safety protection to be able to modify safety data and logic.
Online refinement of the M580 safety Control project.	Switching the controller to maintenance mode to be able to modify safety data and logic online. This applies also to the controller simulator.
All deployment operations on the M580 safety controller that include the safety program.	Entering the safety password is required to be able to proceed with the deployment only if the safety program has been modified. This applies also to the controller simulator. NOTE: A first deployment may require entering the safety password even when the safety program has not been modified.

Safety Password Requirements

Your password must contain between 8 and 16 alphanumeric characters to be accepted.

It can contain lower and upper case as well as special characters.

NOTE: Password strength is increased when it contains a mix of upper and lower case, alphabetical, numeric, and special characters.

Enabling and Disabling the Safety Password

The safety password is enabled by default.

To disable it, use the **Safety** property, page 507 of the M580 safety controller.

Setting the Safety Password

To set the safety password, use the **Manage Password** command, page 507 in the context menu of the M580 safety controller.

This action sets the build state of the executable of the associated safety Control project to **Out Of Date** if it was built.

Changing the Safety Password

To change the safety password, use the **Manage Password** command, page 507 in the context menu of the M580 safety controller.

This action sets the build state of the executable of the associated safety Control project to **Out Of Date** if it was built.

Clearing the Safety Password

To clear the safety password, use the **Clear Password** command, page 507 in the context menu of the M580 safety controller.

This action sets the build state of the executable of the associated safety Control project to **Out Of Date** if it was built.

Resetting a Forgotten Safety Password

NOTE: If you have also forgotten the **Controller** password, proceed with the reset, page 658 of that password before resetting the **Safety** password.

To reset the safety password of an M580 safety controller, proceed as follows.

Step	Action
1	<p>Right-click the topological entity and select Forgot Password > Safety.</p> <p>Result: The Forgot Password dialog box opens and an authentication code is displayed.</p> <p>NOTE: If you close the Forgot Password dialog box and open it again, the same authentication code is displayed.</p>
2	<p>Contact Schneider Electric support, page 104, provide the authentication code, and specify that you want to reset the safety password..</p> <p>Result: You receive a temporary password.</p> <p>NOTE: The temporary password is valid as long as you do not modify the configuration or the associated Control project after requesting it.</p>
3	<p>Right-click the safety controller and select Manage Password > Safety.</p> <p>Result: The Manage Password dialog box opens.</p>
4	<p>Enter the temporary password in the Current Password text box, the new password, and click OK.</p> <p>Result: The new safety password is set.</p>
5	Build the executable of the associated safety Control project if its build state is Out Of Date .
6	If you had already deployed the project to the safety controller, apply the new safety password to the deployed Control project by using the Deploy Changes/Undo Online Changes command.

Exporting and Importing M580 Safety Controllers

This topic describes the aspects of the topology export and import that are specific to M580 safety controllers. For general information about topology export and import operations, refer to the topic describing how to manage the topology, page 548 of a system.

Exporting M580 Safety Controllers

The **Export** command of the **Topology Explorer** lets you export one or more M580 safety controllers and create a non-editable export file (.sbk).

You can also export at the same time other topological entities of the system.

The associated safety Control project is not exported.

Importing M580 Safety Controllers

The **Import** command of the **Topology Explorer** lets you import the M580 safety controllers that are contained in a topology export file (.sbk).

The Global Templates library of the target system must contain the necessary topological safety templates (Foundation library (see EcoStruxure™ Process Expert, Installation and Configuration Guide)) before importing. These templates may be missing if you have migrated a database of an earlier software version. In that case, you will not be able to save your changes when you configure the safety controller.

The table describes the result when you import an M580 safety controller in different scenarios.

Target system	Action selected in the Import window to solve the conflict	Result after import in Topology Explorer	Impact on projects after import
An M580 safety controller with the same identifier does not exists.	N/A	The safety controller is created with the same identifier and configuration. The exported project settings are also retained.	A blank M580 safety Control project is created, which is associated to the safety controller.
An M580 <i>non-safety</i> controller with the same identifier already exists.	Create NOTE: Update is not available.	The safety controller is created with a new identifier but the same configuration. The exported project settings are also retained.	A blank M580 safety Control project is created, which is associated to the safety controller.
An M580 safety controller with the same identifier already exists.	Create	The safety controller is created with a new identifier but the same configuration. The exported project settings are also retained.	A blank M580 safety Control project is created, which is associated to the safety controller.
	Update	The configuration of the existing safety controller is replaced with that of the imported safety controller. The project settings are also replaced.	No impact.

NOTE: During import of a safety controller, the associated M580 Control project is locked.

Safety Password Management When Importing

When you import one or more M580 safety controllers, before the import operation starts, the **Verify Password** window may open prompting you to enter a safety password for each one.

The import operation starts only once all required safety passwords have been entered.

The following table describes which safety password you need to enter when prompted during the import of an M580 safety controller.

Target system	Safety password required	Impact on safety password after import
An M580 safety controller with the same identifier does not exist.	The safety password of the safety controller that you are importing (if set).	The safety password settings are the same as that of the imported safety controller.
An M580 safety controller with the same identifier already exists.	The safety password of the existing safety controller in the target system (if set).	The safety password and password settings are the same as that of the existing safety controller in the target system. For example, if the existing safety controller has no safety password but the safety controller that you are importing has a safety password, after the import, no safety password is set.

M580 Safety Systems in the Mapping Stage

This topic describes the aspects of the Control project mapping that are specific to using M580 safety systems. For general information, refer to the topic describing the mapping stage.

Control Executable Service Mapping

You can create a service mapping only with the M580 safety controller that is associated to the safety Control project.

Only one executable of a safety Control project can be mapped to this controller. If you have created additional executables, you can only map them to station nodes on which a Control service has been created.

Communication Mapping

The table describes the type of peer to peer communication that you can implement by using the **Communication Mapping Editor**.

Controller 1	Controller 2	Peer to peer communication
M580 safety platform	M580, M340, or Quantum platform	Possible by using process variables only.
M580 safety platform	M580 safety platform	Possible by using process variables only. To use variables of the safety program, refine the safety Control projects.

Also, communication between an M580 safety CPU and CIP safety devices cannot be implemented by using the **Communication Mapping Editor**. You need to refine the safety Control project.

Hardware Mapping

The I/O safety modules of the M580 safety controller do not appear in the **Hardware Mapping Editor** and therefore cannot be mapped to interfaces of Control facets.

Mapping of communication interfaces of the CPU and other non-safety communication modules of the safety controller can be performed in the same way as with controllers of the other platforms.

M580 Safety Systems in the Build Stage

This topic describes the aspects of building executables of Control projects in the **Project Explorer** that are specific to M580 safety systems. For general information, refer to the topic describing the build stage.

Building M580 Safety Control projects

Using a build command in the context menu of the executable of a safety Control project locks the associated M580 safety controller.

NOTE: Once you have built the executable of a safety Control project by using the **Build All** command, you cannot modify the service mapping anymore nor delete the executable if it is mapped to the M580 safety controller.

Safe Signature

Each time you build the executable of a safety Control project, if you have changed the safety configuration, safety program, or certain project settings, the software generates a SourceSafeSignature.

For a list of actions that generate a SourceSafeSignature change, refer to the *Modicon M580 Safety Manual*.

The software uses the SourceSafeSignature to inform you of a change when you use a deployment command.

However, the safe signature cannot be viewed at the platform level. To copy it, open the Control project by using the **Open Built Project** command in the context menu of its Control executable.

NOTE: The software sets the **Safety Signature management** setting of the Control Participant to **Automatic** when you build an M580 safety Control project (**Tools > Project Settings > General > Build Settings**).

M580 Safety Systems in the Deployment Stage

This topic describes the aspects of the Control project deployment that are specific to using M580 safety systems. For general information, refer to the topic describing the deployment stage.

Deploying to M580 Safety Controllers

Use the **Deploy Built Project** and **Re-Deploy Last Project** commands in the **Topology Explorer** to deploy the Control project to the M580 safety controller. This lets you deploy both the safety and the process program.

Deploying Changes to M580 Safety Controllers

Use the **Deploy Changes / Undo Online Changes** command in the **Topology Explorer** to deploy Control project changes to the M580 safety controller. This lets you deploy changes, page 682 to the safety and/or the process program.

However, the changes that appear in the **Safe Program** section can only be deployed all at once or not at all.

When you deploy changes to the process program that impact the safety program (for example, a process variable that is used as program interface to exchange data with the safe program), a dialog box opens and allows you to confirm the change or modify the selection.

Safe Signature Changes

Before proceeding with a deployment, re-deployment, or deployment of changes operation that includes the safety program, if the software detects that the SourceSafeSignature of the application that you are deploying is different from that of the application in the controller, it opens a dialog box to inform you.

You have the choice to acknowledge and proceed with the deployment or cancel the operation.

The SourceSafeSignature itself is not displayed nor can it be copied.

Deploying With Safety Password Protection Enabled

For deployment, re-deployment, and deployment of changes operations that include the safety program, if the safety password, page 770 is enabled for the controller and the safety program has been modified, before you can proceed with the deployment, you need to enter the safety password when prompted.

M580 Systems in the Execution Stage

This topic describes the aspects of the project online refinement and update that are specific to using M580 safety systems. For general information, refer to the topic describing the execution stage.

Refining M580 Safety Control Projects Online

Use the **Refine Online** command in the **Topology Explorer** to modify online both the process program and configuration and safety program and configuration of the deployed Control project.

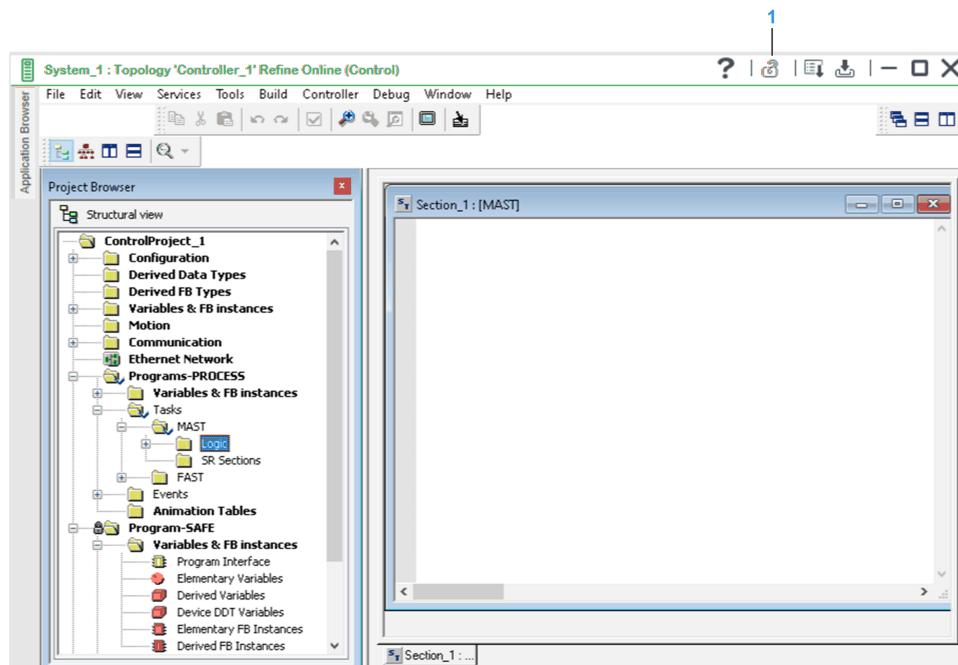
The controller opens in safety mode.

If a safety password, page 770 is set for the controller, before you can modify the safety program or configuration online, you need to switch to maintenance mode by using the button in the toolbar of the Participant window. This requires entering the safety password. Otherwise, if the safety password is disabled, switching to maintenance mode does not require a password.

Safety mode is enabled automatically when you close the Participant window.

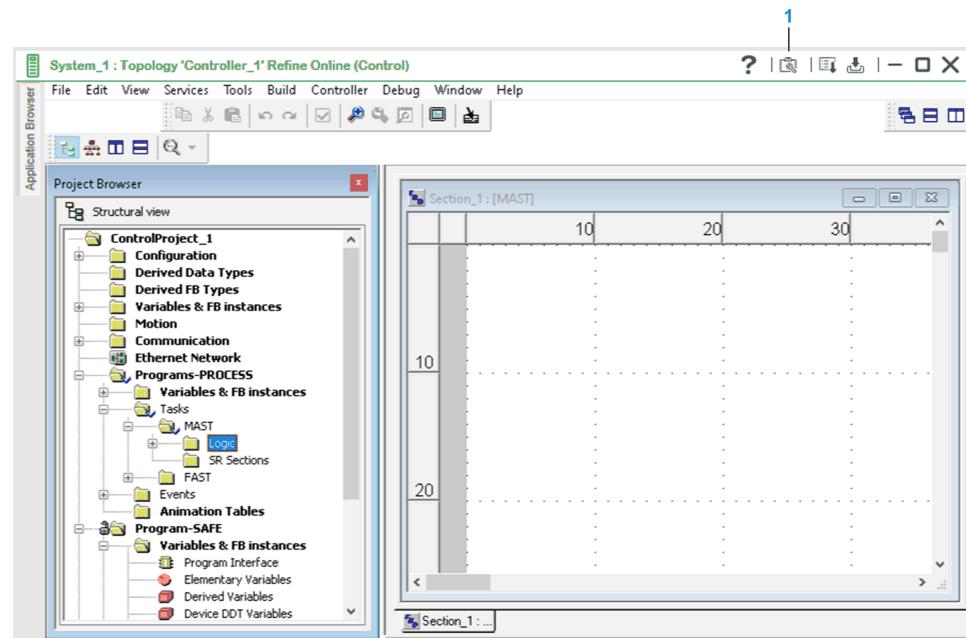
The same applies also when you refine online by using the controller simulator.

The following figure shows an example of the Control Participant window that opens when you refine a safety Control project online while safety password protection is enabled.



1 Button to switch between safety mode and maintenance mode.

The following figure shows an example of the Control Participant window that opens when safety password protection is disabled.



1 Button to switch between safety mode and maintenance mode.

Updating Logical M580 Safety Control Projects

Use the **Update Control Project** command in the **Topology Explorer** to update the logical M580 safety Control project and the configuration with changes that you have made online. This lets you update, page 748 both the process and the safety areas.

However, the changes that appear in the **Safe Program** section can only be updated all at once or not at all.

If the safety password, page 770 is enabled for the controller, before you can update the safety program, you need to enter the safety password.

Using Supervision with M580 Safety Systems

Overview

Because instances of the application cannot be used to create data and logic of the safety program of M580 safety Control projects, you have to create your own Supervision instances to monitor safety Control data.

For general information on working with Supervision instances, refer to the various Supervision topics in the description of the system engineering life cycle.

Using Supervision Templates

Step	Description
1	<p>You can instantiate the following Supervision data facet templates of the Foundation library for each variable of an M580 safety Control project that you want to monitor from the Supervision runtime.</p> <ul style="list-style-type: none"> • <code>\$VariableTag_CD</code> • <code>\$AdvancedAlarm_CD</code> • <code>\$DigitalAlarm_CD</code> • <code>\$TrendTag_CD</code> <p>NOTE: Select Facets as filter in the Browser pane.</p>
2	<p>Configure instances of <code>\$VariableTag_CD</code> as follows:</p> <ul style="list-style-type: none"> • <i>Addr</i> parameter (<i>Configuration</i> category): Name of the safety variable in the Control project preceded by the <code>safe.</code> prefix. For example, if the variable name is <i>MyVar1</i>, enter <code>safe.MyVar1</code>. • <i>Type</i> parameter (<i>Configuration</i> category): Data type of the variable. For example, <code>UINT</code>.
3	If you are using trend tags, open the Asset Workspace Editor and link the trend tag instance to the variable tag instance so that the instance status becomes valid.
4	Follow the system engineering life cycle to complete the other stages for the Supervision project.

Managing Communication

What's in This Chapter

Communication Through Modbus TCP Ethernet Implicit Messaging (I/O Scanning)	780
Communication by Using Modbus TCP Ethernet Explicit Messaging	788
Communication by Using EtherNet/IP Explicit Messaging	794
Communication Through Modbus Explicit Messaging	798
Monitoring Modbus Device Data Directly from Supervision	830
Communication with Devices by Using a PROFIBUS Master Module	835

Overview

This chapter contains information describing how to manage communication between devices and controllers depending on the communication protocol that you want to use. It contains examples that use templates of the General Purpose library Classic.

WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines.¹
- Test each implementation of this library for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

NOTE: The software does not verify the status of the communication between controllers and I/O devices.

Communication Through Modbus TCP Ethernet Implicit Messaging (I/O Scanning)

Overview

This section contains specific procedures to help you configure communication through Modbus TCP Ethernet implicit messaging by using the I/O scanner function. It covers communication between a controller and entities that are configured in the topology of the system, and between several controllers (peer to peer communication). It contains examples that use templates of the General Purpose library Classic.

NOTE: For the engineering of a system with the software, it is not relevant how Modbus TCP devices are physically connected to the real Ethernet network. As long as they are connected to the same logical Ethernet network with a valid IP address, you can proceed with the implementation of communication in the software. However, in your physical architecture, you need to confirm that connections are done properly so that devices are indeed reachable.

For example, in an M580 architecture, whether an STB island is connected directly to the service port of the CPU module, to the port of a BMENOC0301.2 module, or to the service port of a BMXCRA31210 module does not matter. In the software, you configure only the DIO scanner service of the CPU or NOC module to exchange data with the STB island on the same Ethernet network.

Communicating with Devices Through Modbus TCP Implicit Messaging (I/O Scanning)

Overview

The software supports communication through Modbus TCP Ethernet implicit messaging between common devices and supported controller platforms by using the I/O scanner function.

In addition to configuring the controller as a client and enabling its I/O scanner, you need to instantiate the topological template of the Modbus TCP device.

The topological device instance allows you to assign an IP address to the device and connect it to the Ethernet network to which the controller is connected.

The communication channel between the controller and the device is managed through the communication mapping, page 582.

The build process creates the communication channel in the built Control Participant project.

For a list of available device templates, refer to the Device/Communication Matrix (see EcoStruxure™ Process Expert, General Purpose Library Classic Communication Control Services Reference Manual).

NOTE: Refer to the TVDA document *Hot Standby Quantum_EIO* for recommendations when using a NOC communication module to implement communication between Modbus TCP devices and Hot Standby controllers of the Quantum platform.

Using Device Application Templates

The device-specific application template of a Modbus TCP device (for example, \$TsysTE) allows you to edit the Control and Supervision parameters of the device and proceed with the hardware mapping to the controller I/O modules.

It is not required for managing communication with the device using I/O scanning.

Generic Device Templates

If no specific topological template is available for the device that you are using, use one of the generic topological device templates, page 531 and its application counterpart.

The generic application template lets you create an input and an output variable of required length and data type.

The table indicates which device template to use depending on the module that is used to communicate with the device.

Module used to communicate with device	Topological device template to use	Corresponding application device template to use
CPU module of an M580 controller	\$EGenericDeviceHW	\$GenNOCDevice
BMENOC***** communication module		Some manual configuration is required (see EcoStruxure™ Process Expert, General Purpose Library Classic Device Templates Reference Manual).
CPU module of a Quantum controller	\$EGenericDeviceHW	\$GenericDevice
140NOE***** communication module		

Implementing Communication

To implement communication between a controller and a device through Modbus TCP by using I/O scanning, complete these steps.

Step	Description	Stage
1	Create a Control project in the system.	Project Definition
2	Create a Control executable in the project.	
3	Create and configure a controller, page 507 with the I/O scanner enabled and its communication module. Configure the client memory start and length parameters, page 512 of the communication module.	Configuration
4	Create and configure, page 531 the topological device entity (configure IP address and other applicable properties). NOTE: If the device communicates with a communication module that validates IP addresses based on the subnet mask, verify that the IP addresses of the device and the module belong to the same subnet.	
5	Create an Ethernet network, page 538.	
6	Connect the communication module of the controller, page 540 to the Ethernet network.	
7	Connect the I/O device, page 545 to the same Ethernet network.	
8	Instantiate the application template that corresponds to the topological device entity and configure its properties.	
9	Map the Control executable of the project to the controller entity, page 580.	
10	Define the communication channel, page 582 between the controller and the device.	

To complete the other steps to implement the system that contains the controller and the device, follow the system engineering life cycle described in this manual.

NOTE: Refer to the topic describing device communication technologies (see EcoStruxure™ Process Expert, General Purpose Library Classic Device Control Services Reference Manual) for configuration and setup details.

Implementing Communication by Using Generic Device Templates

To implement communication through Modbus TCP by using I/O scanning, between a controller and a device for which no specific topological template exists, complete these steps.

Step	Description	Stage
1	Create a Control project in the system.	Project Definition
2	Create a Control executable in the project.	
3	Create and configure a controller, page 507 with the I/O scanner enabled and its communication module. Configure the client memory start and length parameters, page 512 of the communication module.	Configuration
4	Create and configure the generic topological device entity, page 531, assign an IP address to it, and enter values for the read/write parameters, page 532. NOTE: If the device communicates with a communication module that validates IP addresses based on the subnet mask, verify that the IP addresses of the device and the module belong to the same subnet.	
5	Create an Ethernet network, page 538.	
6	Connect the communication module of the controller, page 540 to the Ethernet network.	
7	Connect the I/O device, page 545 to the same Ethernet network.	
8	Instantiate the generic application device, page 781 template and configure its properties.	

Step	Description	Stage
9	Map the Control executable of the project to the controller entity, page 580.	Mapping (service)
10	Define the communication channel, page 582 between the controller and the device.	Mapping (communication)

To complete the other steps to implement the system that contains the controller and the device, follow the system engineering life cycle described in this manual.

Communicating with STB Islands Through Modbus TCP Implicit Messaging

Overview

The software supports communication through Modbus TCP Ethernet implicit messaging between STB islands and supported controller platforms by using the I/O scanner function.

The controller acts as client, and its I/O scanner is enabled.

The STB island acts as server, and both are connected to the same Ethernet network, on which they are identified with their respective IP address.

The communication channel between the controller and the STB island is defined through the communication mapping, page 582.

The build process creates the communication channel in the built Control Participant project.

Implementing Communication

To implement communication between a controller and an STB island through Modbus TCP (I/O scanning), complete these steps.

Step	Description	Stage
1	Create a Control project in the system.	Project Definition
2	Create a Control executable in the project.	
3	Create and configure a controller, page 507 with the I/O scanner enabled and its communication module. Configure the client memory start and length parameters, page 512 of the communication module.	Configuration
4	Create and configure an STB island, page 523 entity and assign an IP address to its NIM. NOTE: If the NIM communicates with a communication module that validates IP addresses based on the subnet mask, verify that the IP addresses of the NIM and the module belong to the same subnet.	
5	Create an Ethernet network, page 538.	
6	Connect the communication module of the controller, page 542 to the Ethernet network.	
7	Connect the communication module of the STB island, page 545 to the same Ethernet network.	
8	Map the Control executable of the project to the controller entity, page 580.	Mapping (service)
9	Define the communication channel, page 582 between the controller and the STB island.	Mapping (communication)

To complete the other steps to implement the system that contains the controller and the STB island, follow the system engineering life cycle described in this manual.

Communicating with PROFIBUS Remote Masters Through Modbus TCP Implicit Messaging

Overview

The software supports communication through Modbus TCP implicit messaging between PROFIBUS remote masters (PRMs) and supported controller platforms by using the I/O scanner function.

The topological PRM entity allows you to configure decentralized devices (DPs) and assign an IP address to the PRM to connect it to the Ethernet network to which the controller is connected.

In addition to configuring the controller as a client with the I/O scanner enabled, you need to instantiate:

- A PRM management application template.
- A device application template that corresponds to the DP that is configured in the PRM entity.

The communication channel between the controller and the PRM is managed through the communication mapping, page 582.

The build process creates the communication channel in the built Control Participant project.

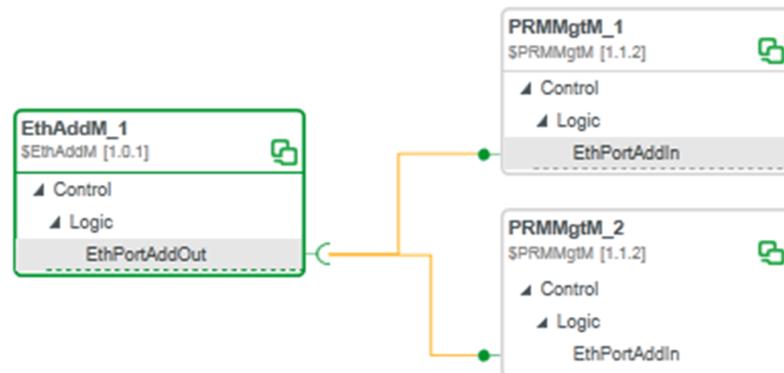
NOTE: The device application templates of the configured DPs allow you to edit the Control and Supervision parameters of the devices and proceed with the hardware mapping to the controller I/O modules. They are not required to manage communication with the DPs by using I/O scanning.

Using several Communication Port Instances

To use one topological communication port with several PRM management instances (application), you need to instantiate the following intermediate templates (fan-out type application templates). Each instance of such template accepts links to n PRM management instances.

PRM management template	Intermediate template
\$PRMMgtM	\$EthAddM
\$PRMMgtQ	\$EthAddQ

The following figure shows an example of how to link two instances of the PRM management template to the instance of an intermediate template. During hardware mapping, the interface of the intermediate template instance is mapped to the interface of the Ethernet port of an M340 or M580 controller.



Implementing Communication

To implement communication between a controller, a PRM and its DP through Modbus TCP by using I/O scanning, complete these steps.

Step	Description	Stage
1	Create and configure a controller, page 507 with the I/O scanner enabled and its communication module. Configure the client memory start and length parameters, page 512 of the communication module.	Configuration
2	Create and configure, page 527 the PRM entity and its DPs (configure IP address and other applicable properties). NOTE: If the PRM communicates with a communication module that validates IP addresses based on the subnet mask, verify that the IP addresses of the PRM and the module belong to the same subnet.	
3	Create an Ethernet network, page 538.	
4	Connect the communication module of the controller, page 540 to the Ethernet network.	
5	Connect the PRM entity, page 546 to the same Ethernet network.	
6	In the Application Explorer , instantiate the following templates: <ul style="list-style-type: none"> The PRM management template. One for each PRM entity that is configured: <ul style="list-style-type: none"> \$PRMMgtQ for a controller of the Quantum platform. \$PRMMgtM for a controller of the M340 or M580 platform. One application template for each DP that you have configured (for example, \$ATV71PB for an Altivar 71 DP). NOTE: If no specific template is available for a given DP, you can instantiate the generic application template \$GenericDevice.	Instantiation
7	If you are using the \$PRMMgtQ PRM management template, in the Instance Editor , configure the StartMemoryHeap parameter. Refer to the help of the template for details.	
8	If you are using several PRM management instances, instantiate the corresponding intermediate template, page 785 and link its instance to each PRM management instance by using the Asset Workspace Editor . NOTE: A link between the DP instance and the PRM management instance is not required.	
9	Create a Control Participant project in the system.	Project Definition
10	Assign the necessary facets to the Control Participant project.	
11	Create a Control executable in the project.	Mapping (service)
12	Map the Control executable of the project to the controller entity, page 580.	
13	Define the communication channel, page 582 between the controller and the device.	Mapping (communication)
14	Map to their respective topological counterpart (HO) by dragging from right to left: <ul style="list-style-type: none"> The EMPortChM or EMPortChQ interface. The PRMCTRLData interface. The PRMEthernetAddress interface. The interface of the device application facet to the interface of the topological DP device. 	Mapping (hardware)

To complete the other steps to engineer the system that contains the controller and the device, follow the system engineering life cycle described in this manual.

NOTE: Refer to the help of the various templates for configuration and setup details.

Peer to Peer Communication Through Modbus TCP Ethernet Implicit Messaging

Overview

Peer to peer communication allows you to share data between two or more Control Participant projects by using the I/O scanner function for any pair of controllers that are configured in the topology of the system.

Network variables are used to exchange data between the Participant projects to which the controllers are mapped.

Client/server communication channels and network variable mappings are managed through communication mapping, page 582.

The build process creates the communication channels and allocates the mapped variables in the respective built Control Participant projects.

For more information about ways to create variables to share data, refer to the topic documenting the creation of network variables, page 452.

Implementing Peer to Peer Communication

To implement peer to peer communication through Modbus TCP implicit messaging by using I/O scanning, follow these steps.

Step	Description	Stage
1	Create two Control projects in the same system.	Project definition
2	Create a Control executable in each project.	
3	Create the network variables that will be exchanged, page 452 between the projects.	Refinement
4	Create and configure two controllers, page 507 with the necessary communication and I/O modules. In the controller acting as client, enable the I/O scanner function.	
5	In the controller acting as client, configure the client memory start and length parameters, page 512 of the communication module.	Configuration
6	In the controller acting as server, configure the server memory start and length parameters, page 512 of the communication module. NOTE: If the server communicates with a communication module that validates IP addresses based on the subnet mask client-side, verify that the IP addresses of the client/server modules belong to the same subnet.	
7	Create an Ethernet network, page 538.	
8	Connect the communication module, page 542 of each controller to this Ethernet network.	
9	Map the Control executable of the owner project to the controller, page 580 acting as server.	Mapping (service)
10	Map the Control executable of the consumer project to the controller, page 580 acting as client.	
11	Define the peer to peer communication channels, page 582 between the controllers.	Mapping (communication)
12	Map the network variables, page 594 to the peer to peer communication channels.	

To complete the other steps to implement the system, follow the system engineering life cycle described in this manual.

Communication by Using Modbus TCP Ethernet Explicit Messaging

Communication with Devices by Using Modbus TCP Ethernet Explicit Messaging

Overview

This topic contains examples that use templates of the General Purpose library Classic.

Supported Architectures

The software supports communication by using Modbus TCP Ethernet explicit messaging between common devices and supported controller platforms.

The process requires to instantiate the topological template of the Modbus TCP device, its application counterpart, and the appropriate application template of the communication port.

The communication port instance lets you configure the number of simultaneous requests that are sent per cycle. The configuration depends on the capacity of the communication module. For example, the various M580 CPU modules can send from 16 to 80 requests as client.

The topological device instance lets you assign an IP address to the device and connect it to the Ethernet network to which the controller is connected.

The IP address is exchanged with the application template of the device through the hardware mapping.

For a list of supported communication architectures, refer to the Device/Communication Matrix (see EcoStruxure™ Process Expert, General Purpose Library Classic Communication Control Services Reference Manual).

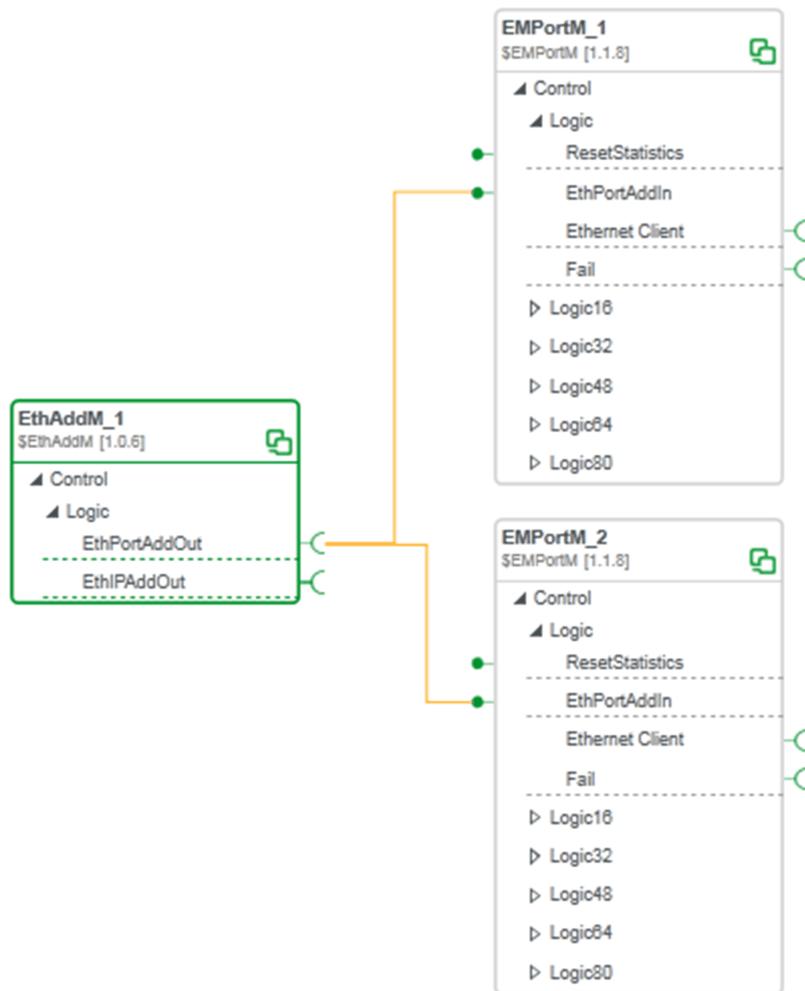
NOTE: If no device-specific topological template is available for the device that you are using, use a generic topological device template, page 531 and its application counterpart *\$EMClient*.

Using several Communication Port Instances

To use one topological communication port with several communication port instances (application), you need to instantiate the following intermediate templates (fan-out type application templates). Each instance of such template accepts links to *n* communication port instances.

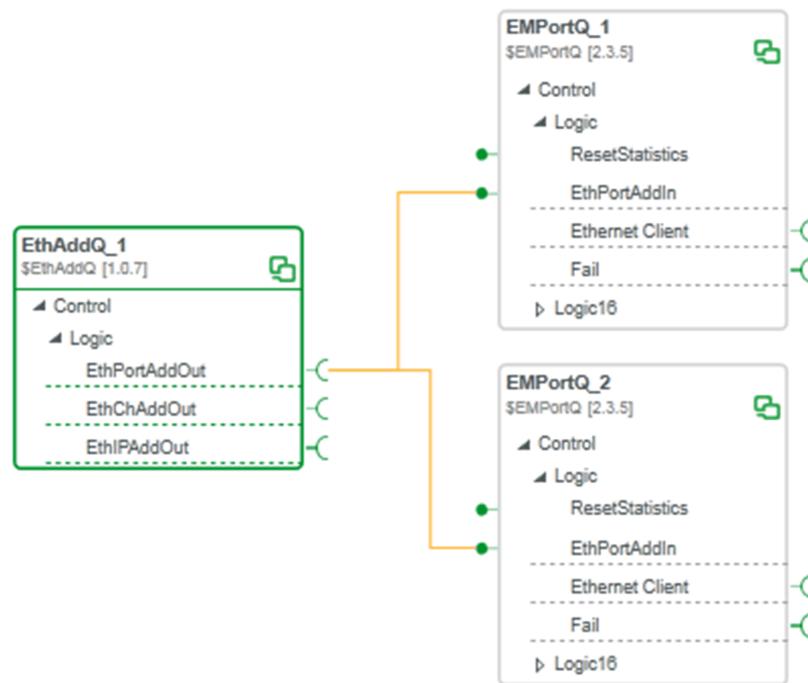
Communication port template	Intermediate template
<i>\$EMPortM</i>	<i>\$EthAddM</i>
<i>\$EMPortQ</i>	<i>\$EthAddQ</i>

The following figure shows an example of how to link two instances of the communication port template to the instance of an intermediate template. In the hardware mapping, the interface of the intermediate template instance is mapped to the interface of the Ethernet port of an M340 or M580 controller.



NOTE: The **EthIPAddOut** interface is not used for Modbus TCP explicit messaging.

The following figure shows an example of how to link two instances of the communication port template to the instance of an intermediate template. In the hardware mapping, the interface of the intermediate template instance is mapped to the interface of the Ethernet port of a Quantum controller.



NOTE: The **EthChAddOut** and **EthIPAddOut** interfaces are not used for Modbus TCP explicit messaging.

Implementing Communication

To implement communication between a controller and a device by using Modbus TCP Ethernet explicit messaging, follow these steps.

Step	Description	Stage
1	Instantiate the process template that corresponds to the piece of equipment that you want the device to communicate with (for example, \$Motor).	Instantiation
2	Instantiate the Modbus TCP explicit messaging (EM) application template that corresponds to device that you want to use (for example, \$TesysTEM). NOTE: If no device-specific application template exists, use the \$EMClient template. Create as many instances of it as you have requests. For example, for one read and one write request you need two instances.	
3	Instantiate the following communication port template: <ul style="list-style-type: none"> • \$EMPortQ for a controller of the Quantum platform. • \$EMPortM for a controller of the M340 or M580 platform. 	
4	Configure the communication port instance: <ul style="list-style-type: none"> • The Control element lets you select a profile, which corresponds to the maximum number of simultaneous requests that the communication module supports. • The corresponding LogicX service lets you optimize communication by reducing the number of simultaneous requests that are sent depending on your needs. X corresponds to the last two digits of the profile you select under Control. For example, to EMPortM16 corresponds Logic16. NOTE: If you select a profile that exceeds the capacity of the communication module, you cannot assign the communication port instance to the Control Participant project. Even if you reduce the number of requests under LogicX . For example, if you select EMPortM32 for a NOC communication module, which can send a maximum of 16 requests simultaneously.	

Step	Description	Stage
5	If you are using one topological communication port with several communication port instances, instantiate the corresponding intermediate template, page 789.	
6	Open the Asset Workspace Editor from the device instance and link to it: <ul style="list-style-type: none"> The port instance The process instance NOTE: In case of a generic device, the device instance is the <i>\$EMClient</i> template instance. NOTE: If you use an intermediate template instance, link it to the port instance.	
7	Configure a controller and its communication module.	Configuration
8	Create, page 531 the corresponding topological device entity and configure its IP address and other applicable properties (for example, <i>\$ETesysTHW</i>). NOTE: If no device-specific topological template exists, use a generic device template.	
9	Create an Ethernet network.	
10	Connect the communication module of the controller and the topological device entity to the same Ethernet network.	
11	Assign the Control facets of the process, device, and communication port instances to the Control Participant project so that the execution order, page 381 is as follows: <ol style="list-style-type: none"> The facet of the Modbus TCP explicit messaging device instance (for example, <i>TesysTEM_1</i>). The facet of the communication port instance (for example, <i>EMPortQ_1</i>). 	Assignment
12	Generate the project.	Generation
13	Map the executable of the Control Participant project to the engine of the controller, page 580.	Mapping (service)
14	Map the EMPortChM or EMPortChQ interface of the communication port instance (or of the intermediate template instance) to its topological counterpart.	Mapping (hardware)
15	Map the interface of the application instance of the device to the interface of the topological device instance.	

To complete the other steps to engineer the system that contains the controller and the device, follow the system engineering life cycle described in this manual.

NOTE: Refer to the following topics for configuration and setup details of:

- Devices (see EcoStruxure™ Process Expert, General Purpose Library Classic Device Control Services Reference Manual)
- Ports (see EcoStruxure™ Process Expert, General Purpose Library Classic Communication Control Services Reference Manual)

Examples

The following examples illustrate the configuration of Modbus TCP explicit messaging to connect to 20 devices from clients of two different platforms with different communication capacities.

Example 1	
Number of devices	20
Controller platform	Modicon Quantum
Communication module	140NOC77101
Communication port template	<i>\$EMPortQ</i>
Communication port instance configuration (<i>PortSelection</i>)	<i>EMPortQ16</i>

Example 1	
Requests sent simultaneously (<i>SimultaneousSends</i>)	10 in <i>Logic16</i>
Cycles to connect once to the number of devices	2 (2 x 10)

Example 2	
Number of devices	20
Controller platform	Modicon M580
Communication module	BMEP582020
Communication port template	\$EMPortM
Communication port instance configuration (<i>PortSelection</i>)	EMPortM32
Requests sent simultaneously (<i>SimultaneousSends</i>)	20 in <i>Logic32</i>
Cycles to connect once to the number of devices	1

Managing IP Address Changes

The topological templates of devices communicating by using Modbus TCP Ethernet explicit messaging use a variable to define the IP address of the device.

During deployment, the templates set up the initial value of this variable, which is connected to the pin of the DFB implementing the device.

When you change the IP address of a device by changing the hardware mapping of the facet modeling the device and deploy the change by using the **Deploy Changes** command, only the initial value, in this case the IP address, is changed online. The current value remains the same.

The IP address change becomes effective, either through a cold start of the controller or by replacing the current value of the variable in the animation table by the new initial value.

Communication by Using EtherNet/IP Explicit Messaging

Communication with Devices by Using EtherNet/IP Explicit Messaging

Overview

This section contains a procedure specifically aimed at managing communication between devices and M580 controllers by using EtherNet/IP explicit messaging. It contains examples that use templates of the General Purpose library Classic.

Supported Architectures

The software supports communication by using EtherNet/IP explicit messaging between common devices and CPU modules of M580 controllers or NOC communication modules.

The process requires to instantiate the topological template of the EtherNet/IP device, its application counterpart, and the appropriate application template of the communication port.

The communication port instance lets you configure the number of simultaneous requests that are sent per cycle. The configuration depends on the capacity of the communication module. For example, the various M580 CPU modules can send from 16 to 80 requests as client.

The topological device instance lets you assign an IP address to the device and connect it to the Ethernet network to which the controller is connected.

The IP address is exchanged with the application template of the device through the hardware mapping.

For a list of supported communication architectures, refer to the help of the Schneider Electric libraries for EcoStruxure Process Expert.

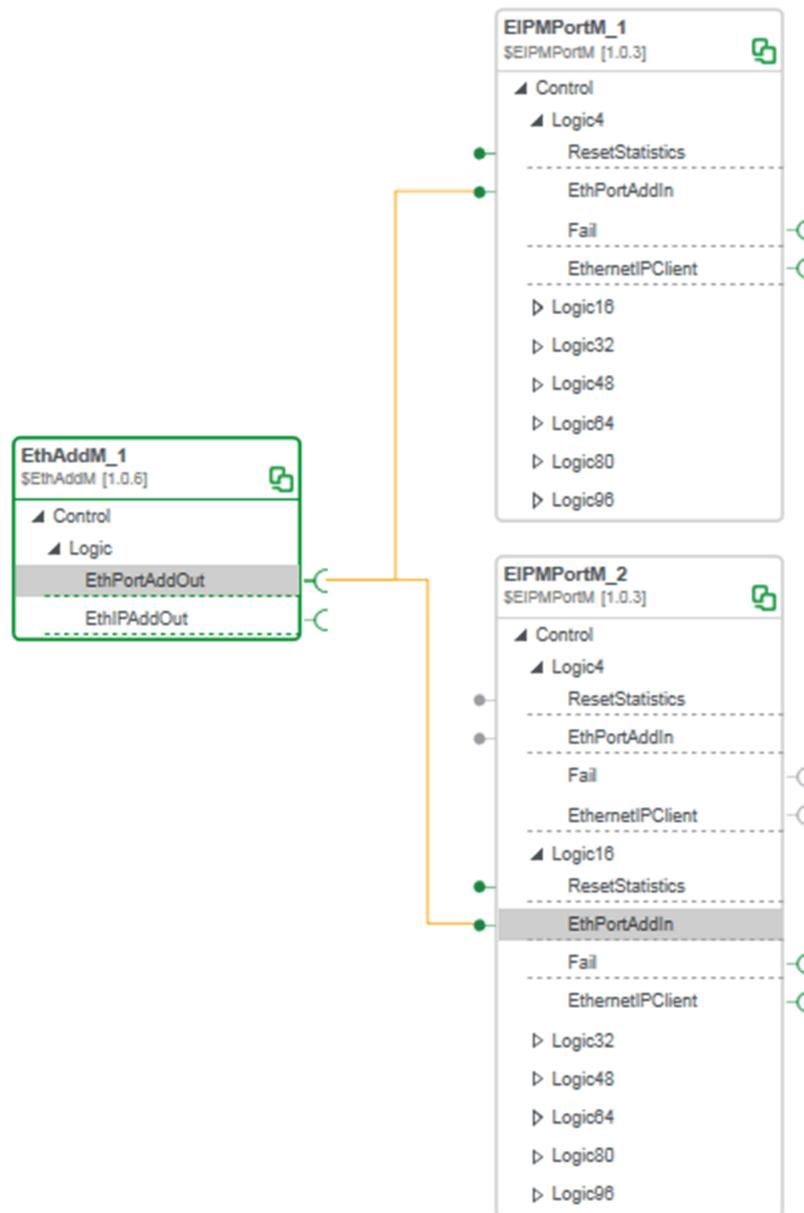
NOTE: If no device-specific topological template is available for the device that you are using, use a generic topological device template, page 531 and its application counterpart `$EIPMCClient`.

Using several Communication Port Instances

To use one topological communication port with several communication port instances (application), you need to instantiate the following intermediate templates (fan-out type application templates). Each instance of such template accepts links to n communication port instances.

Communication port template	Intermediate template
<code>\$EIPMPortM</code>	<code>\$EthAddM</code>

The following figure shows an example of how to link two instances of the communication port template to the instance of an intermediate template. In the hardware mapping, the interface of the intermediate template instance is mapped to the interface of the Ethernet port of an M580 controller.



NOTE: The **EthIPAddOut** interface is not used.

Implementing Communication

To implement communication between a controller and a device by using EtherNet/IP explicit messaging, follow these steps.

Step	Description	Stage
1	<p>Instantiate the EtherNet/IP messaging (EIPM) application template that corresponds to device that you want to use.</p> <p>NOTE: If no device-specific application template exists, use the \$EIPMCClient template. Create as many instances of it as you have requests. For example, for one read and one write request you need two instances.</p>	Instantiation
2	<p>Instantiate the \$EIPMPortM EtherNet/IP communication port template and configure its parameters, such as:</p> <ul style="list-style-type: none"> • Port Selection of the Control element: Enables the corresponding LogicX service where X represents the numeric part of the selected 	

Step	Description	Stage
	<p>port function. It needs to correspond to the maximum number of simultaneous requests that the communication module supports.</p> <p>For example, to EIPMPortM16 corresponds Logic16.</p> <ul style="list-style-type: none"> • Simultaneous Sends of the LogicX service to optimize communication by reducing the number of simultaneous requests that are sent. 	
3	If you are using one topological communication port with several communication port instances, instantiate the corresponding intermediate template, page 795.	
4	<p>Open the Asset Workspace Editor and link the communication port instance to:</p> <ul style="list-style-type: none"> • The device instance. • The intermediate template instance (if used). <p>NOTE: In case of a generic device, the device instance is the \$EIPMCClient template instance.</p>	
5	Configure an M580 controller and its communication module.	Configuration
6	<p>Create, page 531 the corresponding topological device entity and configure its IP address and other applicable properties (for example, \$EIPMPMESWTHW).</p> <p>NOTE: If no device-specific topological template exists, use the \$EIPGenericDeviceHW generic device template.</p>	
7	Create an Ethernet network.	
8	Connect the communication module of the controller and the topological device entity to the same Ethernet network.	
9	Assign the Control facets of the device, the communication port instances, and the intermediate template instance to the Control Participant project.	Assignment
10	Generate the project.	Generation
11	Map the executable of the Control Participant project to the engine of the controller, page 580.	Mapping (service)
12	Map the EMPortChM interface of the communication port instance (or of the intermediate template instance) to its topological counterpart.	Mapping (hardware)
13	Map the interface of the application instance of the device to the interface of the topological device instance.	

To complete the other steps to engineer the system that contains the controller and the device, follow the system engineering life cycle described in this manual.

NOTE: For configuration and setup details of devices and communication ports, refer to the help of the Schneider Electric libraries for EcoStruxure Process Expert.

Managing IP Address Changes

The topological templates of devices communicating by using Modbus TCP Ethernet and EtherNet/IP explicit messaging use a variable to define the IP address of the device.

During deployment, the templates set up the initial value of this variable, which is connected to the pin of the DFB implementing the device.

When you change the IP address of a device by changing the hardware mapping of the facet modeling the device and deploy the change by using the **Deploy Changes** command, only the initial value, in this case the IP address, is changed online. The current value remains the same.

The IP address change becomes effective, either through a cold start of the controller or by replacing the current value of the variable in the animation table by the new initial value.

Communication Through Modbus Explicit Messaging

Overview

This section contains a procedure describing how to manage communication between devices and M340 or Quantum controllers through Modbus explicit messaging.

It also contains a set of four examples illustrating how to use Modbus serial communication templates and do the required hardware mapping. Each example uses a different architecture based on the M580 platform. The examples use templates of the General Purpose library Classic.

Refer to the example overview and select the one that applies to your situation. From there, a link allows you to jump directly to the topic describing it.

Communication with Devices Through Modbus Explicit Messaging

About Communication Port Templates for Modbus Explicit Messaging

The software supports communication through Modbus explicit messaging between common devices and various controller platforms by using specific communication templates.

Controller platform	Communication module	Communication port template
Modicon M580	NOM ⁽¹⁾ communication module in the local rack.	\$MBPortM
	NOM ⁽¹⁾ communication module in a drop.	\$MBPortM58X80
Modicon M340	NOM communication module in the local rack.	\$MBPortM
Modicon Quantum	NOM communication module in an X80 drop.	\$MBPortQX80

(1) For example, BMXNOM0200.2 communication module

Communication through an Ethernet port of the controller is also supported.

For a list of supported communication architectures, refer to the help of the Schneider Electric libraries for EcoStruxure Process Expert.

NOTE: If no device-specific application template is available for the device that you are using, use the \$MBCClient template.

About Intermediate Templates

Intermediate templates are used when more than one communication port instance is used according to the following rules:

- **\$MBAAddM:** When a physical communication module (NOC, NOE, or NOM) needs to pass its port address to more than one communication port instance. Use one template per physical communication port.
This can be case when you are linking more devices to a communication port instance than this instance can handle requests per cycle. For example, The DFB of the \$MBPortM58X80 communication port template can send a maximum of 4 simultaneous requests per cycle. If you want to send 10 requests per cycle, you need to use 3 instances of the \$MBPortM58X80 template. These 3 instances are linked to an instance of the \$MBAAddM template.
- **\$EthAddM:** When either of these conditions is fulfilled:
 - A communication module (CPU, NOC, or NOE) needs to pass its Ethernet port address to more than one communication port instance.
 - A CRA module needs to pass its IP address to more than one communication port instance.

The intermediate templates allow grouping the interfaces coming from several communication port instances to connect them to the corresponding interfaces of the topology. This is because a 1 to N mapping is not possible for topological mapping interfaces.

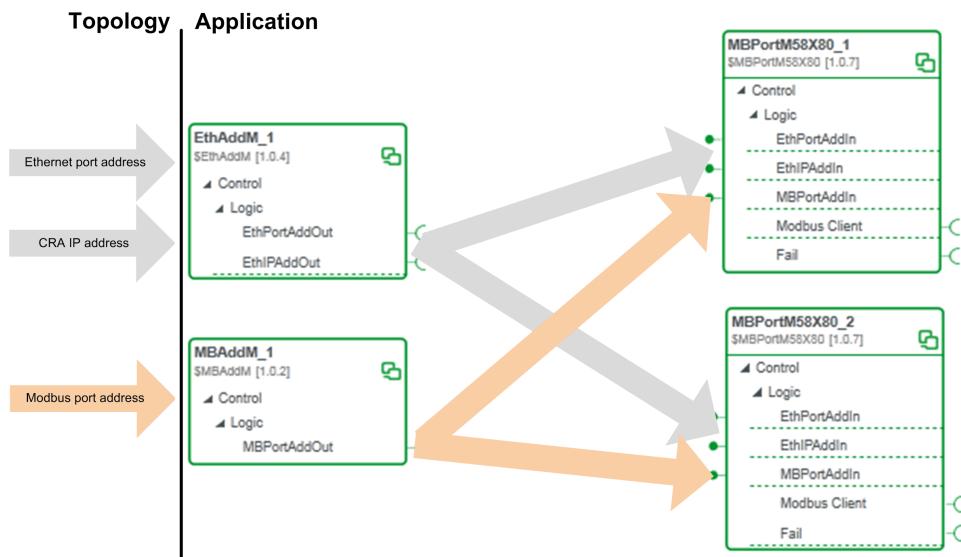
Together, the \$MBAAddM and \$EthAddM templates expose the same interfaces as one \$MBPortM58X80 template.

The table indicates which intermediate templates to instantiate depending on the communication port template that is used and also indicates the data that is transferred to the communication port instance.

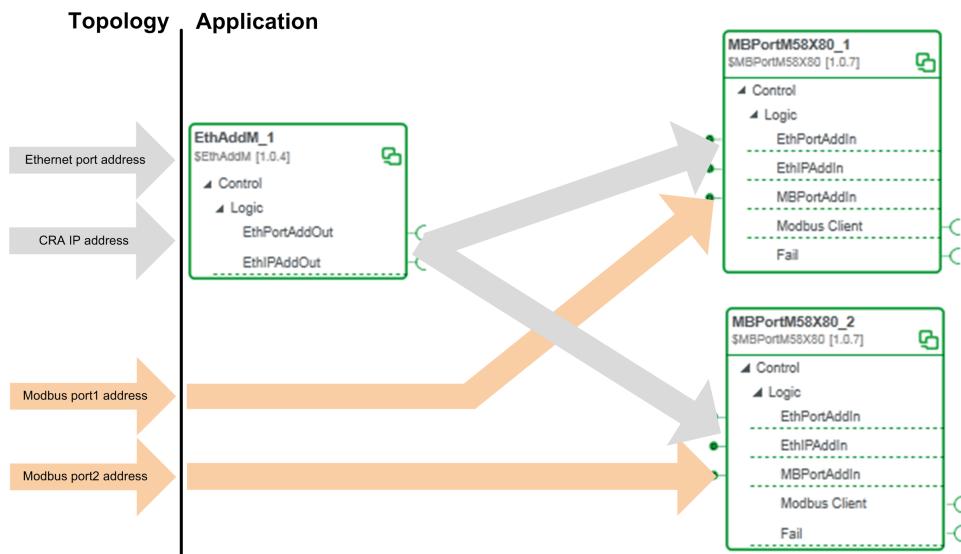
Communication port template	Intermediate template	Transferred address
\$MBPortM	\$MBAddM	Port address of NOM communication module.
\$MBPortM58X80	\$MBAddM	Port address of NOM communication module.
	\$EthAddM	Port address of M580 CPU or NOC communication module. IP address of the CRA module.
\$MBPortQX80	\$MBAddM	Port address of NOM communication module.
	\$EthAddQ	Port address of Quantum CPU or NOC communication module. IP address of the CRA module.

The following figures illustrate the transfer of data between the topology and the application in various scenarios.

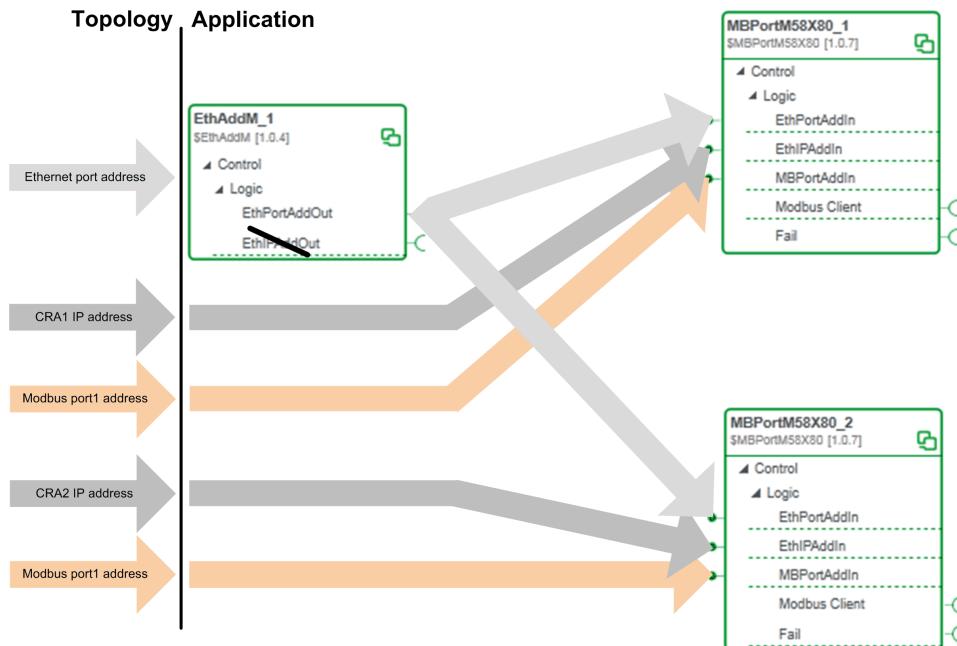
Multiple communication port instances per physical NOM port. In this case, only one NOM port is used and located in a drop.



One communication port instance per physical NOM port. In this case, the two NOM ports are located in the same drop.



One communication port instance per physical NOM port. In this case, the two NOM ports are located in separate drops.



Implementing Communication with Controllers Through Modbus Port

To implement communication between an M580, M340, or Quantum controller with Modbus port and devices through Modbus explicit messaging, follow these steps.

Step	Description	Stage
1	Instantiate the process template that corresponds to the piece of equipment that you want the device to communicate with (for example, \$Motor).	Instantiation
2	Instantiate a Modbus explicit messaging application template that corresponds to the device that you want to use (for example, \$TesysTMB), and configure its Modbus address. NOTE: If no device-specific application template exists, use the \$SMBClient template. Create as many instances of it as you have requests. For example, for one read and one write request you need two instances.	
3	Instantiate the corresponding communication port template, page 799.	
4	If you are using one topological communication port with several communication port instances, instantiate the corresponding intermediate template, page 799.	
5	Open the Asset Workspace Editor and link the device instance to the communication port and the process instance. NOTE: In case of a generic device, the device instance is the \$SMBClient template instance.	
6	Configure the controller with a Modbus port as master.	Configuration
7	Assign the Control facets of the process, device, and communication port instances to the Control Participant project so that the execution order, page 381 is as follows: 1. The facet of the Modbus explicit messaging device instance (for example, <i>TesysTMB_1</i>). 2. The facet of the communication port instance (for example, <i>MBPort58X80_1</i>).	Assignment
8	Generate the project.	Generation

Step	Description	Stage
9	Map the executable of the Control Participant project to the engine of the controller.	Mapping (service)
10	If the communication module is in the local rack, map by dragging from right to left: <ul style="list-style-type: none"> The EMPortChM or EMPortChQ interface to its topological counterpart. The MBPortChM or MBPortChQ interface to the interface of the NOM module of the drop. If the communication module is in a drop, in addition, map by dragging from right to left: <ul style="list-style-type: none"> The CRAIPAddress interface to its topological counterpart. 	Mapping (hardware)

To complete the other steps to engineer the system, follow the system engineering life cycle described in this manual.

NOTE: For details on Modbus technology, refer to the help of the Schneider Electric libraries for EcoStruxure Process Expert.

Implementing Communication with Controllers Through Ethernet Port

To implement communication between an M340, M580, or Quantum controller with Ethernet port and devices through Modbus explicit messaging, follow these steps.

Step	Description	Stage
1	Instantiate the process template that corresponds to the piece of equipment that you want the device to communicate with (for example, \$Motor).	Instantiation
2	Instantiate a Modbus explicit messaging application template that corresponds to the device that you want to use (for example, \$TsysTMB), and configure the Modbus address. NOTE: If no device-specific application template exists, use the \$MCClient template. Create as many instances of it as you have requests. For example, for one read and one write request you need two instances.	
3	Instantiate the following communication port templates: <ul style="list-style-type: none"> Gateway template \$EGtwMB. \$EMPortM for a controller of the M340 or M580 platform. \$EMPortQ for a controller of the Quantum platform. 	
4	Open the Asset Workspace Editor and link the device instance to the gateway and the process instances. NOTE: In case of a generic device, the device instance is the \$MCClient template instance.	
5	In the Asset Workspace Editor , link the gateway instance to the port instance.	
6	Create an Ethernet network.	
7	Configure Ethernet communication of the controller.	
8	Connect the communication module of the controller to the Ethernet network.	
9	Assign the Control facets of the process, device, and communication port instances to the Control Participant project so that the execution order, page 381 is as follows: <ol style="list-style-type: none"> The facet of the Modbus explicit messaging device instance (for example, <i>TsysTMB_1</i>). The facet of the gateway instance (for example, <i>EGtwMB_1</i>). The facet of the communication port instance (for example, <i>EMPortM_1</i>). 	Assignment
10	Generate the project.	Generation

Step	Description	Stage
11	Map the executable of the Control Participant project to the engine of the controller.	Mapping (service)
12	Map the EMPortChM or EMPortChQ interface of the communication port instance (CO) to its topological counterpart (HO).	Mapping (hardware)

To complete the other steps to implement the system that contains the controller and devices, follow the system engineering life cycle described in this manual.

NOTE: For configuration and setup details of gateways, refer to the help of the Schneider Electric libraries for EcoStruxure Process Expert.

Example Overview

Supported Architectures

These examples are based on a controller of the M580 platform configured with one BMENOC0301.2 communication module.

They can be used also for controllers of the Quantum platform by using the corresponding templates with the Q variant (for example, \$EthAddQ or \$MBPortQX80).

NOTE: The examples are not meant to represent a functional system. Only the elements that are relevant for the examples are configured and illustrated. Default identifiers have been edited to be more comprehensible.

About Communication Port Templates for Modbus Explicit Messaging

Various Modbus serial communication port templates, page 799 exist depending on the architecture, which contains the BMXNOM0200 communication module and its location.

When the communication module is located in a drop, the \$MBPortM58X80 template is used. Its instance exposes three mapping interfaces (**HW Interface** column) to transfer the required data to the process objects:

- **MBPortChM:** The port address of the NOM communication module.
- **EMPortChM:** The port address of the CPU or NOC communication module.
- **CRAIPAddress:** The IP address of the CRA module.

Available for mapping				
AppInstance	AppFacet	AppTemplate	AppMappingInterface	
MBPortM58X80	MBPortM58X80_MBPM58X80	\$MBPortM58X80	MBPortChM	
MBPortM58X80	MBPortM58X80_MBPM58X80	\$MBPortM58X80	EMPortChM	
MBPortM58X80	MBPortM58X80_MBPM58X80	\$MBPortM58X80	CRAIPAddress	

Instantiate one \$MBPortM58X80 template for each physical communication port used. When you need several communication port templates, use also intermediate templates, page 799.

The \$MBPortM58X80 template allows mapping with the topology, which exposes the following interfaces in the **Hardware Mapping Editor** when the NOM communication module is located in a drop.

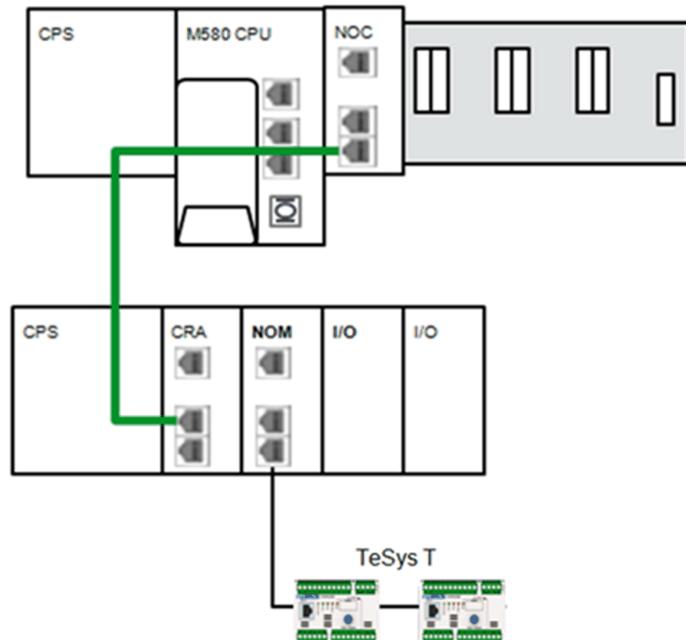
Available for mapping				
HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	
0:PrILocal 0:D:0:R:0:BMEP584040	\$M580CPUEthCRP	MBTCPEthM580.EMPortChM	\$MEthernetPortCH/HO	
0:PrILocal 0:D:0:R:0:2:BMENOC0301.2	\$MNOCIOSClient	MBTCPEth.EMPortChM	\$MEthernetPortCH/HO — 1	
2:EhRIO 1:D:0:R:0:BMECRA31210.2	\$CRAEth	CRAIPAddress	\$EthernetAddress/HO — 2	
2:EhRIO 1:D:0:R:0:1:BMXNOM0200.3	\$HWNOMMB	MB_M340_M580_Master_CH0.MModbusPortCH	\$MModbusPortCH/HO — 3	
2:EhRIO 1:D:0:R:0:1:BMXNOM0200.3	\$HWNOMMB	MBControllerSlave_CH1.ModbusAddress	\$ModbusAddress/HO	

Item	Description
1	Port address of the NOC communication module in the local rack.
2	IP address of the CRA module.
3	Port address of the NOM communication module when the NOM port is configured for Modbus link.

Example A

The table indicates the specifics of this example in terms of remote drop configuration and application objects used.

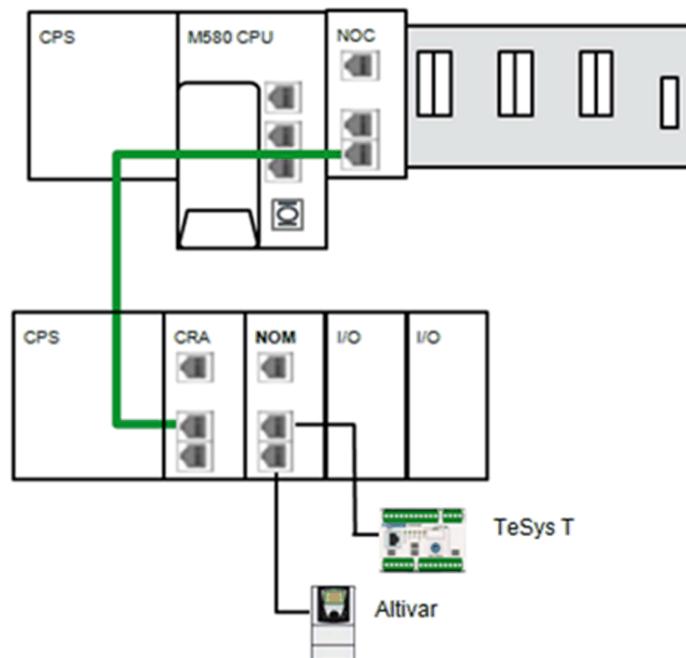
M580 remote drop configuration	Application objects	Example description
1 BMXNOM0200 communication module.	2 devices communicating through 1 communication port.	A, page 808



Example B

The table indicates the specifics of this example in terms of drop configuration and application objects used.

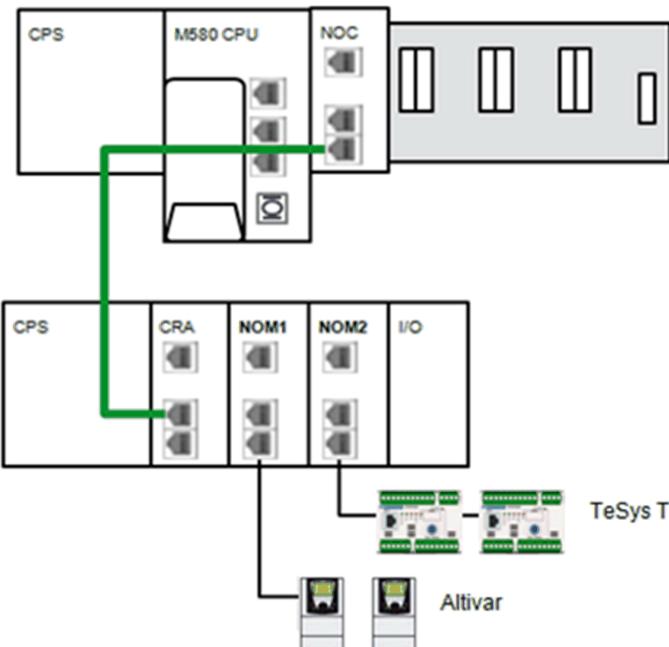
M580 remote drop configuration	Application objects	Example description
1 BMXNOM0200 communication module.	2 devices, each communicating through 1 communication port (2 ports total).	B, page 812



Example C

The table indicates the specifics of this example in terms of drop configuration and application objects used.

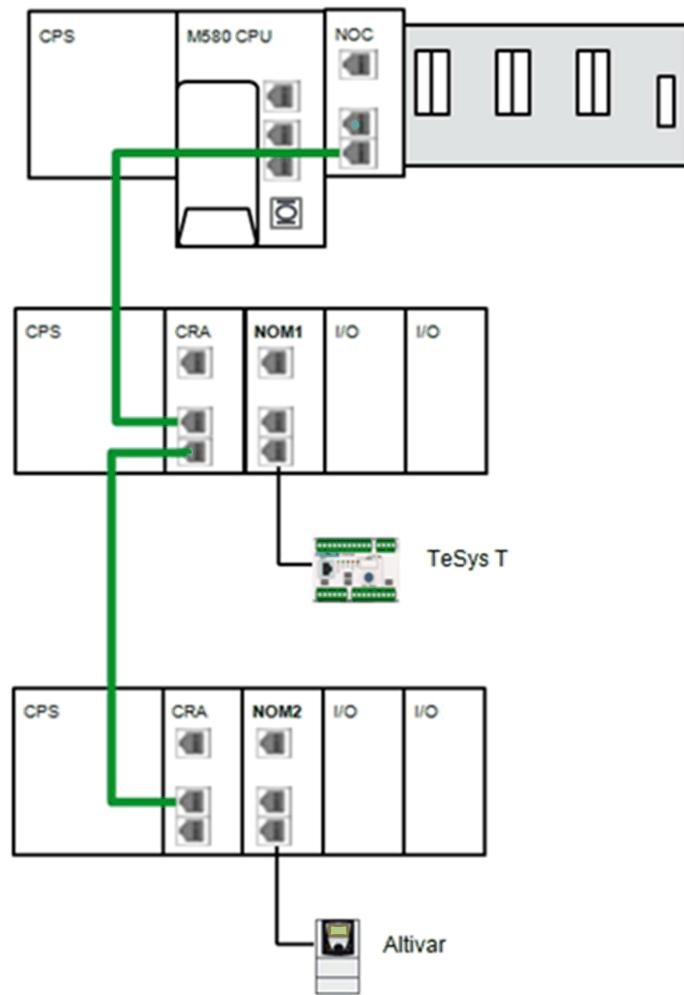
M580 remote drop configuration	Application objects	Example description
2 BMXNOM0200 communication modules	For each NOM, 2 devices communicating through 1 communication port.	C, page 818



Example D

The table indicates the specifics of this example in terms of drop configuration and application objects used.

M580 remote drop configuration	Application objects	Example description
2 drops with 1 BMXNOM0200 communication module in each.	For each drop, 1 device communicating through 1 communication port.	D, page 825

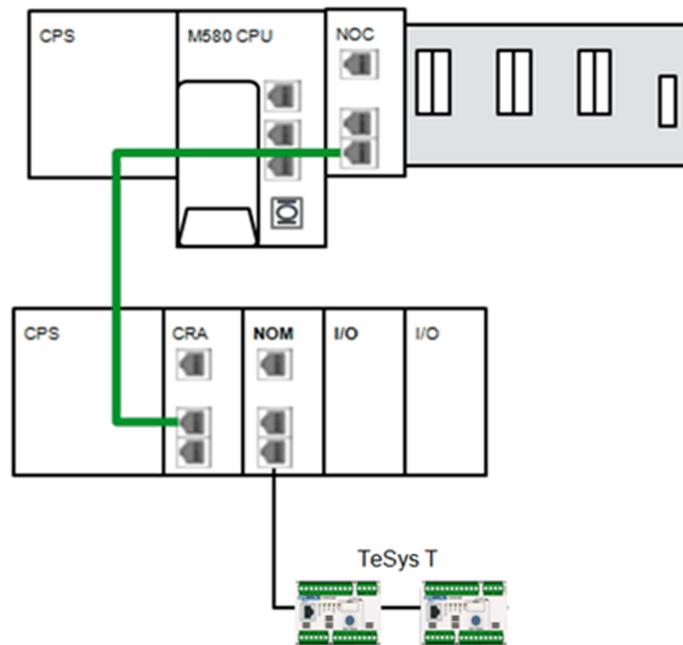


Example A

Example Description

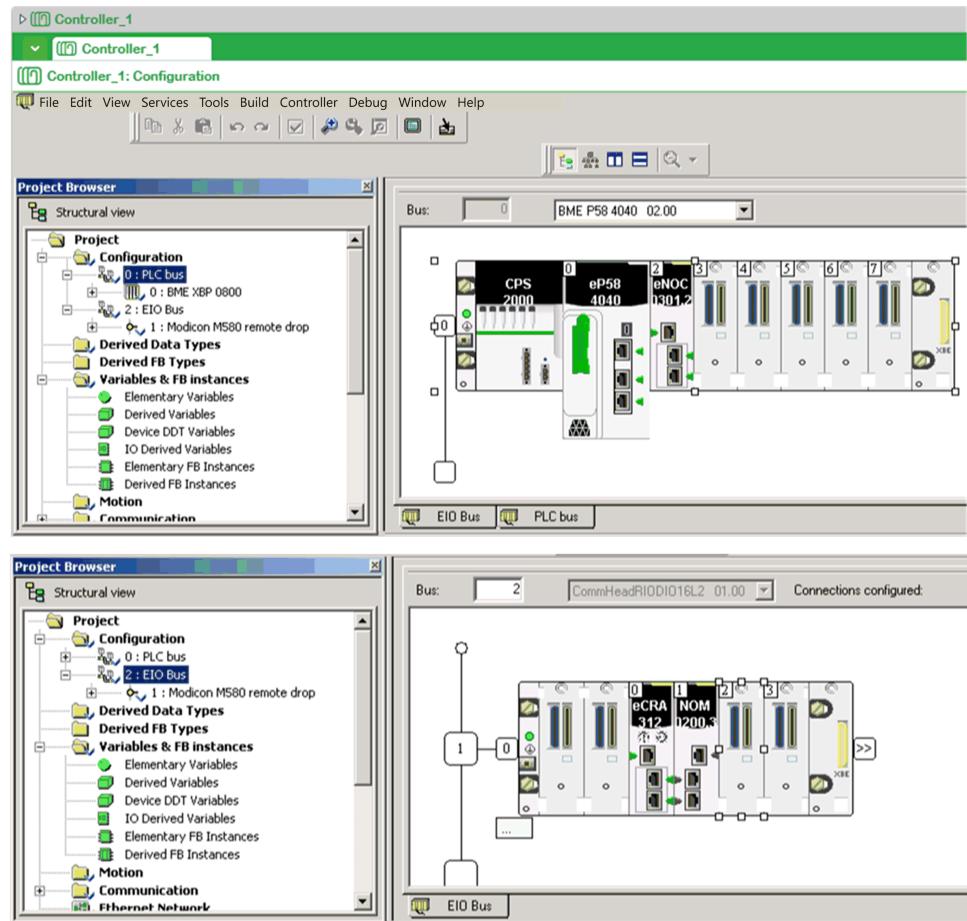
The table indicates the specifics of this example in terms of remote drop configuration and application objects used.

M580 remote drop configuration	Application objects
1 BMXNOM0200 communication module.	2 devices communicating through 1 communication port.



Configuration Stage

The following figure shows the controller configuration with the drop and the communication module.



Instantiation Stage

The following Instances are required to implement communication at the application level with two devices (TeSys T motor controllers) by using Modbus serial.

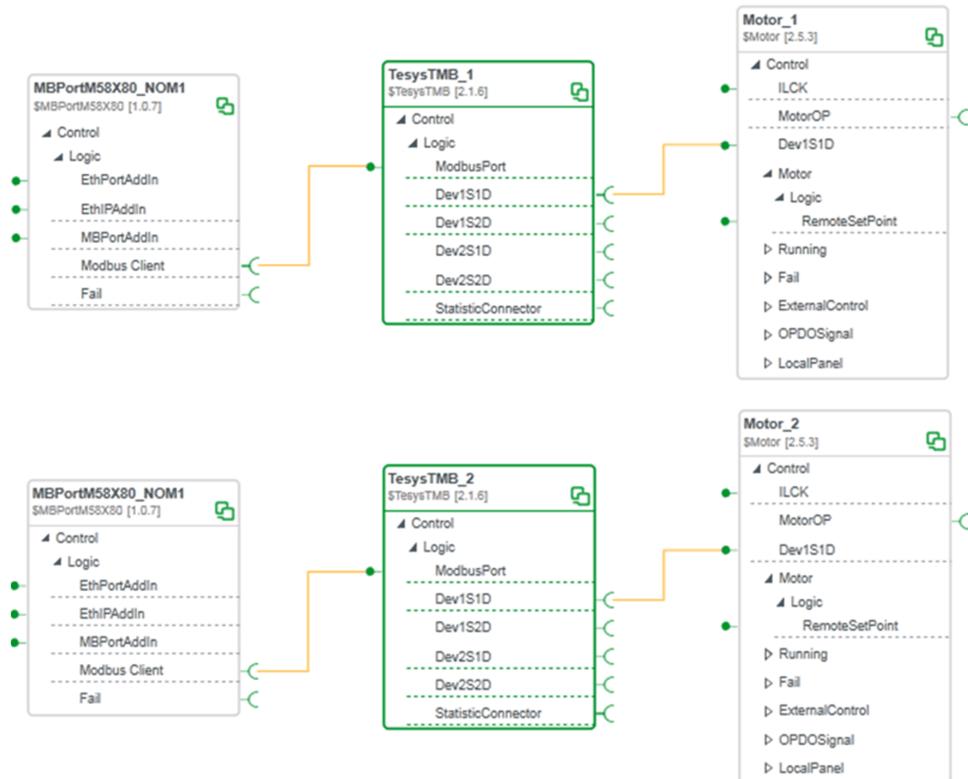
The *MBPortM58X80_NOM1* Modbus serial communication port instance is to be used when the NOM communication module is located in the remote drop of an M580 controller. In this example, one instance is created because one physical communication port is used.

Identifier	Template	Version	Data	Link	Assigned State
MBPortM58X80_NOM1	SMBPortM58X80	1.0.7	Valid	Valid	Unassigned
Motor_1	SMotor	2.5.3	Valid	Valid	Unassigned
Motor_2	SMotor	2.5.3	Valid	Valid	Unassigned
TesysTMB_1	STesysTMB	2.1.6	Valid	Invalid	Unassigned
TesysTMB_2	STesysTMB	2.1.6	Valid	Invalid	Unassigned

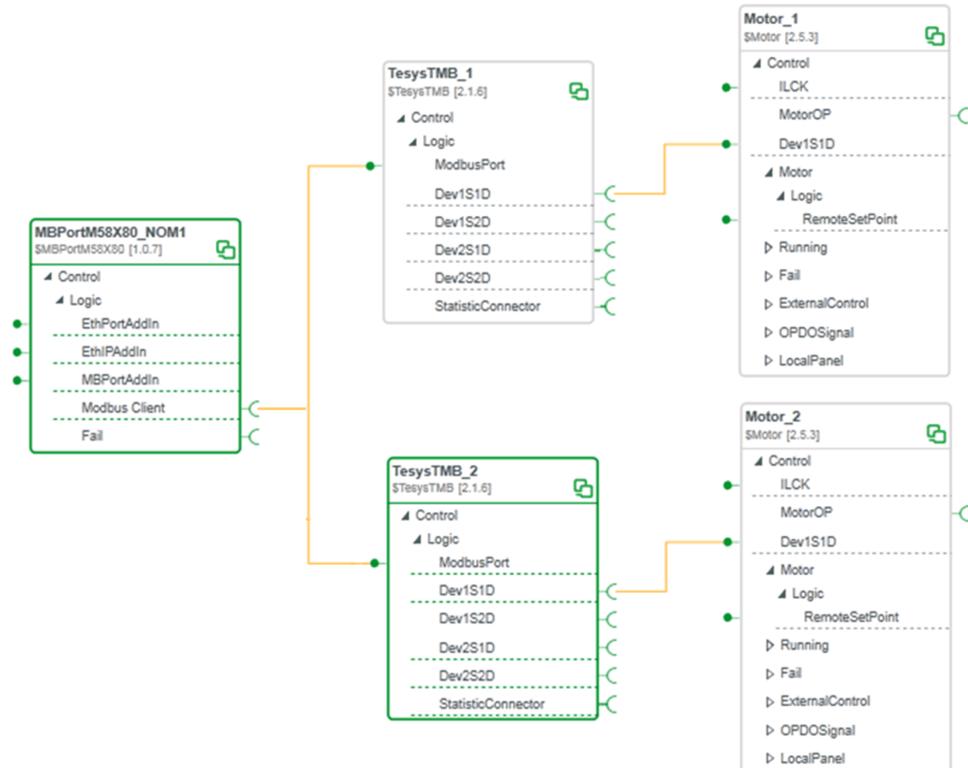
Application Interface Links

The following links are created in the **Asset Workspace Editor** for the TeSys T devices.

The process instance (*Motor_n*), the device instance (*TesysTMB_n*), and the communication port instance (*MBPortM58X80_NOM1*) are linked.



These steps result in the following links.



The communication port instance (MBPortM58X80_NOM1) passes the following data to the process instances:

- Port address of the NOC communication module
- IP address of the CRA module of the drop
- Port address of the NOM communication module

Control Facet Assignment

Facets of the instances are assigned to a Control Participant project in an FBD section.

Identifier	Container	Instance	Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation
FBDSection_1	TesysTMB_1	\$TesysTMB	Valid	0	TesysTMB_1_TesysT	STesysTMB_UL	Control	0	Assigned	Generated
FBDSection_1	TesysTMB_2	\$TesysTMB	Valid	1	TesysTMB_2_TesysT	STesysTMB_UL	Control	1	Assigned	Generated
FBDSection_1	MBPortM58X80_NOM1	\$MBPortM58X80	Valid	2	MBPortM58X80_NOM1_MBPM58X80	SMBPortM58X80_UL	Control	2	Assigned	Generated
FBDSection_1	Motor_1	\$Motor	Valid	3	Motor_1_DEVCTL	SDEVCTL_UL	Control/Motor	3	Assigned	Generated
FBDSection_1	Motor_1	\$Motor	Valid	4	Motor_1_CONSUM	SCONDSUM_UL	Control/Failures	4	Assigned	Generated
FBDSection_1	Motor_1	\$Motor	Valid	5	Motor_1_CONDSUMER	SCONDSUMER_UL	Control/Interlocks	5	Assigned	Generated
FBDSection_1	Motor_2	\$Motor	Valid	6	Motor_2_DEVCTL	SDEVCTL_UL	Control/Motor	6	Assigned	Generated
FBDSection_1	Motor_2	\$Motor	Valid	7	Motor_2_CONSUM	SCONDSUM_UL	Control/Failures	7	Assigned	Generated
FBDSection_1	Motor_2	\$Motor	Valid	8	Motor_2_CONDSUMER	SCONDSUMER_UL	Control/Interlocks	8	Assigned	Generated

Generation and Service Mapping

The Control Participant project is generated.

An executable is created in the Control Participant and mapped to the M580 controller.

Hardware Mapping

The **Hardware Mapping Editor** shows the following interfaces.

(Lines have been drawn to show the mappings that you need to do.)

HWInstance	HWTTemplate	HWMappingInterface	HWInterfaceType	Available for mapping
:PILocal 0 D 0 R 0.BMEPE584040	\$M580CPUetherCRP	MBTCPETHMS80_EMPortCHM	\$MEthernetPortCHCHO	MBPortM58X80_NOM1
:PILocal 0 D 0 R 02.BMENOC0301.2	\$MNOCIOSClient	MBTCPETH_EMPortCHM	\$MEthernetPortCHCHO	MBPortM58X80_NOM1
:EthRIO 1 D 0 R 0 BMECRA31210.2	\$CRAIEth	CRAIAddress	\$EthernetAddress/HO	MBPortM58X80_NOM1
:EthRIO 1 D 0 R 01.BMXNOM0200.3	\$HWNOMMB	MB_M340_MS80_Master_CH0.MMModbusPortCH	\$ModbusPortCHHO	TesysTMB_1
:EthRIO 1 D 0 R 01.BMXNOM0200.3	\$HWNOMMB	MBCControllerSlave_C1H.ModbusAddress	\$ModbusAddress/HO	TesysTMB_2

NOTE: You do not need to map **MBAddress** interfaces (for example, of facet **TesysTMB_1_TesysT**).

Completing the System Engineering Life Cycle

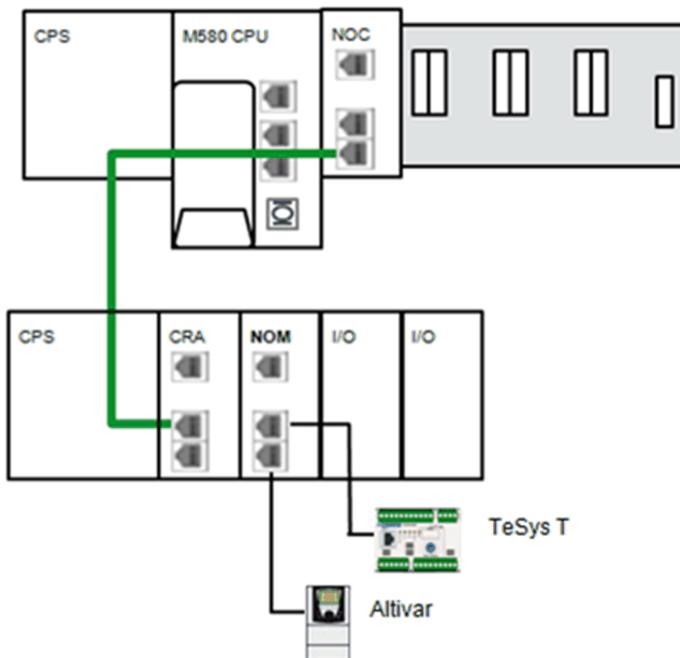
Complete the creation of the system by performing the other steps of the system engineering life cycle.

Example B

Example Description

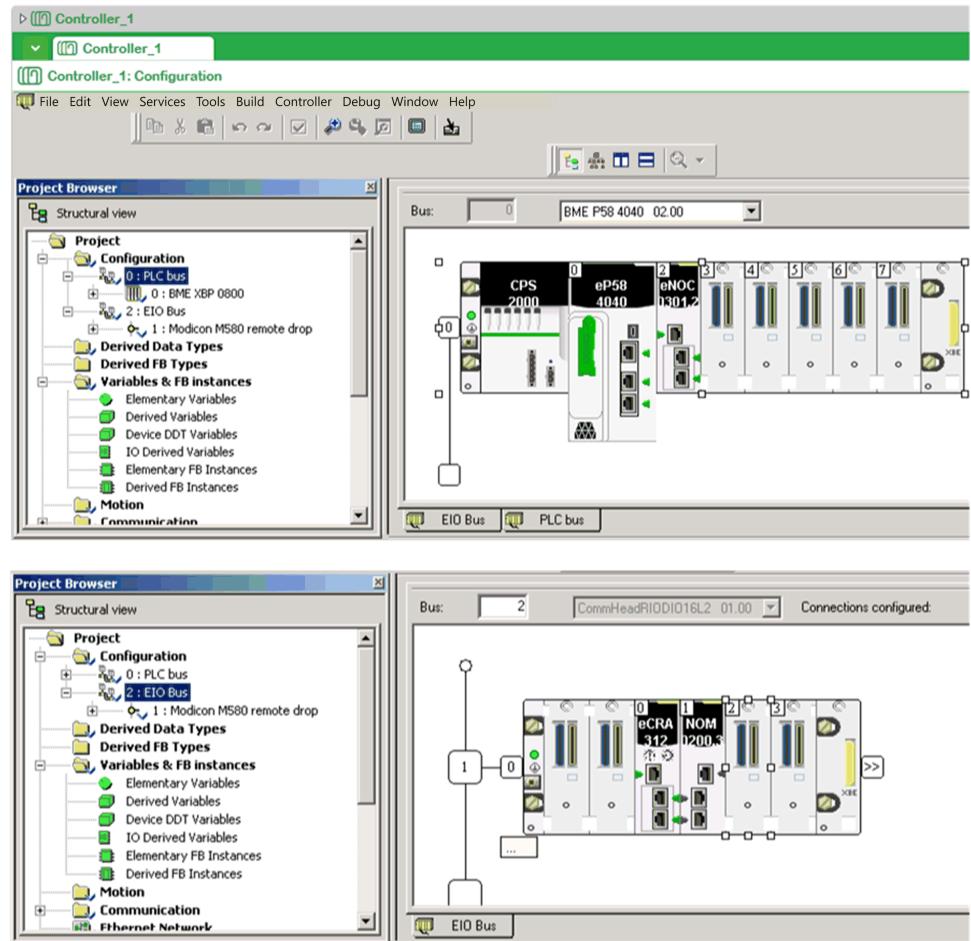
The table indicates the specifics of this example in terms of drop configuration and application objects used.

M580 remote drop configuration	Application objects
1 BMXNOM0200 communication module.	2 devices, each communicating through 1 communication port (2 ports total).



Configuration Stage

The following figure shows the controller configuration with the drop and the communication module.



Instantiation Stage

The following Instances are required to implement communication at the application level with two devices (a TeSys T motor controller and an Altivar variable speed drive) by using Modbus serial.

The *MBPortM58X80_x* Modbus serial communication port instances are to be used when the NOM communication module is located in the drop of an M580 controller. In this example, two instances are created, one for each physical communication port used.

Instances are grouped in two folders for clarity, one for each communication port instance.

Identifier	Template	Version	Data	Link	Assigned State
EthAddM_NOC_CRA	SEthAddM	1.0.4	Valid	Valid	Unassigned
MBPortM58X80_Port1	SMBPortM58X80	1.0.7	Valid	Valid	Unassigned
Motor_1	SMotor	2.5.3	Valid	Valid	Unassigned
TesysTMB_1	STesysTMB	2.1.6	Valid	Invalid	Unassigned

Identifier	Template	Version	Data	Link	Assigned State
ATV12MB_2	SATV12MB	2.1.7	Valid	Invalid	Unassigned
MBPortM58X80_Port2	SMBPortM58X80	1.0.7	Valid	Valid	Unassigned
Motor_2	SMotor	2.5.3	Valid	Valid	Unassigned

These steps result in the following links.

The communication port instance (MBPortM58X80_NOM1) passes the following data to the process instances:

- Port address of the NOC communication module
- IP address of the CRA module of the drop
- Port address of the NOM communication module

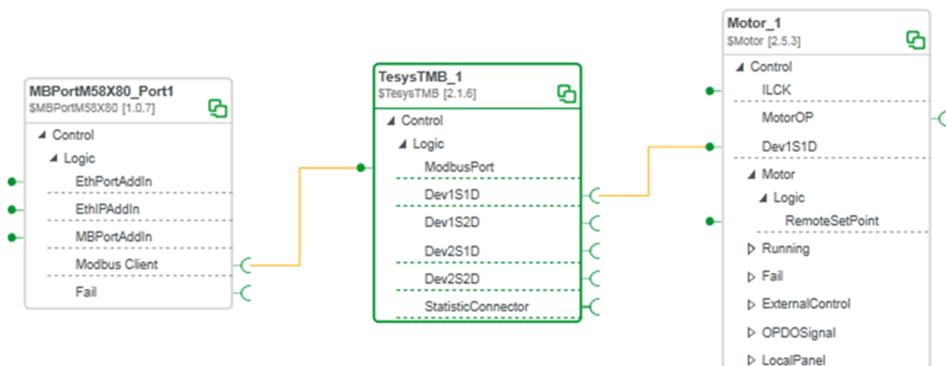
Intermediate Template Instances

Only intermediate template \$EthAddM_NOC_CRA is used because more than one communication port instance is used. It is common to both communication port instances. Intermediate template \$MBAddM is not used because on each NOM port, only one communication port instance is used.

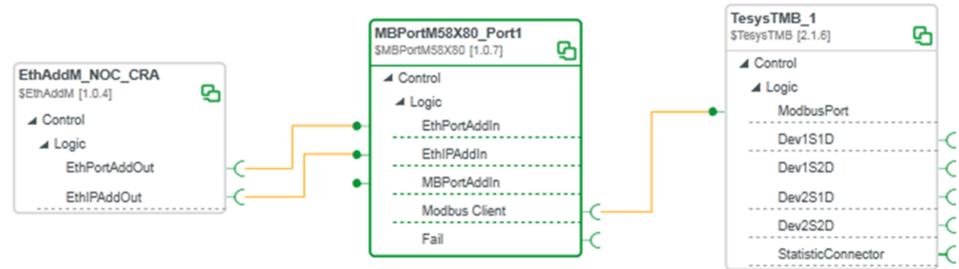
Creating Links

The following links are created in the **Asset Workspace Editor** for the TeSys T device.

The process instance (*Motor_1*), the device instance (*TesysTMB_1*), and the communication port instance (*MBPortM58X80_Port1*) are linked.

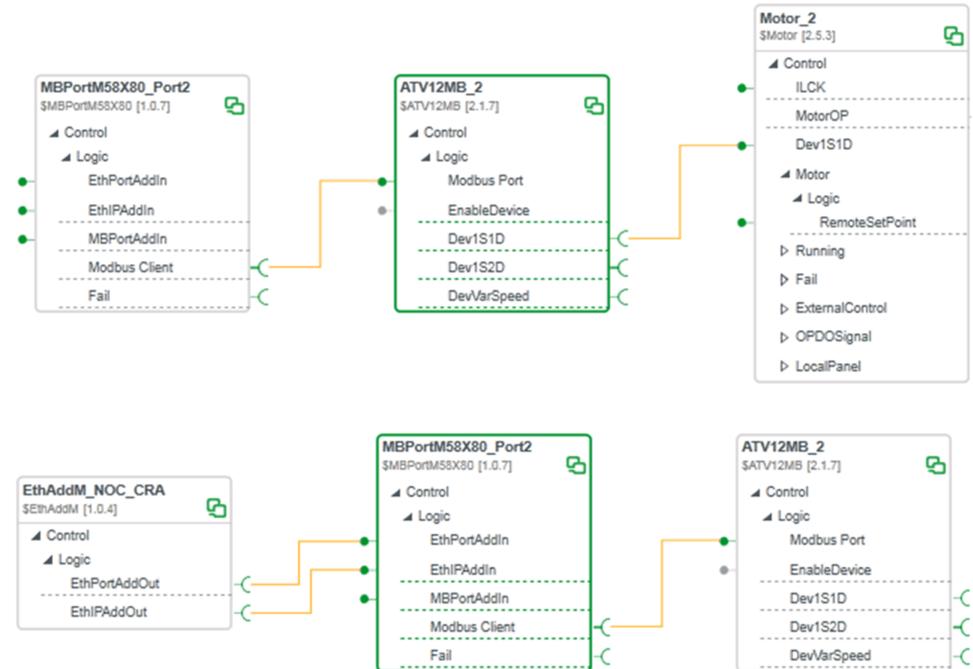


Then, link the *MBPortM58X80_Port1* instance to both interfaces of the *EthAddM_NOC_CRA* Ethernet intermediate template instance.

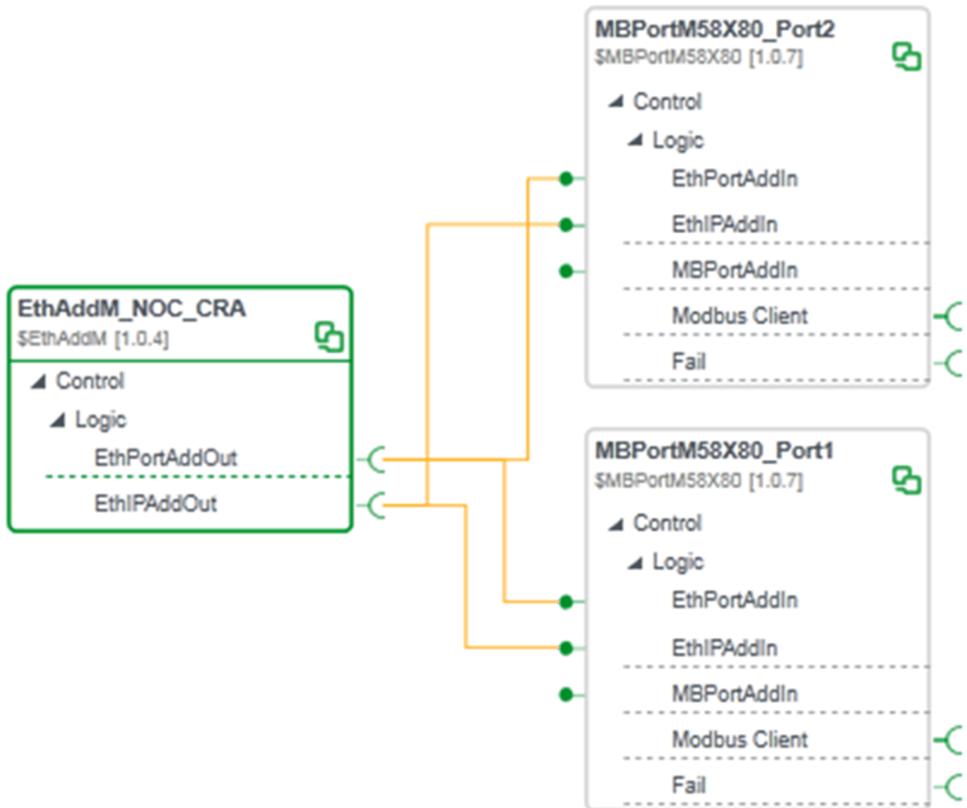


The **EthAddM_NOC_CRA** instance passes the NOC port address as well as the IP address of the **BMECRA312** module to the communication port instance.

Similarly, the following links are created in the **Asset Workspace Editor** for the Altivar device.



The following figure show the existing links from the perspective of the intermediate template.



Control Facet Assignment

Facets of the instances are assigned to a Control Participant project in two separate FBD sections.

Folder *B* is assigned to *FBDSection_1*.

FBDSection_1 - Assignments									
Identifier	Container	Instance	Instance Template	State	Facet	Facet Template	Path	Order	Assignment
FBDSection_1	TesysTMB_1	STesysTMB	Valid		TESYS_TMB_1_TesysT	STESYS_TMB_UL	Control	0	Assigned
FBDSection_1	MBPortM58X80_Port1	MBPortM58X80	Valid		MBPortM58X80_Port1_MBPM58X80	MBPortM58X80_UL	Control	1	Assigned
FBDSection_1	EthAddM_NOC_CRA	SEthAddM	Valid		ETHADDM_NOC_CRA_PA	SETHADDM_UL	Control	2	Assigned
FBDSection_1	Motor_1	SMotor	Valid		MOTOR_1_DEVCTL	SDEVCTL_UL	Control/Motor	3	Assigned
FBDSection_1	Motor_1	SMotor	Valid		MOTOR_1_CONDSUM	SCONDSUM_UL	Control/Failures	4	Assigned
FBDSection_1	Motor_1	SMotor	Valid		MOTOR_1_CONDSUM1	SCONDSUM1_UL	Control/Interlocks	5	Assigned

Folder *B_2* is assigned to *FBDSection_2*.

FBDSection_2 - Assignments									
Identifier	Container	Instance	Instance Template	State	Facet	Facet Template	Path	Order	Assignment
FBDSection_2	ATV12MB_2	SATV12MB	Valid		ATV12MB_2_ATV	SATV12MB_UL	Control	0	Assigned
FBDSection_2	MBPortM58X80_Port2	MBPortM58X80	Valid		MBPortM58X80_Port2_MBPM58X80	MBPortM58X80_UL	Control	1	Assigned
FBDSection_2	Motor_2	SMotor	Valid		MOTOR_2_DEVCTL	SDEVCTL_UL	Control/Motor	2	Assigned
FBDSection_2	Motor_2	SMotor	Valid		MOTOR_2_CONDSUM	SCONDSUM_UL	Control/Failures	3	Assigned
FBDSection_2	Motor_2	SMotor	Valid		MOTOR_2_CONDSUM1	SCONDSUM1_UL	Control/Interlocks	4	Assigned

Generation and Service Mapping

The Control Participant project is generated.

An executable is created in the Control Participant and mapped to the M580 controller.

Hardware Mapping

The **Hardware Mapping Editor** shows the following interfaces.

(Lines have been drawn to show the mappings that you need to do.)

mapping				Available for mapping			
HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	AppInstance	AppFacet	AppTemplate	AppMappingInfo
PrLocal 0:D:0 R:0 BMEP584040	\$M580CPUEHC_RP	MBTCPERMS80_EMPortCHM	\$MEthernetPortCHHO	ATV12MB_2	ATV12MB_2_ATV	SATV12MB	MBAddress
PrLocal 0:D:0 R:02 BMENOC0301.2	\$MNOCIOClient	MBTCPER_EMPortCRM	\$MEthernetPortCHHO	EthAddm_NOC_CRA	EthAddm_NOC_CRA_PA	SEIPAddm	EMPortCHM
EthRIO 1:D:0 R:0 BMCR4A31210.2	\$CR4AEth	CRAIPAddress	\$EthernetAddress/HO	EthAddm_NOC_CRA	EthAddm_NOC_CRA_PA	SEIPAddm	CRAIPAddress
EthRIO 1:D:0 R:01 BMXNOM0200.3	\$HWNOMMB	MB_M340_MS80_Master_CH0_MMModbusPortCH	SMModbusPortCHHO	MBPortMS80_Port1	MBPortMS80_Port1_MBPM58x80	MBPortMS80x80	MBPortCHM
EthRIO 1:D:0 R:01 BMXNOM0200.3	\$HWNOMMB	MB_M340_MS80_Master_CH1_MMModbusPortCH	SMModbusPortCHHO	MBPortMS80_Port2	MBPortMS80_Port2_MBPM58x80	MBPortMS80x80	MBPortCHM
				TesysTMB_1	TesysTMB_1_TesysT	STesysTMB	MBAddress

NOTE: You do not need to map **MBAddress** interfaces (for example, of facet **TesysTMB_1_TesysT**).

Completing the System Engineering Life Cycle

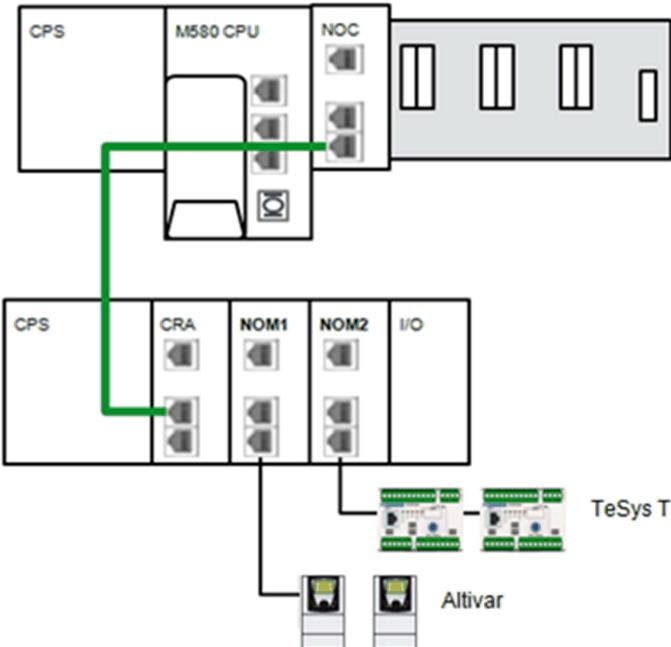
Complete the creation of the system by performing the other steps of the system engineering life cycle.

Example C

Example Description

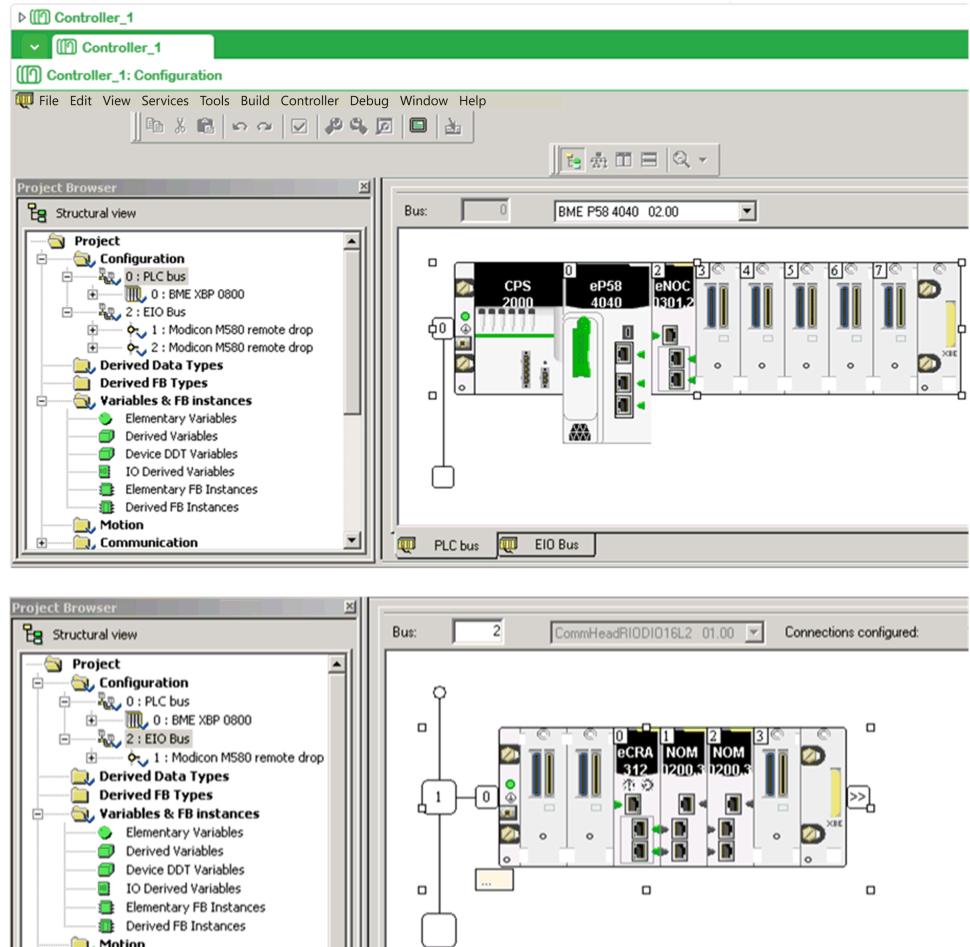
The table indicates the specifics of this example in terms of drop configuration and application objects used.

M580 remote drop configuration	Application objects
2 BMXNOM0200 communication modules	For each NOM, 2 devices communicating through 1 communication port.



Configuration Stage

The following figure shows the controller configuration with the drop and the communication module.



Instantiation Stage

The following Instances are required to implement communication at the application level with four devices (two TeSys T motor controllers and two Altivar variable speed drives) by using Modbus serial.

The *MBPortM58X80_x* Modbus serial communication port instances are to be used when the NOM communication module is located in the drop of an M580 controller. In this example, two instances are created, one for each physical communication port used.

Instances are grouped in two folders for clarity, one for each communication port instance.

The screenshot shows two tables of communication port instances in the Asset Workspace Editor:

Identifier	Template	Version	Data	Link	Assigned State
EthAddM_NOC_CRA	SEthAddM	1.0.4	Valid	Valid	Unassigned
MBPortM58X80_NOM1	SMBPortM58X80	1.0.7	Valid	Valid	Unassigned
Motor_10	SMotor	2.5.3	Valid	Valid	Unassigned
Motor_11	SMotor	2.5.3	Valid	Valid	Unassigned
TesysTMB_10	STesysTMB	2.1.6	Valid	Invalid	Unassigned
TesysTMB_11	STesysTMB	2.1.6	Valid	Invalid	Unassigned

Identifier	Template	Version	Data	Link	Assigned State
ATV61MB_20	SATV61MB	1.2.5	Valid	Invalid	Unassigned
ATV61MB_21	SATV61MB	1.2.5	Valid	Invalid	Unassigned
MBPortM58X80_NOM2	SMBPortM58X80	1.0.7	Valid	Valid	Unassigned
Motor_20	SMotor	2.5.3	Valid	Valid	Unassigned
Motor_21	SMotor	2.5.3	Valid	Valid	Unassigned

These steps result in the following links.

The communication port instance (MBPortM58X80_NOM1) passes the following data to the process instances:

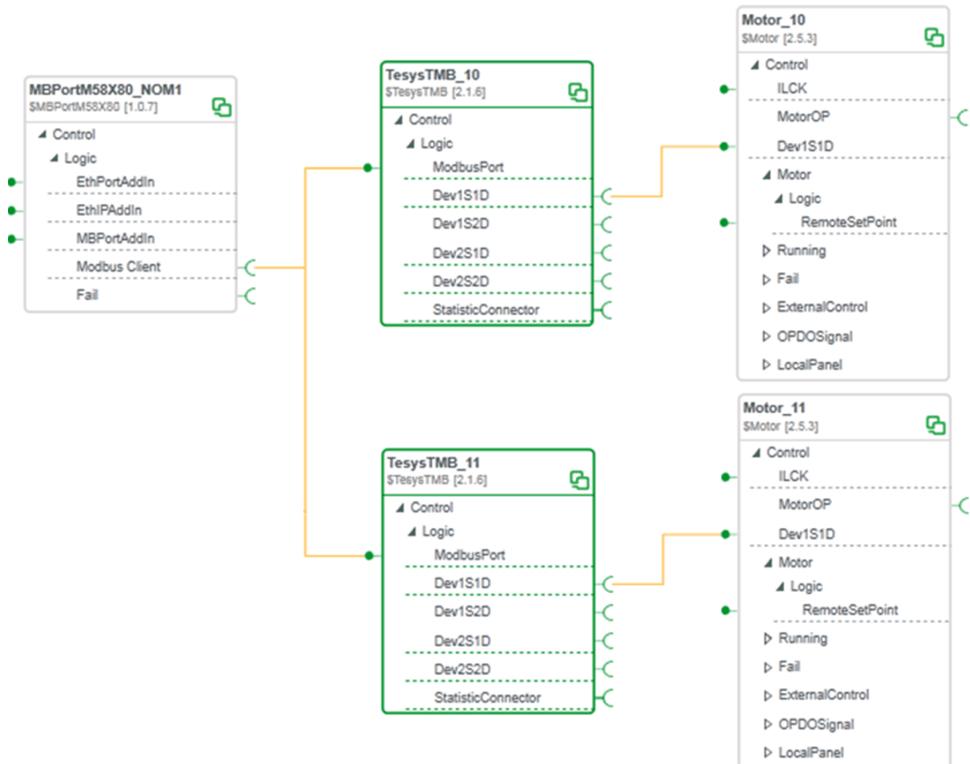
- Port address of the NOC communication module
- IP address of the CRA module of the drop
- Port address of the NOM communication module

Intermediate Template Instances

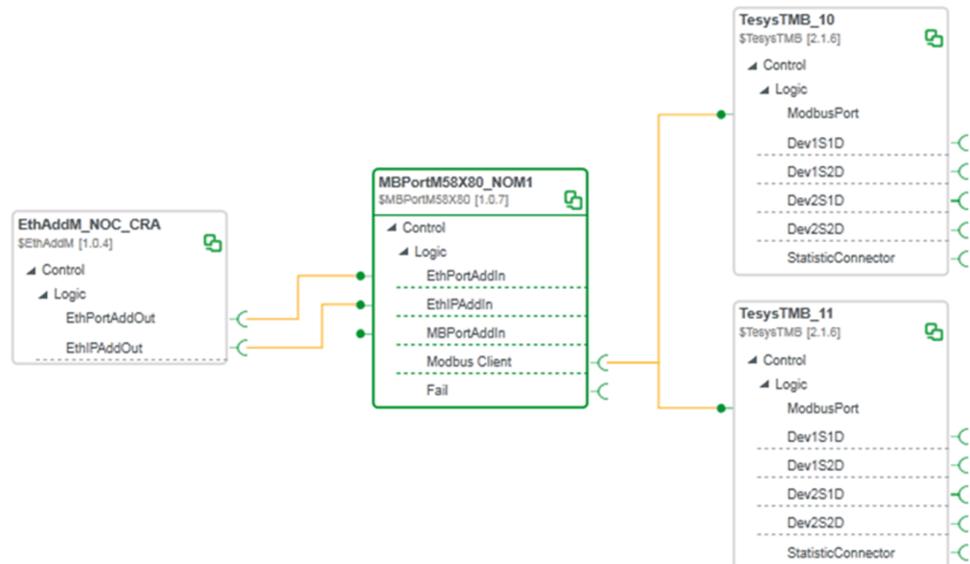
Only intermediate template \$EthAddM_NOC_CRA is used because more than one communication port instance is used. It is common to both communication port instances. Intermediate template \$MBAAddM is not used because on each NOM port, only one communication port instance is used.

Application Interface Links

The following links are created in the **Asset Workspace Editor** for both TeSys T devices.

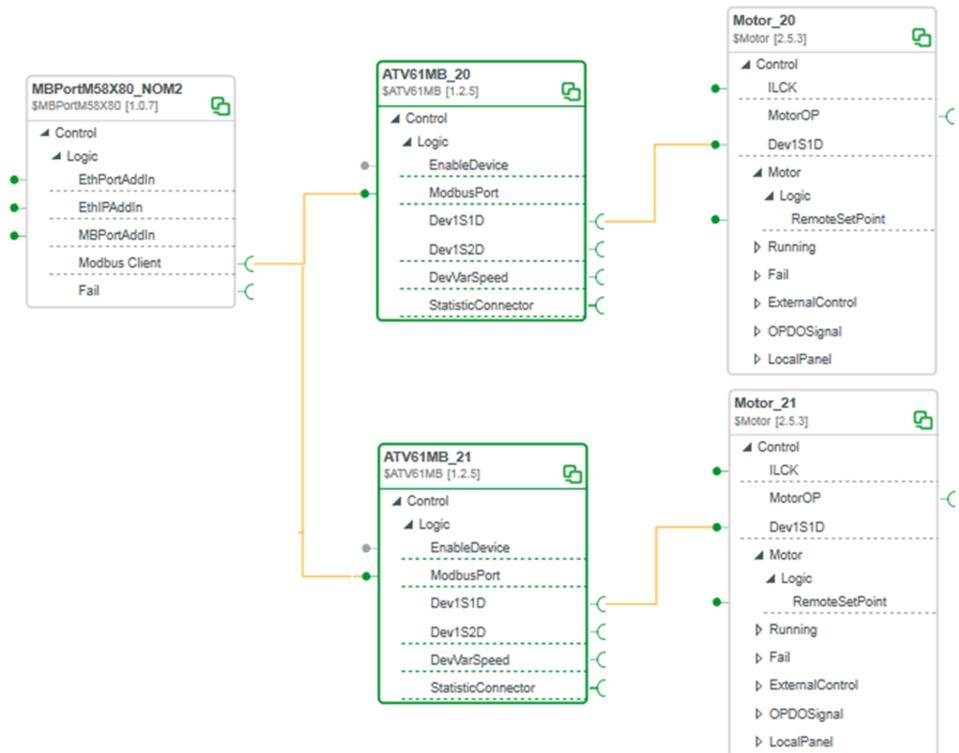


Then, link the communication port instance **MBPortM58X80_NOM1**, which communicates with the NOM communication module located in drop 1 to both interfaces of the intermediate template instance **EthAddM_NOC_CRA**.

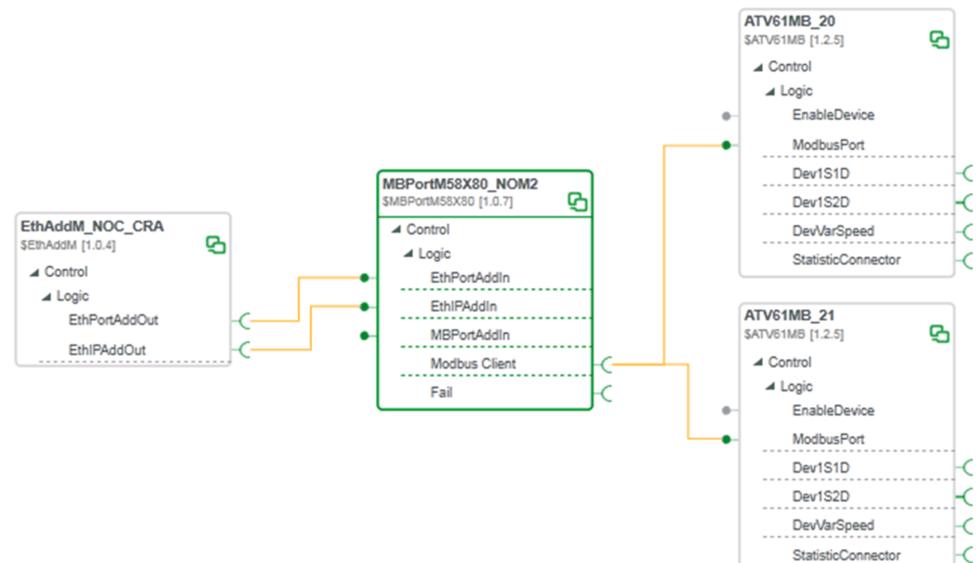


The **EthAddM_NOC_CRA** instance passes the physical NOC port address as well as the IP address of the **BMECRA312** module to the first communication port instance.

Similarly, the following links are created in the **Asset Workspace Editor** for both Altivar devices.

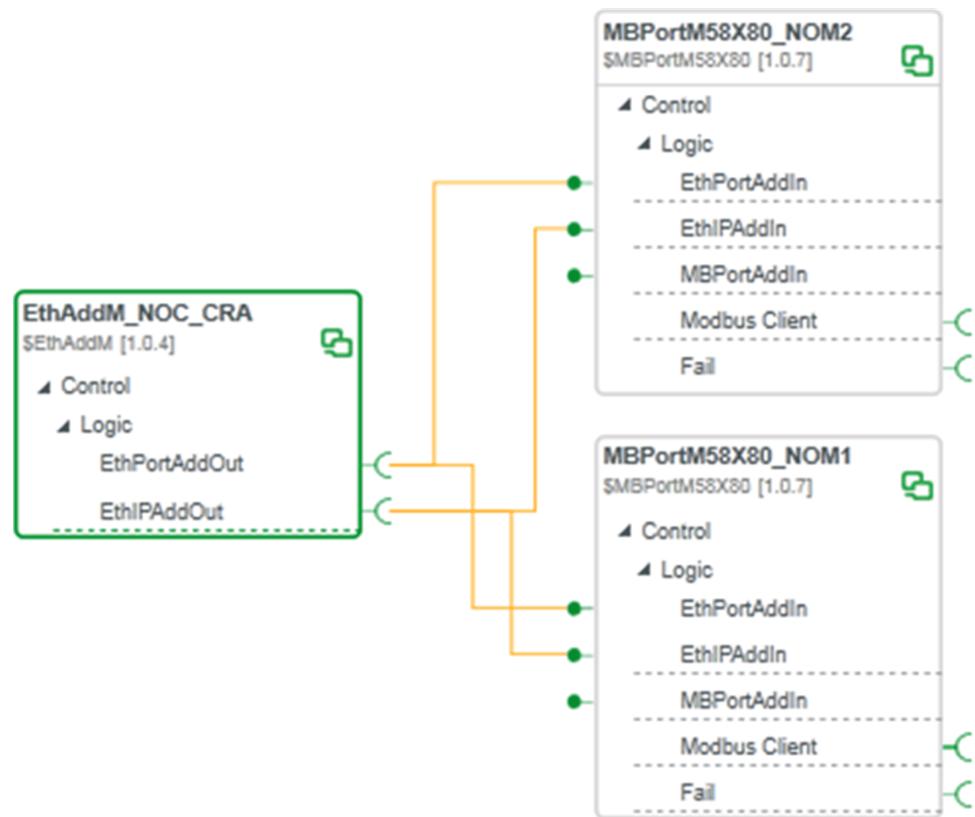


Proceed similarly with the communication port instance **MBPortM58X80_NOM2**, which communicates with the NOM communication module located in drop 2 by linking it to both interfaces of the intermediate template instance **EthAddM_NOC_CRA**.



The **EthAddM_NOC_CRA** instance passes the physical NOC port address as well as the IP address of the **BMECRA312** module to the second communication port instance.

The figure shows the existing links from the perspective of the intermediate template.



Control Facet Assignment

Facets of the instances are assigned to a Control Participant project in two separate FBD sections.

Folder C is assigned to *FBDSection_1*.

FBDSection_1 - Assignments											
Identifier	Container	Instance	Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation	
FBDSection_1	TesysTMB_10	\$TesysTMB	Valid		0 TesysTMB_10_TesysT	STesysTMB_UL	Control	0	Assigned	Generated	
FBDSection_1	TesysTMB_11	\$TesysTMB	Valid		0 TesysTMB_11_TesysT	STesysTMB_UL	Control	1	Assigned	Generated	
FBDSection_1	MBPortM58X80_NOM1	\$MBPortM58X80	Valid		0 MBPortM58X80_NOM1_MBPM58X80	\$MBPortM58X80_UL	Control	2	Assigned	Generated	
FBDSection_1	EthAddM_NOC_CRA	\$EthAddM	Valid		0 EthAddM_NOC_CRA_PA	SEthAddM_UL	Control	3	Assigned	Generated	
FBDSection_1	Motor_10	\$Motor	Valid		0 Motor_10_DEVCTL	SDEVCTL_UL	ControlMotor	4	Assigned	Generated	
FBDSection_1	Motor_10	\$Motor	Valid		0 Motor_10_CONDSUM	SCONDSUM_UL	ControlFailures	5	Assigned	Generated	
FBDSection_1	Motor_10	\$Motor	Valid		0 Motor_10_CONDSUM1	SCONDSUM1_UL	ControlInterlocks	6	Assigned	Generated	
FBDSection_1	Motor_11	\$Motor	Valid		0 Motor_11_DEVCTL	SDEVCTL_UL	ControlMotor	7	Assigned	Generated	
FBDSection_1	Motor_11	\$Motor	Valid		0 Motor_11_CONDSUM	SCONDSUM_UL	ControlFailures	8	Assigned	Generated	
FBDSection_1	Motor_11	\$Motor	Valid		0 Motor_11_CONDSUM1	SCONDSUM1_UL	ControlInterlocks	9	Assigned	Generated	

Folder C_2 is assigned to *FBDSection_2*.

FBDSection_2 - Assignments											
Identifier	Container	Instance	Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation	
FBDSection_2	ATV61MB_20	\$ATV61MB	Valid		0 ATV61MB_20_ATV	SATV61MB_UL	Control	0	Assigned	Generated	
FBDSection_2	ATV61MB_21	\$ATV61MB	Valid		0 ATV61MB_21_ATV	SATV61MB_UL	Control	1	Assigned	Generated	
FBDSection_2	MBPortM58X80_NOM2	\$MBPortM58X80	Valid		0 MBPortM58X80_NOM2_MBPM58X80	\$MBPortM58X80_UL	Control	2	Assigned	Generated	
FBDSection_2	Motor_20	\$Motor	Valid		0 Motor_20_DEVCTL	SDEVCTL_UL	ControlMotor	3	Assigned	Generated	
FBDSection_2	Motor_20	\$Motor	Valid		0 Motor_20_CONDSUM	SCONDSUM_UL	ControlFailures	4	Assigned	Generated	
FBDSection_2	Motor_20	\$Motor	Valid		0 Motor_20_CONDSUM1	SCONDSUM1_UL	ControlInterlocks	5	Assigned	Generated	
FBDSection_2	Motor_21	\$Motor	Valid		0 Motor_21_DEVCTL	SDEVCTL_UL	ControlMotor	6	Assigned	Generated	
FBDSection_2	Motor_21	\$Motor	Valid		0 Motor_21_CONDSUM	SCONDSUM_UL	ControlFailures	7	Assigned	Generated	
FBDSection_2	Motor_21	\$Motor	Valid		0 Motor_21_CONDSUM1	SCONDSUM1_UL	ControlInterlocks	8	Assigned	Generated	

Generation and Service Mapping

The Control Participant project is generated.

An executable is created in the Control Participant and mapped to the M580 controller.

Hardware Mapping

The **Hardware Mapping Editor** shows the following interfaces.

(Lines have been drawn to show the mappings that you need to do.)

Available for mapping		HWTemplate		HWMappingInterface		HWInterfaceType		Available for mapping		AppInstance		AppFacet		AppTemplate		AppMappingInterf	
HVInstance								AppInstance									
0.Pri.local.0.D.R.0.RMEP584040	\$M580CPUIECRP	MBTCPEthM580	EMPortCHM	SMEthernetPortCHHO				ATV61MB_20		ATV61MB_20_ATV		SATV61MB		MBAddress			
0.Pri.local.0.D.R.0.2.BMENOC0301.2	\$MNOCClient	MBTCPEthEMPortCHM	SMEthernetPortCHHO					ATV61MB_21		ATV61MB_21_ATV		SATV61MB		MBAddress			
2.EtherIO.1.D.R.0.R.0.BMERCRA31210.2	\$CRAETH	.CRAIPAddress		\$EthernetAddressHO				EthAddM_NOC_CRA		EthAddM_NOC_CRA_PA		SEthAddM		EMPortCHM			
2.EtherIO.1.D.R.0.R.0.1.BMXNOMD200.3	\$HWNOMMB	MB_M340_M580_Master_CH0.ModbusPortCH	\$ModbusPortCHHO					EthAddM_NOC_CRA		EthAddM_NOC_CRA_PA		SEthAddM		CRAIPAddress			
2.EtherIO.1.D.R.0.R.0.1.BMXNOMD200.3	\$HWNOMMB	MBControllerSlave_CH1.ModbusAddress	\$ModbusAddressHO					MBPortM58x80_NOM1		MBPortM58x80_NOM1_MBPM58x80		SMBPortM58x80		MBPortCHM			
2.EtherIO.1.D.R.0.R.0.2.BMXNOMD200.3	\$HWNOMMB	MB_M340_M580_Master_CH0.ModbusPortCH	\$ModbusPortCHHO					MBPortM58x80_NOM2		MBPortM58x80_NOM2_MBPM58x80		SMBPortM58x80		MBPortCHM			
2.EtherIO.1.D.R.0.R.0.2.BMXNOMD200.3	\$HWNOMMB	MBControllerSlave_CH1.ModbusAddress	\$ModbusAddressHO					TesysTMB_10		TesysTMB_10_Tesyst		STesysTMB		MBAddress			
								TesysTMB_11		TesysTMB_11_Tesyst		STesysTMB		MBAddress			

NOTE: You do not need to map **MBAdress** interfaces (for example, of facet **TesysTMB_10_Tesyst**).

Completing the System Engineering Life Cycle

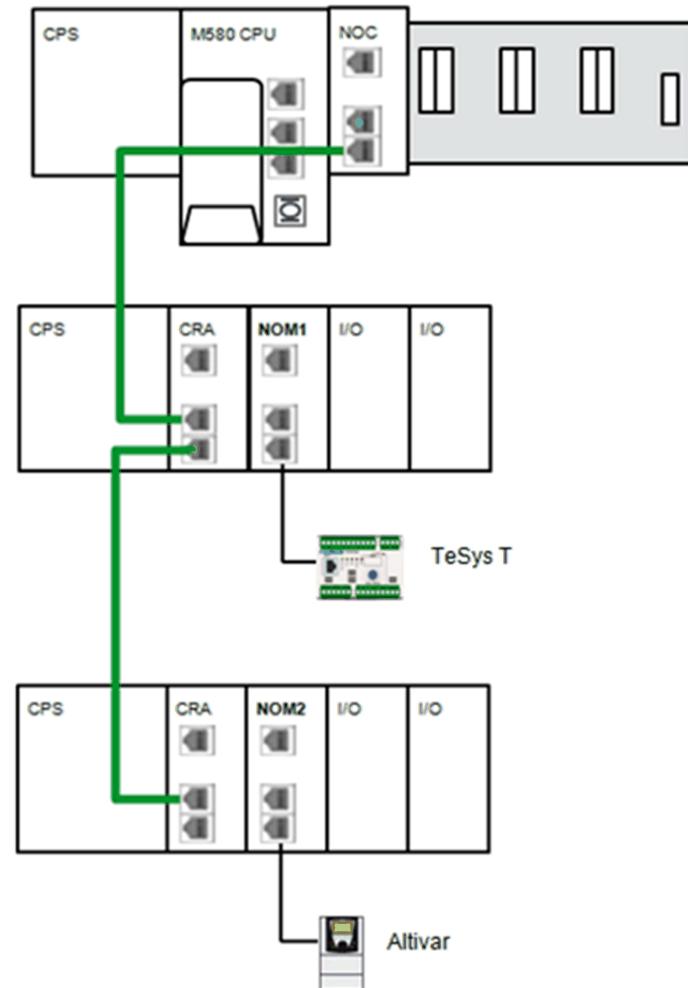
Complete the creation of the system by performing the other steps of the system engineering life cycle.

Example D

Example Description

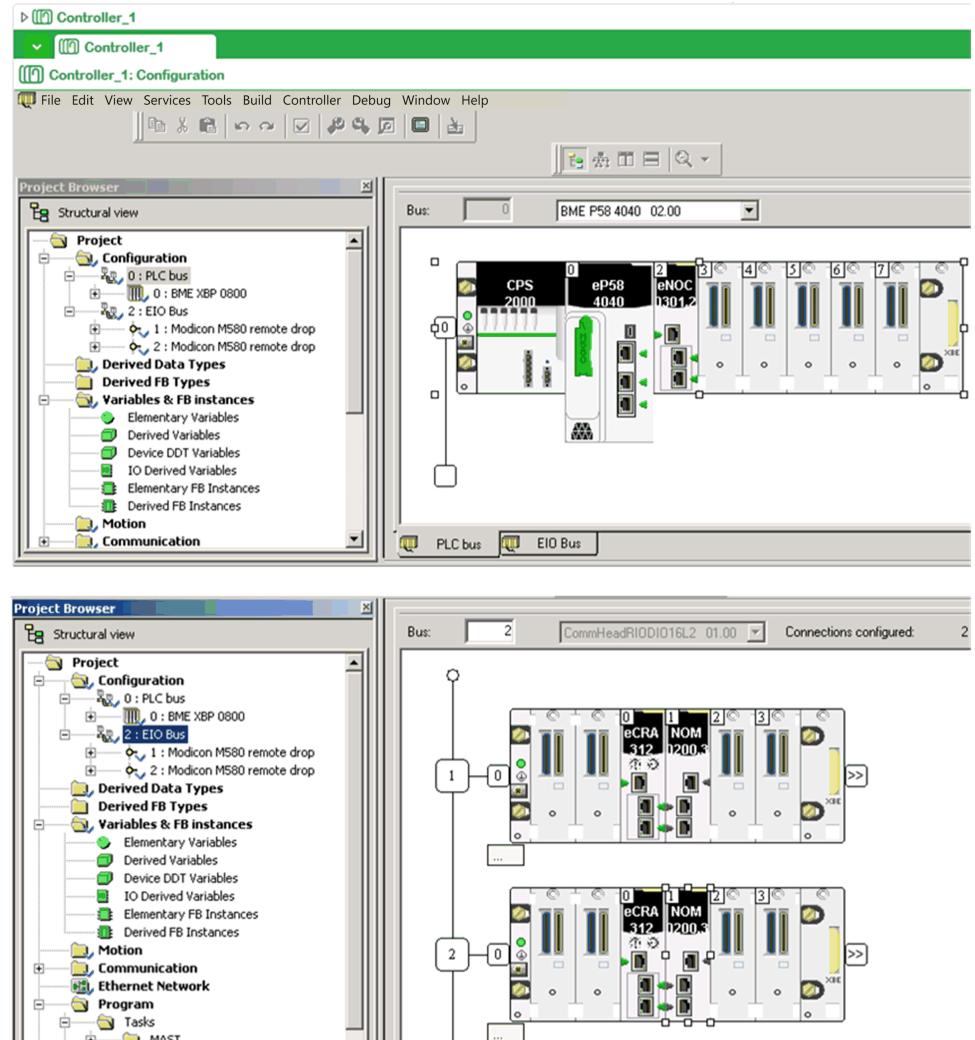
The table indicates the specifics of this example in terms of drop configuration and application objects used.

M580 remote drop configuration	Application objects
2 drops with 1 BMXNOM0200 communication module in each.	For each drop, 1 device communicating through 1 communication port.



Configuration Stage

The following figure shows the controller configuration with the drop and the communication module.



Instantiation Stage

The following Instances are required to implement communication at the application level with two devices (a TeSys T motor controller and an Altivar variable speed drive) by using Modbus serial.

The *MBPortM58X80_x* Modbus serial communication port instance are to be used when the NOM communication module is located in the drop of an M580 controller. In this example, two instances are created, one for each physical communication port used.

Instances are grouped in two folders for clarity, one for each communication port instance.

The screenshot shows two tables of communication port instances in the Asset Workspace Editor:

Identifier	Template	Version	Data	Link	Assigned State
EthAddM_NOC	SEthAddM	1.0.4	Valid	Valid	Unassigned
MBPortM58X80_NOM1	SMBPortM58X80	1.0.7	Valid	Valid	Unassigned
Motor_1	SMotor	2.5.3	Valid	Valid	Unassigned
TesysTMB_1	STesysTMB	2.1.6	Valid	Invalid	Unassigned

Identifier	Template	Version	Data	Link	Assigned State
ATV61MB_2	\$ATV61MB	1.2.5	Valid	Invalid	Unassigned
MBPortM58X80_NOM2	SMBPortM58X80	1.0.7	Valid	Valid	Unassigned
Motor_2	SMotor	2.5.3	Valid	Valid	Unassigned

These steps result in the following links.

The communication port instance (MBPortM58X80_NOM1) passes the following data to the process instances:

- Port address of the NOC communication module
- IP address of the CRA module of the drop
- Port address of the NOM communication module

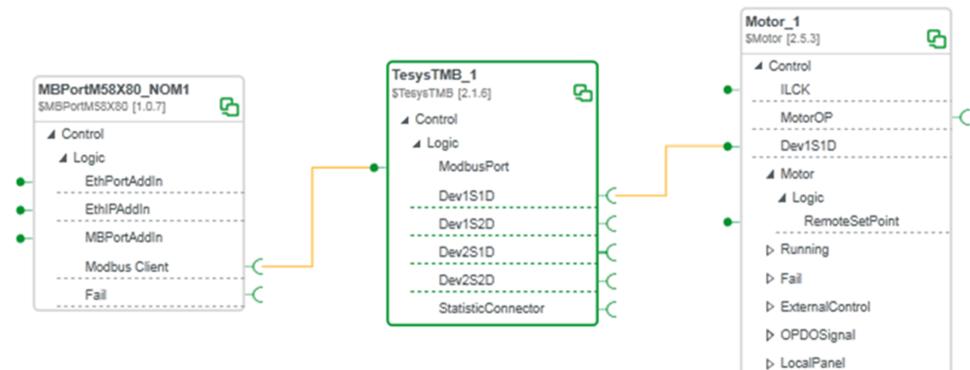
Intermediate Template Instances

Only intermediate template \$EthAddM_NOC_CRA is used because more than one communication port instance is used. It is common to both communication port instances. Intermediate template \$MBAddM is not used because on each NOM port, only one communication port instance is used.

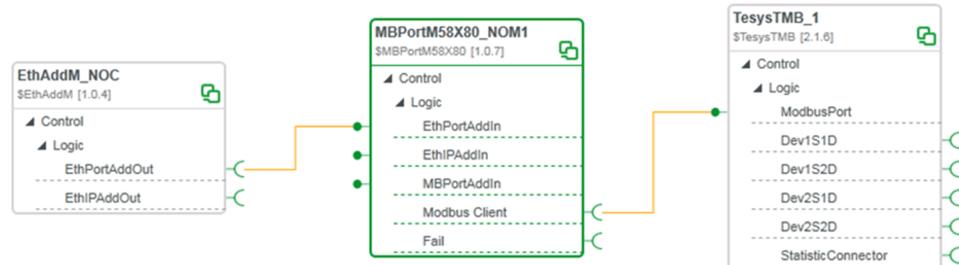
Application Interface Links

The following links are created in the **Asset Workspace Editor** for the TeSys T devices.

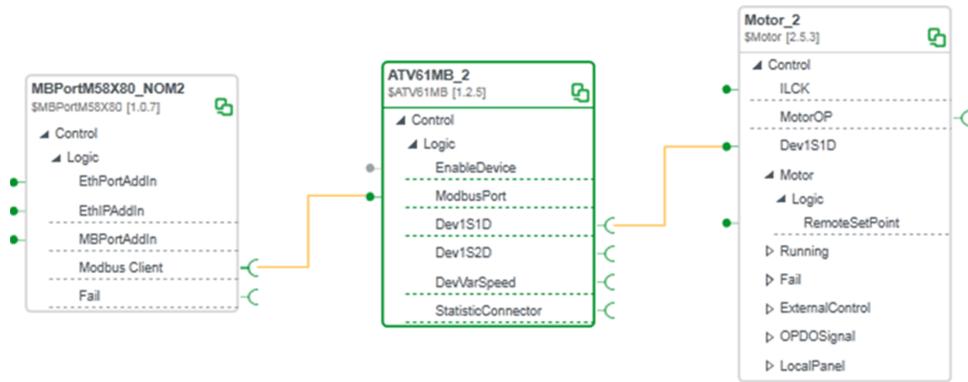
The process instance (*Motor_1*), the device instance (*TesysTMB_1*), and the communication port instance (*MBPortM58X80_NOM1*) are linked.



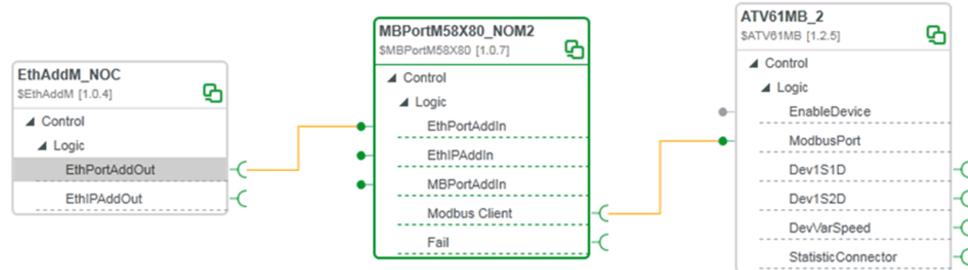
Then, link the *MBPortM58X80_NOM1* instance to the **EthPortAddOut** interface of the *EthAddM_NOC* intermediate template instance.



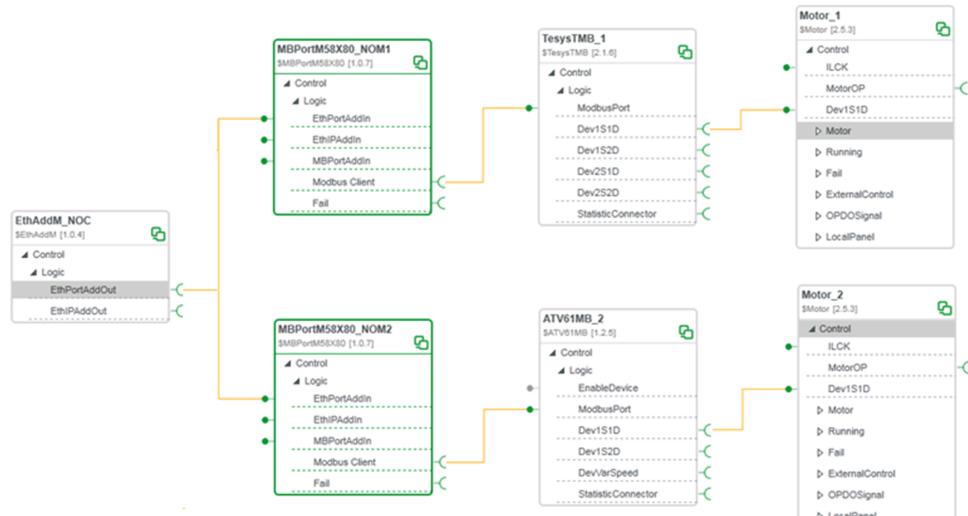
Similarly, the process instance (*Motor_2*), the device instance (*ATV61MB_2*), and the communication port instance (*MBPortM58X80_NOM2*) are linked.



Then, link the *MBPortM58X80_NOM2* instance to the **EthPortAddOut** interface of the *EthAddM_NOC* intermediate template instance.



These steps result in the following links.



The communication port instances (*MBPortM58X80_x*) pass the following data to the process instances:

- IP address of the CRA module of each drop
- Port address of the NOM communication module of each drop

The intermediate template instance (*EthAddM_NOC*) passes the following data to the process instances:

- Port address of the NOC communication module in the local rack

Control Facet Assignment

Facets of the instances are assigned to a Control Participant project in two separate FBD sections.

Folder *D* is assigned to *FBDSection_1*.

FBDSection_1 - Assignments											
Identifier	Container	Instance	Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation	
FBDSection_1	TesysTMB_1	TesysTMB		Valid	\$TesysTMB_UL	\$TesysTMB_UL	Control	0	Assigned	Generated	
FBDSection_1		MBPortM58X80_NOM1	MBPortM58X80	Valid	\$MBPortM58X80_UL	\$MBPortM58X80_UL	Control	1	Assigned	Generated	
FBDSection_1	EthAddM_NOC	\$EthAddM		Valid	\$EthAddM_NOC_PA	\$EthAddM_UL	Control	2	Assigned	Generated	
FBDSection_1	Motor_1	\$Motor		Valid	\$Motor_1_DEVCTL	\$DEVCTL_UL	Control/Motor	3	Assigned	Generated	
FBDSection_1	Motor_1	\$Motor		Valid	\$Motor_1_CONDSUM	\$CONDSUM_UL	Control/Failures	4	Assigned	Generated	
FBDSection_1	Motor_1	\$Motor		Valid	\$Motor_1_CONDSUM1	\$CONDSUM1_UL	Control/Interlocks	5	Assigned	Generated	

Folder *D_2* is assigned to *FBDSection_2*.

FBDSection_2 - Assignments											
Identifier	Container	Instance	Instance Template	State	Facet	Facet Template	Path	Order	Assignment	Generation	
FBDSection_2	ATV61MB_2	ATV61MB		Valid	\$ATV61MB_2_ATV	\$ATV61MB_UL	Control	0	Assigned	Generated	
FBDSection_2		MBPortM58X80_NOM2	MBPortM58X80	Valid	\$MBPortM58X80_NOM2_MBPM58X80	\$MBPortM58X80_UL	Control	1	Assigned	Generated	
FBDSection_2	Motor_2	\$Motor		Valid	\$Motor_2_DEVCTL	\$DEVCTL_UL	Control/Motor	2	Assigned	Generated	
FBDSection_2	Motor_2	\$Motor		Valid	\$Motor_2_CONDSUM	\$CONDSUM_UL	Control/Failures	3	Assigned	Generated	
FBDSection_2	Motor_2	\$Motor		Valid	\$Motor_2_CONDSUM1	\$CONDSUM1_UL	Control/Interlocks	4	Assigned	Generated	

Generation and Service Mapping

The Control Participant project is generated.

An executable is created in the Control Participant and mapped to the M580 controller.

Hardware Mapping

The **Hardware Mapping Editor** shows the following interfaces.

(Lines have been drawn to show the mappings that you need to do.)

mapping		HWInstance	HWTemplate	HWMappingInterface	HWInterfaceType	Available for mapping	AppInstance	AppFacet	AppTemplate	AppMappingInfo
PriLocal	0.0.R.0.BMEPE584040	\$M580CPU1EhCRP	MBTCP1eIM580_EhPortCHM	\$MEthernetPortCHCHO	ATV61MB_2_ATV	ATV61MB_2_ATV	SATV61MB	MBAddress		
PriLocal	0.0.R.0.02.BMEENCC0301.2	\$MNOCIOSClient	MBTCP1eIM580_EhPortCHM	\$MEthernetPortCHCHO-----	EthAddM_NOC	EthAddM_NOC_PA	SEIAAddM	EMPortCHM		
EhRIO	1.0.R.0.BMECR3121.2	SCRA1Eth		\$EthernetAddress/HO	EhAddM_NOC	EthAddM_NOC_PA	SEIAAddM	CRAIAddress		
EhRIO	1.0.D.0.01.BMKNM0200.3	SHWNWOMB	MB_M340_MS58_Master_CH0.MModbusPortCH	\$ModbusPortCHHO	MBPortM58X80_NOM1	MBPortM58X80_NOM1_MBPM58X80	MBPortM58X80	CRAIAddress		
EhRIO	1.0.D.0.01.BMKNM0200.3	SHWNWOMB	MBControllerSlave_CH1.ModbusAddress	\$ModbusAddress/HO	MBPortM58X80_NOM1	MBPortM58X80_NOM1_MBPM58X80	MBPortM58X80	MBPortCHM		
EhRIO	2.0.D.0.0.BMECR3121.2	SCRA1Eth		\$EthernetAddress/HO	MBPortM58X80_NOM2	MBPortM58X80_NOM2_MBPM58X80	MBPortM58X80	CRAIAddress		
EhRIO	2.0.D.0.01.BMKNM0200.3	SHWNWOMB	MB_M340_MS58_Master_CH0.MModbusPortCH	\$ModbusPortCHHO	MBPortM58X80_NOM2	MBPortM58X80_NOM2_MBPM58X80	MBPortM58X80	MBPortCHM		
EhRIO	2.0.D.0.01.BMKNM0200.3	SHWNWOMB	MBControllerSlave_CH1.ModbusAddress	\$ModbusAddress/HO	TesysTMB_1	TesysTMB_1_TesysT	STesysTMB	MBAddress		

NOTE: You do not need to map **MBAddress** interfaces (for example, of facet *TesysTMB_1_TesysT*).

Completing the System Engineering Life Cycle

Complete the creation of the system by performing the other steps of the system engineering life cycle.

Monitoring Modbus Device Data Directly from Supervision

Monitoring Modbus Device Data Directly from Supervision

Overview

This topic provides specific procedures to help you implement connectivity between Modbus TCP and Modbus serial devices and the Supervision infrastructure that is defined in the system.

For Modbus TCP devices, you can use the OPC or OPC UA driver.

This allows, for example, monitoring additional Modbus device variables without adding load to the controller.

The implementation is possible for the following topological entities:

- Modbus TCP I/O devices.
- STB islands.
- PROFIBUS remote masters (PRMs).
- Third-party I/O devices modeled by using the generic topological device template.

It also allows connecting Modbus serial devices by using the generic Modbus TCP I/O device acting as gateway.

The implementation supplements the engineering effort to set up communication between the controller and Modbus devices and to operate these devices from the runtime. However, it does not use the application control module templates that come into play in the usual system engineering life cycle but requires creating your own application objects by using Supervision data facet templates, page 841.

By following the steps described in this topic, the software generates Supervision tags for variables to be monitored based on the application objects that you have created.

Implementing Monitoring of Modbus TCP Device Data

The table describes the specific steps that you need to perform to be able to monitor data of Modbus TCP devices directly from the Supervision infrastructure. It is assumed that the required topological entities, Control and Supervision Participant projects, the application instances, assignments, and mappings to implement communication with Modbus TCP devices, page 779 already exist.

Step	Description	Stage
1	<p>Instantiate a Supervision data facet template for each variable of a Modbus TCP device that you want to monitor from Supervision. You can use the following templates:</p> <ul style="list-style-type: none"> • <code>\$VariableTag_CD</code> • <code>\$AdvancedAlarm_CD</code> • <code>\$TrendTag_CD</code> <p>Configure the identifiers of each instance so that the relationship with the corresponding Modbus TCP device can be easily seen.</p> <p>NOTE: Select Facets as filter in the Browser pane.</p>	Instantiation
2	<p>Configure instances of <code>\$VariableTag_CD</code> as follows:</p> <ul style="list-style-type: none"> • Addr parameter (<i>Configuration</i> category): <ul style="list-style-type: none"> ◦ For the OPC driver: Address of the variable by using the format <code>Alias!%MW<i></code>. For example, <code>ATV61IOD_IODEVICE_1 ! %MW3233</code>. ◦ For the OPC UA driver: Name of the variable in the symbol table. • Type parameter (<i>Configuration</i> category): Data type of the variable. For example, <code>UINT</code>. <p>NOTE: Alternatively, for OPC, you can specify the address for the device by adding an appropriate entry in the <code>citect.ini</code> file. For details, refer to <i>OPC Access Path Parameters</i> in the <i>OPC Driver</i> topic of the <i>Driver Reference Help</i>, which you can open from the Windows® Start menu or from within the Supervision Participant (Topology/I/O Devices).</p>	

Step	Description	Stage
3	If you are using trend tags, open the Asset Workspace Editor and link the trend tag instance to the variable tag instance so that the instance status becomes valid.	
4	Create additional tag containers, page 340; one for each Modbus TCP device that you want to monitor from Supervision and configure Protocol as either OPC or OPC UA.	Project Definition (Supervision)
5	Create a dedicated communication board, page 302 and configure Type as either OPC or OPC UA.	
6	Create a communication port for each Modbus TCP device.	
7	Create an I/O device, page 299 for each tag container and configure the following parameters: <ul style="list-style-type: none"> • Address: <ul style="list-style-type: none"> ◦ For the OPC driver: Enter Schneider-Aut.OFS, which corresponds to the Prog ID of the OFS server. ◦ For the OPC UA driver: The address parameter is not used. • TagContainers: Associate each I/O device to a separate tag container. • Ports 	
8	Assign the Supervision facets related to each device to the corresponding tag container. For example, if you have created two Modbus TCP devices (<i>EATV71HW_1</i> and <i>ETesystHW_1</i>) and for each device, you have created one variable tag instance (<i>ATV71_Var1</i> and <i>TesysT_Var1</i>), then you need to assign: <ul style="list-style-type: none"> • <i>ATV71_Var1</i> to <i>TagContainer_ATV71</i> • <i>TesysT_Var1</i> to <i>TagContainer_TesysT</i> 	Assignment
9	Generate the project.	Generation
10	In the Communication Mapping tab, page 624, perform the following additional mappings for each I/O device that you have created for a Modbus TCP device: <ul style="list-style-type: none"> • Select the corresponding topological entity (the Modbus TCP device). • Select its communication service (MBTCPServiceServer or device-specific service). • Verify the Modbus Unit ID value for the device and modify it as needed (refer to the user manual of the device). 	Mapping (communication mapping for Supervision executables)
11	When using the OPC UA driver, you must create a symbol table (.csv) where the variable name and data type correspond to the values that you have entered for the Addr and Type parameters (see step 2). For details, refer to <i>Modbus Device Address Space</i> in the help of EcoStruxure OPC UA Server Expert.	EcoStruxure OPC UA Server Expert Configuration Tool
12	Associate the symbol table file to the device alias by using the Symbol table file browse button. Verify that Symbol table is selected under Preload settings .	

To complete the other steps to engineer the system, follow the system engineering life cycle described in this manual.

Implementing Monitoring of Modbus Serial Device Data

The table describes the specific steps that you need to perform to be able to monitor data of Modbus serial devices directly from the Supervision infrastructure by using the OPC driver. It is assumed that the required topological entities, Control and Supervision Participant projects, the application instances, assignments, and mappings to implement communication with devices by using Modbus explicit messaging, page 798 already exist.

Step	Description	Stage
1	Create a Modbus TCP device entity by using the <i>\$EGenericDeviceHW</i> template, page 531. The entity acts as a gateway.	Configuration
2	Configure the <i>IPAddress</i> and <i>SubnetMask</i> parameters of the gateway entity.	

Step	Description	Stage
	If you are already using an application template of the Modbus serial device (for example, an instance of \$ATV71MB) and have linked it to an instance of the \$EGtwMB gateway application template, enter the GatewayAdd parameter value of the gateway instance. Otherwise, enter the IP address of the gateway. NOTE: Enter values for the read/write length and address parameters only if you are going to implement I/O scanning with the gateway entity.	
3	Connect the gateway entity to the same Ethernet network to which the station node that is mapped to the I/O server is connected.	
4	Instantiate a Supervision data facet template for each variable of a Modbus serial device that you want to monitor from Supervision. You can use the following templates: <ul style="list-style-type: none"> • \$VariableTag_CD • \$AdvancedAlarm_CD • \$TrendTag_CD Configure the identifiers of each instance so that the relationship with the corresponding Modbus serial device can be easily seen. NOTE: Select Facets as filter in the Browser pane.	Instantiation
5	If you are using trend tags, open the Asset Workspace Editor and link the trend tag instance to the variable tag instance so that the instance status becomes valid.	
6	Configure instances of \$VariableTag_CD as follows: <ul style="list-style-type: none"> • <i>Addr</i> parameter (<i>Configuration</i> category): Address of the variable by using the format <i>Alias!%MW<i></i>. For example, ATV61IOD_IODEVICE_1!%MW3233. • <i>Type</i> parameter (<i>Configuration</i> category): Data type of the variable. For example, <i>UINT</i> for <i>ATV61IOD_IODEVICE_1%MW3233</i>. NOTE: Alternatively, you can specify the address for the device by adding an appropriate entry in the <i>citect.ini</i> file. For details, refer to <i>OPC Access Path Parameters</i> in the <i>OPC Driver</i> topic of the <i>Driver Reference Help</i> , which you can open from the Windows® Start menu or from within the Supervision Participant (Topology\I/O Devices).	
7	Create additional tag containers, page 340; one for each Modbus serial device that you want to monitor from Supervision. Configure Protocol as OPC .	Project Definition (Supervision)
8	Create a dedicated communication board, page 302 and configure Type as OPC .	
9	Create a communication port for each Modbus TCP device.	
10	Create an I/O device, page 299 for each tag container and configure the following parameters: <ul style="list-style-type: none"> • <i>Address</i>: Enter <i>Schneider-Aut.OFS</i>, which corresponds to the Prog ID of the OFS server. • <i>TagContainers</i>: Associate each I/O device to a separate tag container. • <i>Ports</i> 	
11	Assign the Supervision facets related to each device to the corresponding tag container. For example, if you want to monitor two Modbus serial devices and for each device, you have create one variable tag instance (<i>ATV71MB_Var1</i> and <i>TesysTMB_Var1</i>), then you need to assign: <ul style="list-style-type: none"> • <i>ATV71MB_Var1</i> to <i>TagContainer_ATV71MB</i> • <i>TesysT_Var1</i> to <i>TagContainer_TesysT</i> 	Assignment
12	Generate the project.	Generation

Step	Description	Stage
13	<p>In the Communication Mapping tab, page 624, perform the following additional mappings for the first I/O device that you have created for a Modbus serial device:</p> <ul style="list-style-type: none"> • Select the gateway as topological entity (instance of <code>\$EGenericDeviceHW</code>). • Select its communication service (MBTCPServiceServer). • Enter the Modbus Unit ID value for the device. If you have already configured an application template of the device (for example, an instance of <code>\$ATV71MB</code>), enter the ModbusAddress parameter value of the instance. Otherwise, enter a unique serial slave ID for the device. 	Mapping (communication mapping for Supervision executables)
14	Repeat step 13 for each additional Modbus serial device by selecting the same gateway entity but entering a new Modbus Unit ID value.	

To complete the other steps to implement the system, follow the system engineering life cycle described in this manual.

Communication with Devices by Using a PROFIBUS Master Module

Communication with Devices by Using a PROFIBUS Primary Module

Overview

The topology supports the PMEPXM0100 PROFIBUS primary module. The configuration of the module is performed partially in the **Topology Explorer** and completed by using the external ProSoft Configurator for Modicon. Additional steps need to be performed in various stages of the system engineering life cycle.

Prerequisites

- Read and understand the *PMEPX0100 ProSoft User Manual*.
- Install ProSoft Configurator for Modicon. On a computer on which the engineering client is installed, you can use the EcoStruxure Process Expert installer.

NOTE: To download the latest version of the tool and the manual, visit www.se.com/en/download.

Workflow

The table describes the workflow to integrate a PMEPXM0100 PROFIBUS primary module and secondary devices into a system by using a standalone or redundant M580 controller.

Step	Description	Stage
1	Add a PMEPXM0100 PROFIBUS module to an M580 standalone or redundant controller. NOTE: For details, including steps that you need to perform when using a redundant controller, refer to the topic describing the configuration of the PMEPXM0100 PROFIBUS module. (see EcoStruxure™ Process Expert, Control Participant Services, User Guide)	Configuration
2	If you have templates of PROFIBUS secondary devices, create instances thereof and configure them as needed. NOTE: If you want to use device templates of a Schneider Electric library (for example, \$ATV71PB or \$TeSysTPB), refer to the specific procedure describing how to use them.	Instantiation
3	Assign the facets of application objects to the Control project and generate the project.	Assignment and Generation
4	With the help of the ProSoft Configurator for Modicon, configure the PMEPXM0100 module and secondary devices by following the steps described in the <i>PMEPX0100 ProSoft User Manual</i> . NOTE: In the Export EcoStruxure Control Expert Export Options dialog box select <i>Enhanced for I/O Scan Mode</i> .	External
5	Refine the Control project and create the device DDT variable, page 445 of the PMEPXM0100 module. NOTE: If you are using a redundant controller, create both device DDT variables.	Refinement
6	Import the exported project file by following the steps described in the <i>PMEPX0100 ProSoft User Manual</i> .	Refinement
7	Analyze the Control project.	
8	Build and deploy the Control project to the controller.	Build and Deployment
9	With the help of the ProSoft Configurator for Modicon, download the primary and secondary device configuration to the module by following the steps described in the <i>PMEPX0100 ProSoft User Manual</i> .	External
10	Refine the deployed Control project online and follow the steps described in the <i>PMEPX0100 ProSoft User Manual</i> to set the operating mode of the PMEPXM0100 module.	Execution

To complete the other steps to engineer the system, follow the system engineering life cycle described in this manual.

Managing, Modifying, and Creating Templates

What's in This Part

Global Templates Library	838
Global Templates Editors	852
Managing Global Templates.....	880

Overview

Creating, modifying, updating, replacing, or duplicating templates may affect the function of these templates and/or systems, and must be performed by qualified personnel.

Refer also to the topic containing information about template design good practices and key aspects (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

WARNING

LOSS OF CONTROL

After creating, updating, or modifying templates:

- Verify that templates generate correct addresses when their mapping interfaces are mapped by using the **Hardware Mapping Editor**.
- Perform a Failure Mode and Effects Analysis (FMEA) of the system that contains new, updated, or modified templates and apply preventive, detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines¹.
- Test each implementation of a template for proper operation before placing it into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* and to NEMA ICS 7.1 (latest edition), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* or their equivalent governing your particular location.

Global Templates Library

What's in This Chapter

Global Templates Explorer	839
Global Templates Folders	845
Global Templates	847
Managing the Control Constituent Password	849

Overview

This chapter describes the Global Templates library, the tools that are available to manage it, and context menu commands.

Global Templates Explorer

Overview

The **Global Templates Explorer** shows the templates of the Global Templates library, which regroups the Schneider Electric libraries for EcoStruxure Process Expert that are installed (for example, the Foundation Library, the General Purpose Library) as well as templates that you create and/or import.

These templates are displayed in a folder structure and can be of the following types:

- Interface models
- Facet templates
- Composite templates

The **Global Templates Explorer** allows you to perform the following actions on Global Templates:

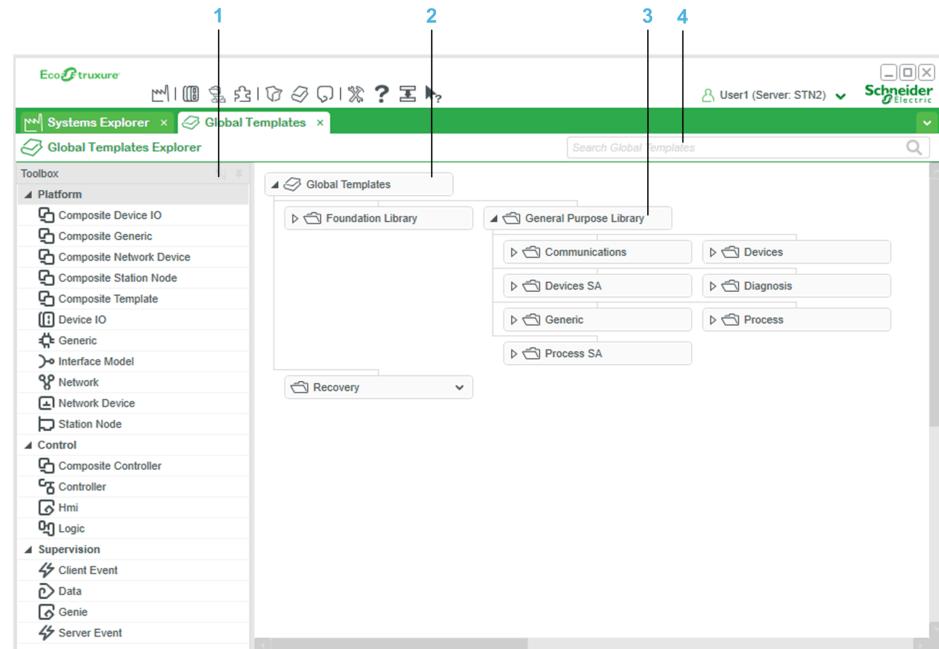
- Browse Global Templates.
- Export and import templates, page 881.
- Open template editors, page 852.
- Create copies of templates, page 885.
- Create (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) your own folders, page 845 and templates.
- Update and replace templates, page 898.

Opening the Global Templates Explorer

Open the **Global Templates Explorer** by clicking the **Global Templates** button in the toolbar, page 101.

Global Templates Explorer

The following figure shows an example of the **Global Templates Explorer** window. The content of the folder structure may vary depending on the libraries that are installed.



Item	Description
1	Toolbox pane, page 841. The pin button allows you to minimize the Toolbox pane to the left edge of the screen. Move the pointer above the Toolbox tab to display the pane. Click the pin button again to restore the pane.
2	Global Templates library root folder.
3	Tree view of the installed libraries. You can modify the layout, page 843 of the tree view and open parts of it in new windows, page 843.
4	Search field, page 840.

Global Templates Explorer Actions

Right-clicking the white space of the **Global Templates Explorer** opens a context menu containing commands, page 846 that apply to the Global Template library except for the **Settings** command, which lets you modify the folder layout, page 843 inside the workspace.

Global Templates Naming Convention

The Schneider Electric Global Templates follow a naming convention:

\$NAME_XX

Where

- \$ indicates that the template is provided by Schneider Electric.
- NAME is the name of the template
- _XX indicates the type of the template. Control module templates have no suffix.

NOTE: Some templates provided by Schneider Electric may not have the \$ prefix to indicate that you can modify them to meet your specific requirements.

The table describes the suffixes that are used to identify types of Global Templates.

Suffix	Description
_UL	Control logic facet template.
_UH	Control HMI facet template.
_UC	Control logic and HMI composite template.
_CD	Supervision data composite and facet templates.
_CG	Supervision genie composite and facet templates.
_CS	Supervision composite template.
_CR	Supervision server composite template.
_CC	Supervision client composite template.

Searching Global Templates

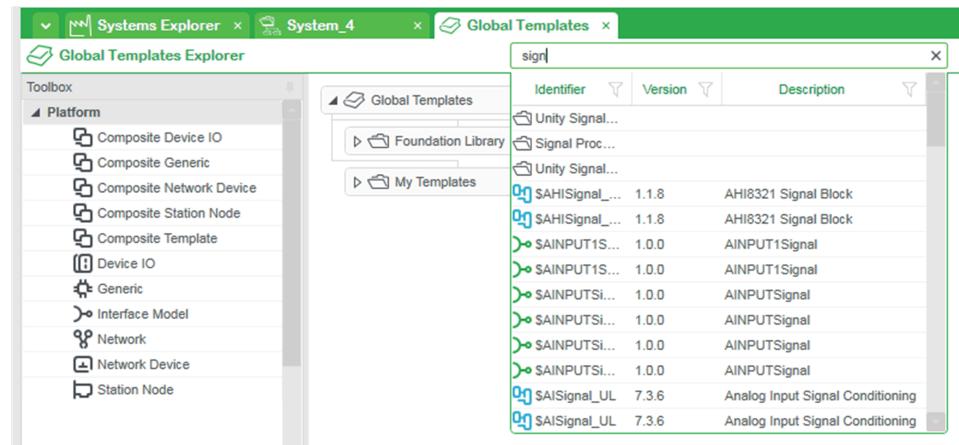
The search field of the **Global Templates Explorer** lets you find templates and folders by typing the entire or part of their identifier. You do not need to use wildcards. The search is not case-sensitive. Results are displayed as you type in a pane with columns. Your last three matching search keywords are stored and appear in a keyword menu. The stored keyword entries persist even after the system server is restarted.

For example, va returns templates and folders that contain va in their identifier, such as templates \$Valve, \$HValve_CD, and folder Advantys.

By default, results are displayed in ascending alphabetical order with folders shown first, templates next.

The results pane provides basic information on templates and remains open until you close it by clicking the close button. A tooltip provides additional information.

The following figure shows an example of the search results that can be shown when you enter `sign` in the search field.



Right-click a search result in the results pane to open a context menu with the following commands.

Command	Description
Navigate	<ul style="list-style-type: none"> For a template: Opens/expands the containing folder and highlights the instance. For a folder: Expands the folder structure and selects the folder. <p>You can also double-click the item.</p> <p>NOTE: If the template is hidden because a filter is applied, the template is not shown but the border of the folder containing the instance flashes briefly.</p>
Open	<p>The command is available for folders only.</p> <p>Opens a new Global Templates Explorer workspace and displays only the folder.</p> <p>NOTE: Searches that you perform by using the search field of this workspace are limited to the contents of the folder.</p>
Inspect	These commands are available for templates only.
Read-Only	For a description of these commands, refer to the topic describing Global Templates actions, page 848.
Edit	

You can sort and filter, page 118 search results.

NOTE: Search results are not updated automatically if templates or folders were created, modified, or deleted in the meantime. You need to start over the search by clicking the search field and pressing **Enter**.

Global Templates Toolbox

The **Toolbox** contains the following base Global Templates, which you can use to create new templates or modify, page 852 existing ones:

- Interface models
- Facet templates
- Composite templates

Templates are grouped in the following categories:

- Platform:** Templates and interface models that are used at the platform level. These templates can reference components of this category and/or of the other two categories.
- Control:** Facet templates that encapsulate constituents of the Control Participant. You can reference them inside composite templates.

- **Supervision:** Facet templates that encapsulate constituents of the Supervision Participant. You can reference them inside composite templates.

The table describes the items that are in the **Platform** category.

Item	Description
Composite Template	Generic template that you can use to: <ul style="list-style-type: none"> • Reference facet templates and other composite templates. • Build control module templates.
Composite Device IO and Device IO (facet templates)	Topological templates that you can use to model devices and their decentralized I/O modules in the topology of the system: <ul style="list-style-type: none"> • Devices (energy and motor devices, instrumentation...) • STB island I/O modules NOTE: You can use this type of template to model third-party devices in the topology of a system if they will be connected to the network of such a system as nodes.
Composite Generic and Generic (facet templates)	Topological templates that you can use to model unknown devices in a generic manner.
Interface Model	Object that you can define to create interfaces, which you can use to: <ul style="list-style-type: none"> • Link to each other facet and/or composite templates referenced by a control module template during template creation. • Expose connectors that allow linking instances to each other during instantiation.
Network	Topological template that you can use to model the following types of networks that are used in the topology of a system: <ul style="list-style-type: none"> • Ethernet • Modbus • CANopen • PROFIBUS DP • RIO S908 • RIO Ethernet • Local bus of topological entities
Composite Network Device and Network Device (facet templates)	Topological templates that you can use to model network devices that are used in the topology of a system (for example, switches, routers, gateways...).
Composite Station Node and Station Node (facet templates)	Topological templates that you can use to model a workstation in the topology of a system. Such a workstation instance supports the following hardware and software subcomponents: <ul style="list-style-type: none"> • Network interface card (NIC) • Supervision software • OPC Factory Server and OPC UA Server Expert software • Control Expert software • Platform and platform backup servers

The table describes the templates that are in the **Control** category.

Item	Description
Composite Controller and Controller (facet templates)	Topological templates that you can use to model controller hardware modules (CPUs, power supplies, network cards, I/O modules, and so on) of supported platforms (see EcoStruxure™ Process Expert, Control Participant Services, User Guide). NOTE: Do not use this type of template to model distributed I/O modules such as STB island I/O modules, for example in the topology of a system. Use a Platform DeviceIO template instead.
HMI	Application Control facet template that you can use to model one operator screen.
Logic	Application Control facet template that you can use to encapsulate Control constituents to implement a specific function. For example, you can encapsulate a DFB and one or more variables to manage on/off motor functions.

The table describes the templates that are in the **Supervision** category:

Item	Description
Client Event	Application client script facet template that you can use to create an event to be executed during runtime in the context of the Supervision client.
Data	Application Supervision facet template that you can use to model tags and user interface messages required for runtime data acquisition.
Genie	Application Supervision facet template that you can use to encapsulate an animated graphic (Genie), which is contained in an included project, page 496 and which can display data of the associated control module during runtime.
Server Event	Application server script facet template that you can use to create an event to be executed during runtime in the Supervision server.

Modifying the Tree View Layout

By using the **Settings** command, you can configure how many folders to display per row in the workspace of the **Global Templates Explorer**.

The setting applies to all the **Global Templates Explorer** windows that you open on the computer.

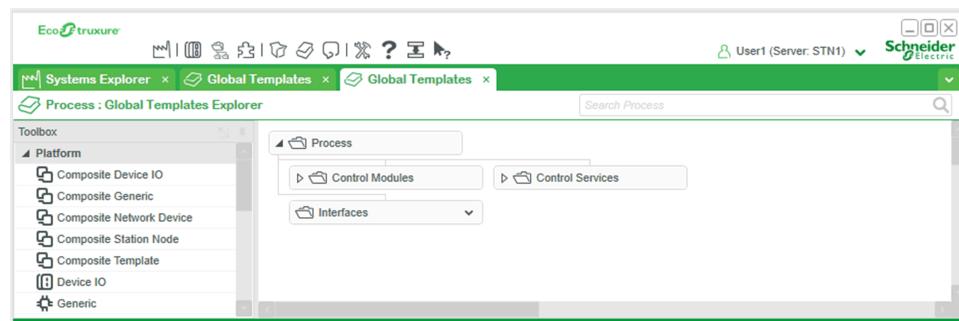
Opening Items in a New Window

You can open part of a folder structure that appears in the **Global Templates Explorer** in a new explorer window.

To open a folder structure in a new window, proceed as follows.

Step	Action
1	Drag the folder onto the tabs bar of the work area. Result: The folder opens in a new window as a parent folder. NOTE: If several work areas are open, you can drag the folder to the tabs bar of another work area.

The following figure shows an example where by expanding the sample **General Purpose Library** folder and dragging the **Process** folder to the tabs bar, a new explorer window opens in which the **Process** folder is the parent folder of the structure.



Recovery Folder

When you start the system server, if the software detects missing references in templates, it creates the **Recovery** folder. You can see the folder in the **Global Templates Explorer**.

This folder contains the templates that are missing.

This can be the case, for example, after migrating the database of an earlier version. If the database contains a template that is used by another template (determined by using the **Used By** command) but you cannot locate this parent template by searching for it, most likely you can find it in the **Recovery** folder.

You can work with templates inside the **Recovery** folder like with any other templates by right-clicking them and selecting an action, page 848.

Global Templates Folders

Creating Global Templates Folders

To create a Global Templates folder, proceed as follows.

Step	Action
1	Right-click the Global Templates root folder or an existing folder in the Global Templates Explorer and select Create Folder . Result: It creates a folder with the identifier in edit mode and displays it in the tree view.
2	To change the identifier, type it; otherwise, press Enter to accept the default name.

Global Templates Folder Actions

Right-click a Global Templates folder to open a context menu containing the following commands.

Command	Description						
Template Creation Wizard	Opens the Template Creation Wizard , which lets you create a control module template (see EcoStruxure™ Process Expert, Global Templates Reference Manual) with a few clicks given you have completed the necessary preparation work.						
Create Folder	Creates a new folder and displays it in the tree view.						
Create	Opens a submenu that allows you to create in the folder any base composite and facet template, which is available in the Toolbox .						
Control Constituent Password	Opens a submenu containing the following commands. The menu is only available if the Control constituent application password protection setting, page 849 for Control facet templates is enabled. NOTE: Using either command starts an operation that may take time and during this time, you cannot use the software.						
	<table border="1"> <tr> <td>Manage Password</td><td>Opens the Manage Password dialog box, which lets you set and change, page 850 the Controlconstituent password.</td></tr> <tr> <td>Clear Password</td><td>Opens the Clear Password dialog box, which lets you remove, page 851 the Controlconstituent password.</td></tr> <tr> <td>Forgot Password</td><td>Opens the Forgot Password dialog box, which contains an authentication code that you need to reset, page 851 the forgotten password.</td></tr> </table>	Manage Password	Opens the Manage Password dialog box, which lets you set and change, page 850 the Controlconstituent password.	Clear Password	Opens the Clear Password dialog box, which lets you remove, page 851 the Controlconstituent password.	Forgot Password	Opens the Forgot Password dialog box, which contains an authentication code that you need to reset, page 851 the forgotten password.
Manage Password	Opens the Manage Password dialog box, which lets you set and change, page 850 the Controlconstituent password.						
Clear Password	Opens the Clear Password dialog box, which lets you remove, page 851 the Controlconstituent password.						
Forgot Password	Opens the Forgot Password dialog box, which contains an authentication code that you need to reset, page 851 the forgotten password.						
Copy , page 885	Copies the folder, any subfolders and their contents.						
Paste , page 885	Pastes the items that you have copied last to a new location.						
Export , page 886	Opens the Export dialog box, which allows you to select which templates contained in the folder, including any subfolders you want to export to a file in .sbk format.						
Import , page 889	Opens the Import dialog box, which allows you to select an export file in .sbk format and displays the compatible contents that you can select for import.						
Purge , page 896	Opens the Purge dialog box, which allows you to select templates to purge in the folder and any subfolder.						
Delete	Deletes the folder and its contents from the database after you confirm the command. The command is not available if the folder or any subfolder contains a template.						
Update	Opens the Update Templates in Folder window, page 905, which allows you to create a new version of templates contained in the folder and, optionally, in its subfolders. It lists the templates for which at least one referenced child template with a later version exists in the Global Templates library. You can select the templates that you want to update and define various update settings. The command is not available from the Global Templates library root folder.						
Rename	Lets you enter a new name for the folder. The command is not available if the folder or any subfolder contains a template.						
Settings	Opens the Global Settings window, which lets you configure parameters, page 849 that impact template creation and modification. The command is only available from the Global Templates library root folder.						

Global Templates

Global Templates Description

The following fields are displayed in a Global Templates folder in the tree view of the **Global Templates Explorer**.

Field	Description
Identifier	The fields come from the header of the composite/facet template or interface.
Version	Refer to the following topics for a description of the fields for:
Type	<ul style="list-style-type: none">• Composite templates (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
Subtype	<ul style="list-style-type: none">• Facet templates (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
Valid	<ul style="list-style-type: none">• Interface models (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
State	
Description	

Global Templates Actions

Right-click a template in the **Global Templates Explorer** to open a context menu with the following commands.

Command	Description
Show Changes Log	opens the Changes Log pane, page 874, which lets you view the history of changes that were made to the template and saved.
View	Opens the template in the corresponding editor in view-only mode. The editor allows you to view the composition of the template or interface. NOTE: A password may be required to view a template.
Edit , page 853	Opens the template in the corresponding editor in edit mode. The editor allows you to view and modify the definition and composition of the template or interface. NOTE: A password may be required to edit a template.
State	Opens a submenu that allows you to set the usability state (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of the template to either Obsolete or Deprecated . The command is available only if: <ul style="list-style-type: none">• The template is used by an instance or referenced by another template.• A higher version of the template exists.• The usability state of the next highest version of the template is Approved.
Inspect	Allows you top open the Used By , Dependencies Tree , or External References panes, page 874 in which you can view respectively: <ul style="list-style-type: none">• The templates that reference it.• Templates that it references. For each reference you can view its interfaces.• Systems in which an instance of the template exists.
Locations	Opens a submenu that allows you view where the template is located. If the template is present at more than one location: <ul style="list-style-type: none">• The current location is grayed out.• You can click the other locations to open the folder and highlight the template.
Copy , page 885	Copies the template for pasting.
Export , page 886	Opens the Export dialog box, which allows you to save the template data to an export file in .sbk format.
Remove	Removes the copy of the template. The command is available only if copies, page 885 of the template exist. When only one instance of the template remains, the command is replaced by Delete .
Delete	Deletes the template from the database after you confirm the command. The command is not available if either condition is satisfied: <ul style="list-style-type: none">• The template is referenced by another template, which is not being deleted.• The template is used by a facet of in a system.
Update , page 901	Opens the Update dialog box, which allows you to create a new version of the template by using the latest version of any referenced child templates that exist in the Global Templates library.
Duplicate , page 912	Opens the Duplicate window, which allows you to create, in a single step, a copy or new version of the template with the possibility to keep, duplicate, or replace any child template that is references.

Global Template Settings

Section	Description
Template Editors	<p>When the Defer New Parameters And Interfaces Automatically setting is enabled (default), the new parameters and interfaces of a template and its elements (referenced templates) are automatically deferred (see EcoStruxure™ Process Expert, Global Templates Reference Manual) to the next higher level.</p> <p>This applies to the following operations on templates:</p> <ul style="list-style-type: none"> Referencing a template that has parameters and/or interfaces (some of which can be deferred from its own child elements). Updating a template or an element of a template with changes that include adding a new parameter or interface. Replacing an element with a template that has parameters and/or interfaces. <p>Existing parameters and interfaces are not deferred.</p> <p>You can open the Global Settings window from the Global Templates library root folder context menu to disable the functionality.</p>
Control Facet Templates	<p>The Control constituent application password protection setting lets you enable the Control Constituent Password context menu of the Global Templates root folder, page 846, which lets you set and manage a password, page 849 for the Control facet templates of the Global Templates library.</p> <p>The default value is inherited from the system server Control application and facet template password protection setting (see EcoStruxure™ Process Expert, Installation and Configuration Guide), which is configured in the Control Passwords section of the System Server Configuration Wizard. When the server-level setting is enabled, the Control constituent application password protection setting is enabled and read-only.</p> <p>Possible values:</p> <ul style="list-style-type: none"> Enabled: You are required to set a password for Control facet templates in the Manage Password dialog box after you enable the setting and click OK in the Settings window. Disabled (default): You cannot set a password for Control facet templates. To disable the setting, you must enter the current password in the Manage Password dialog box, which also clears it. <p>To change the setting, no other tab than the Global Templates Explorer must be open in the engineering client and no other clients must be connected to the system server.</p> <p>NOTE: Changing the setting may take time, page 849 and during this time, you cannot use the software.</p>

Managing the Control Constituent Password

Overview

The **Control Constituent Password** that you manage at the Global Templates level by using the folder context menu has the following purpose:

- It is used as application password in the Control Participant projects (.STU), which encapsulate Control constituents in a Control facet template.
- It enables file encryption for these Control projects.

Setting, changing, or clearing the password takes about 45 min. to complete with the Schneider Electric Global Templates library. More time is required if you have installed additional libraries and/or templates. During this time, you cannot use the software.

NOTE:

For an overview of the entire Control project password management functionality, refer to the topic describing the system and Control facet template passwords (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Prerequisites

The following are prerequisites to setting and managing the password:

- The **Control constituent application password protection** setting, page 849 for Global Templates is enabled.
- No other tab than the **Global Template Explorer** is open in the engineering client.
- No other client is connected to the system server.

Scope of the Control Constituent Password Protection

You are required to enter the password that is set for Control facet templates to perform the following actions:

- Click **Templatizer** in the **Facet Editor** when editing a Control facet template.
- Open the Global Templates creation wizard.
- Export templates.

Further, the application password and file encryption is applied to constituents that you encapsulate in new Control facet templates (*Logic*) that you create.

NOTE: After you have entered password once, you are not required to enter it again until close the engineering client or restart the system server.

Control Constituent Password Requirements

To be valid, the Control constituent password must contain the following:

- Between 8 and 16 characters.
- At least one uppercase letter from the classical Latin alphabet (A...Z).
- At least one lowercase letter from the classical Latin alphabet (a...z).
- At least one base-10 digit (0...9).
- At least one special character (~, !, @, \$, %, ^, &, *, _, +, -, =, `), |, \, (,), [,], :, “, ‘, <, >, {, }, ;, #).

Changing the Control Constituent Password

To change the Control constituent password of Control facet templates, proceed as follows.

Step	Action
1	In the Global Template Explorer , right-click the Global Templates folder (root) and click Control Constituent Password > Manage Password . Result: The Manage Password dialog box opens.
2	Enter the current password in the Current Password text box.
3	Enter the new password and confirm the new password in the corresponding text boxes.
4	Click OK . Result: The new password is applied to the Control facet templates of the Global Templates library. During this operation, you cannot use the software.

Clearing the Control Constituent Password

Clearing the Control constituent password of facet templates has the following impact:

- Clears the application password of Control Participant projects (.STU), which encapsulate Control logic in Control facet templates of the Global Templates library.
- Disables file encryption for these Control Participant projects.

NOTE: You cannot clear the password when the **Control application and facet template password protection** setting (see EcoStruxure™ Process Expert, Installation and Configuration Guide) is enabled at the system server level.

To clear the Control constituent password of Control facet templates, proceed as follows.

Step	Action
1	In the Global Template Explorer , right-click the Global Templates folder (root), click Control Constituent Password > Clear Password and confirm the command. Result: The Clear Password dialog box opens.
2	Enter the current password in the Current Password text box.
3	Click OK and confirm the command. Result: The password is cleared for each Control facet template of the Global Templates library. During this operation, you cannot use the software.

Resetting a Forgotten Control Constituent Password

To reset the Control constituent password of Control facet templates, proceed as follows.

Step	Action
1	In the Global Template Explorer , right-click the Global Templates folder (root) and click Control Constituent Password > Forgot Password . Result: The Forgot Password dialog box opens and an authentication code is displayed.
2	Copy the authentication code, contact Schneider Electric support, page 104 and provide the code. Result: You receive a temporary password.
3	Right-click the Global Templates folder and click Control Constituent Password > Manage Password . Result: The Manage Password dialog box opens.
4	Follow the procedure to change the password, page 850 and enter the temporary password in the Current Password text box.

Password Management When Exporting and Importing Templates

When you export a template from the Global Templates library, its application password is removed and file encryption is disabled.

When you import a template, an application password is set and file encryption is enabled if the **Control constituent application password protection** setting, page 849 of the Global Templates library in which you import is enabled and the template supports the functionality.

Information about Control facet templates to which the application password and file encryption could not be applied is shown in the system server console and in the notification panel of engineering clients.

Contact Schneider Electric support, page 13 if you are faced with this situation and want password protection and file encryption to be applied.

Global Templates Editors

What's in This Chapter

Opening Global Templates Editors	853
Interface Editor.....	854
Facet and Composite Template Editors	859
Common Template Editor Components.....	867

Overview

This chapter describes the Global Templates editors, which allow you to view and edit the entire definition of Global Templates. They also allow you to create new templates based on existing ones or by starting from empty base templates.

You can view and edit:

- Composite templates by using the **Composite Editor**.
- Facet templates by using the **Facet Editor**.
- Interface models and interfaces by using the **Interface Editor**.

Starting from the control module template level, you can drill down through the entire composition of the template. At each level, you can view dependencies and which templates reference the one you are viewing or editing.

NOTE: Some of the commands that are described in this chapter may not be available when you open a template editor in read-only mode.

Opening Global Templates Editors

Opening Global Templates Editors

Overview

Modifying Schneider Electric Global Templates or templates created by users may affect the function of these templates and must be performed by qualified personnel. Before proceeding, refer to the Overview of this part, page 837.

Opening Global Templates Editors

To open a Global Templates editor, you need to open a template. You cannot open a Global Templates editor without opening a template either in read-only or editing mode.

Step	Action
1	In the tree view of the Global Templates Explorer , open the folder that contains the template that you want to view or edit.
2	Right-click the template and select: <ul style="list-style-type: none">• Read-only to open the template in the corresponding editor in read-only mode. This mode does not allow you to change the definition of the template. The background of the workspace has a light shade of gray to distinguish it from the editing mode.• Edit to open the template in the corresponding editor in editing mode. This mode allows you to modify the entire definition of the template. NOTE: Double-clicking the template opens the corresponding editor in read-only mode by default.

NOTE: From the **Global Templates explorer**, you can also look up templates by entering a key word in the search field, page 839. In the list of results, right-click the template and select **Read-only** or **Edit**. The software opens the selected template in the corresponding editor in either mode.

Working Copy of Templates

When you open a template in read-only or editing mode, the software creates a temporary working copy of the template with the same identifier and a specific version (see EcoStruxure™ Process Expert, Global Templates, Reference Manual). It removes this working copy when you close the template editor.

Interface Editor

Interface Editor

Overview

The graphical **Interface Editor** allows you to manage the entire definition of interface models and interfaces.

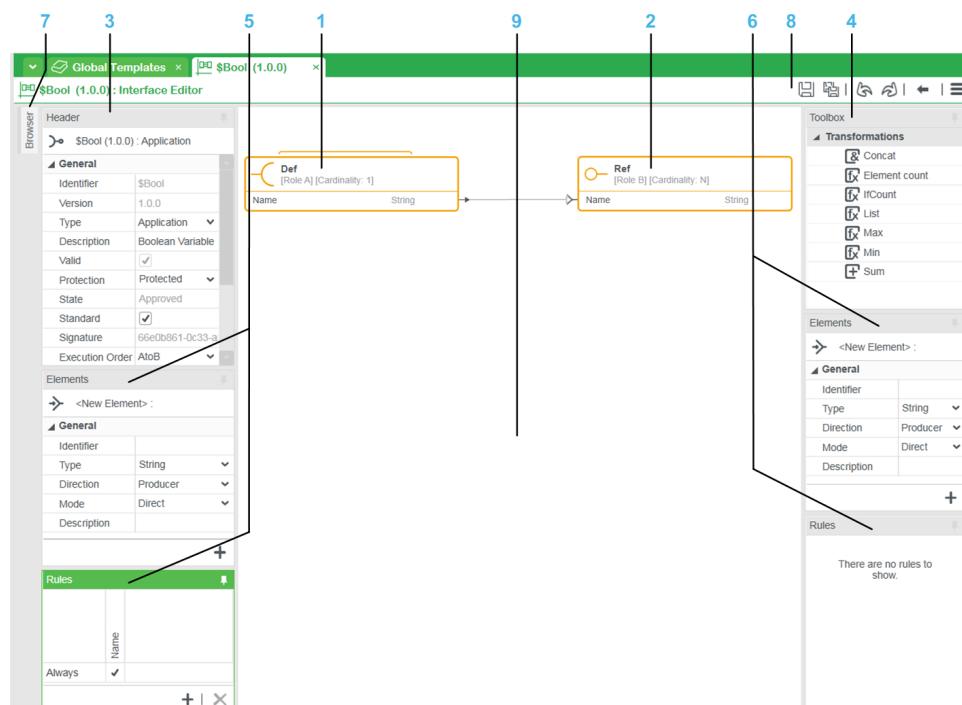
This topic describes the panes, toolbars, and menus of the **Interface Editor**.

Certain menus and commands may be available only in editing mode.

NOTE: This topic does not contain information on the definition of interface models or interfaces (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

Interface Editor

The following figure shows an example of the **Interface Editor** in edit mode.



Item	Description
1	Graphical representation of role A of the interface
2	Graphical representation of role B of the interface
3	Interface Header pane
4	Toolbox pane
5	Elements and Rules panes for role A of the interface
6	Elements and Rules panes for role B of the interface
7	Button to show the interface Browser pane
8	Editor toolbar containing commands that apply to the interface being viewed or edited
9	Workspace. You can use the mouse to move the view, page 106.

Editor Panes

The **Interface Editor** uses various panes to group properties and data that are related. The table describes the purpose of the panes.

Pane	Description
Header	Allows you to view and/or edit general properties of the interface.
Toolbox	Allows you to browse and select transformation functions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) to be used in the template.
Elements	Allows you to view, create, and modify elements of a role of the interface.
Rules	Allows you to view, create, and modify rules for the elements of a role of the interface.
Browser	Allows you to search interface models and interfaces that are available in the Global Templates library. Move the pointer over the Browser tab to display the pane. Click the pin button to toggle between minimized and visible pane view. NOTE: The pane is available only in editing mode.
Used By⁽¹⁾	Allows you to view the templates and interfaces that reference the interface, which is open in the editor. You can right-click a template or interface in the Used By pane to open it in a new editor window. You can also view its dependencies, page 855 or which other templates reference it.
Locations⁽¹⁾	Allows you to view the path to the interface and its copies, page 885. Right-click an entry and select Open Reference Container to open the corresponding folder in the Global Templates explorer.
Changes Log⁽¹⁾	Keeps track of changes that are made to the interface and saved. The pane indicates: <ul style="list-style-type: none"> • The action that was performed. • The state of the interface after the change. • The mandatory description of changes that is entered when the interface is saved. • The version of the interface after the change. • The user name of the user logged in to the engineering client.
(1) Use the corresponding command in the editor toolbar menu to open the pane.	

NOTE: For information on working with panes, refer to the topic describing the engineering client workspace, page 111.

Additional Information

The list indicates where you can find a detailed description of the contents of the **Interface Editor** panes:

- Header definition (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) and header common definition (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) (shared among interface models, facet, and composite templates).
- Elements definition (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
- Rules definition editors (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
- Interface toolbox (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

Editor Toolbar

The figure shows the toolbar that is located in the top right corner of the **Interface Editor**.



Item	Description
1	Edit button. Switches the editor from read-only to editing mode.
2	Save button. Opens the Save dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.
3	Save as button. Opens the Save as dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.
4	Undo and redo buttons. Lets you undo and redo most actions.
5	Shows the parent from which you have opened the interface and highlights it. If you had opened an interface that is an element of a template, only the template is highlighted. The parent template must be open for the command to be enabled. The command is not enabled when the interface is opened from the Global Templates Explorer .
6	Opens the editor toolbar menu that contains editor-specific commands.

Editor Menu

The table describes the commands that are available in the editor toolbar menu of the **Interface Editor**.

Command	Description		
New...	Opens the Global Templates dialog box, which contains the same base templates as the Toolbox of the Global Templates explorer. It allows you to create a new template in one or more existing folders of the Global Templates library and open it in edit mode in the corresponding editor. For more information, refer to the topic describing how to create templates (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).		
View	Show Changes Log	The commands allow you to bring the corresponding pane to the front.	
	Used By	If the pane is closed, the command opens it.	
	Locations	For a description of the panes, refer to the topic describing editor panes, page 855.	
Save	Save button. Opens the Save dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.		

Command		Description
Save As		Save as button. Opens the Save As dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.
Export	Standard Backup	Opens the Export window, which allows you to export the interface definition to file (.sbk). You can only export an interface if you have saved changes. For more information on export feature, refer to the topic describing template export, page 886.
	Various image file formats	Each command opens a Save dialog box, which allows you to create an image file with the corresponding file extension. The image captures the contents of the workspace as if it were shown with the Fit to content display ratio. Panes and toolbars are not captured.
Close		Closes the template editor. If you have made changes, opens the Save dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.

Workspace Actions

Right-click an empty area of the workspace to open a context menu with the following commands.

Command	Description
Show changes log	Opens the Changes log pane.

Interface and Interface Element Actions

Right-click the header of an interface role in the workspace to open a context menu with the following command.

Command	Description
Rename	Allows you to modify the identifier of the role. For more information, refer to the topic describing interface roles (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

Right-click an element of an interface role to open a context menu with the following commands. Commands may vary depending on the role of the element.

Command	Description
Switch to Optional	Sets the rule for the element, which is the producer to <i>optional</i> . In such case, the element is not required to provide data for the interface link to be valid. The default value is <i>required</i> . NOTE: If you set a role to optional and later you change the role to consumer, the role is reset to <i>required</i> .
Switch to Required	Sets the rule for the element, which is the producer to <i>required</i> (default value). In such case, the element is required to provide data for the interface link to be valid.
Use as Consumer	Switches the direction from <i>producer</i> to <i>consumer</i> . NOTE: Using this command resets the element rule to <i>required</i> if it was set to <i>optional</i> .
Use as Producer	Switches the direction from <i>consumer</i> to <i>producer</i> .

Command	Description
Switch Mode	<p>Switches the mode of the link between role A and role B of an element from <i>Direct</i> to <i>Transform</i> and the other way around.</p> <p>Right-clicking an element and switching the mode to:</p> <ul style="list-style-type: none"> • <i>Transform</i> <ul style="list-style-type: none"> ◦ Removes the direct link between role A and role B of the element. ◦ Removes the element from the other role. ◦ Requires that you create a new element in the other role with mode <i>Transform</i> and with the opposite direction. • <i>Direct</i> <ul style="list-style-type: none"> ◦ Removes the link between the element and the transformation function. ◦ Creates the element in the other role of the interface. ◦ Creates a direct link between both roles of the element. ◦ Requires that you reconfigure the transformation function, and possibly the elements that are still linked to it. <p>NOTE: When you switch the mode to <i>Transform</i>, verify that the cardinality that is defined in the Header pane for either role A or role B makes it possible to use transformation functions.</p> <p>NOTE: The command is not available for nested interfaces.</p>
Delete	Deletes the element from both roles of the interface.

Right-click a nested interface of an interface role to open a context menu with the following commands. Commands may vary depending on the role of the element.

Command	Description
Switch to Optional	<p>Sets the rule for the element to <i>optional</i>. In such case, the element is not required to provide data for the interface link to be valid.</p> <p>The default value is <i>required</i>.</p> <p>NOTE: The command is available by default for the <i>Def</i> role of the nested interface. To make the command available for the <i>Ref</i> role, add the element to the Rules table first by using the Add to Rules table command first.</p>
Switch to Required	Sets the rule for the element, which is the producer to <i>required</i> (default value). In such case, the element is required to provide data for the interface link to be valid.
Add to Rules table	Adds the element that has the <i>Ref</i> role to the Rules table so that you can define a rule for it (<i>optional</i> or <i>required</i>).
Remove from Rules table	Removes the element that has the <i>Ref</i> role from the Rules table.
NOTE: Removing the element from the Rules table makes the Switch to Optional and Switch to Required commands unavailable.	
Inspect	Opens a submenu with the Used By and Dependencies Tree , page 855 commands.
Read-only	opens the nested interface in another Interface Editor in read-only mode.
Edit	opens the nested interface in another Interface Editor in editing mode.
Update	Allows you to update the version of the nested interface to the latest one, which exists in the Global Templates library.
Replace	Opens the Replace dialog box. It allows you to replace the nested interface by a different one, which exists in the Global Templates library.
Delete	Deletes the nested interface from both roles of the interface.

Saving Changes

Refer to Saving Changes in Global Templates, page 882.

Facet and Composite Template Editors

Overview

This topic describes the user interface of the **Facet Editor** and **Composite Editor**.

Facet Editor

Overview

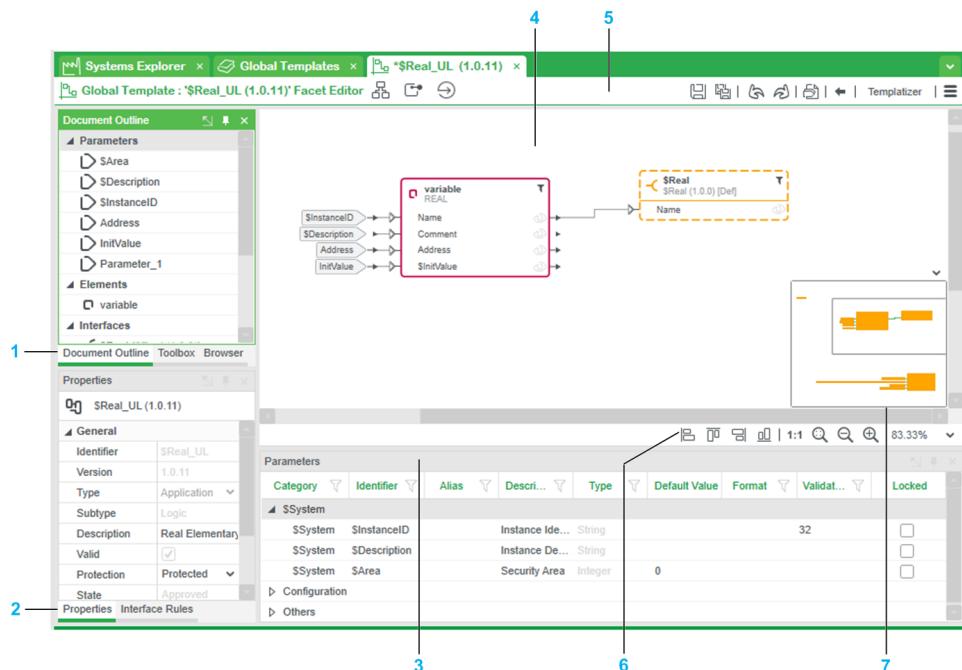
The graphical **Facet Editor** allows you to manage the entire definition of Control and Supervision facet templates.

For a description of panes, toolbars, and menus, refer to the topic describing common template editor components, page 867.

NOTE: This topic does not contain information on the definition of facet templates (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

Facet Editor

The following figure shows an example of the **Facet Editor** in editing mode.



Item	Description
1	Document Outline, Toolbox, and Browser panes.
2	Properties and Interface Rules panes.
3	Parameters pane.
4	Workspace area displaying a graphical representation of the elements of the template that you are editing.
5	Editor toolbar, page 868 containing commands that apply to the template being viewed or edited and/or its elements.
6	Workspace toolbar, page 869.
7	Radar view, page 877.

Workspace Actions

For a description of the context menu that opens when you right-click an empty area of the workspace, refer to the description of the workspace context menu, page 872.

Element Actions

Right-click the header of an element in the workspace of the **Facet Editor** to open a context menu with the following commands.

Command	Description
Optimize Bindings	Reroutes bindings of the element correctly.
Show/Hide Unchecked	The commands have been replaced by a filter menu, page 876 at the element level.
Show/Hide Unbound	
Go To	Opens a submenu that contains the name and path of other objects and/or their properties that are linked to the element through bindings. It lets you locate them (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) easily. When you select an entry, the object and the binding are highlighted in the workspace.
Copy	Copies the element for pasting in this template or another template.
Exclude	After you confirm the command, removes the element from the workspace and makes it available again in the Elements section of the Document Outline pane. Any existing bindings with the element are discarded. NOTE: The command is available only for Control elements.
Delete	After you confirm the command, removes the element from the facet template. Any existing bindings with the element are discarded. NOTE: The command is available only for certain types of elements.
Rename	Allows you to modify the identifier of the element.
Properties	Displays the properties of the element in the Properties pane.

Right-click the property of an element in the workspace of the **Facet Editor** to open a context menu with the following commands.

Command	Description
Create Parameter	Allows you to create a parameter that you can customize (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
Create Input Value	Allows you to enter an absolute value.
Show/Hide \$IsConnected	Displays or hides the <code>\$IsConnected</code> property, which outputs the boolean value <code>TRUE</code> when the interface link is successfully established. This implies that applicable interface and interface element rules are satisfied; otherwise the value is <code>FALSE</code> . The command is available for interfaces only.
Go To	Opens a submenu that contains the name and path of other objects and/or their properties that are linked to this property through bindings. It lets you locate them (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) easily. When you select an entry, the object and the binding are highlighted in the workspace.
Binding To Here From...	Use this command (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) on the property that is the destination of the binding to be created. It enables the Bind to command.
Binding From Here To...	Use this command (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) on the object that is the source of the binding to be created. It enables the Bind from command.
Properties	Displays the properties of the element in the Properties pane.

Right-click the header of an interface element in the workspace of the **Facet Editor** to open a context menu with the following commands.

Command	Description
Switch to Optional/Required	Optional: Makes the element optional in the context of the template. The element is represented with a dotted outline. Required: Makes the element mandatory in the context of the template. The element is represented with a solid outline.
Optimize Bindings	Reroutes bindings of the element correctly.
Show/Hide Unchecked	The commands have been replaced by a filter menu, page 876 at the element level.
Show/Hide Unbound	
Inspect	The menu entry has been removed. Use the left-hand toolbar buttons instead to open the Used By and Dependencies Tree panes, page 874.
View	Opens the element in the corresponding editor in read-only mode.
Edit	Opens the element in the corresponding editor in edit mode.
Edit/Extend Interface	Opens the Edit/Extend Interface window, page 917, which lets you edit, add, and remove elementary elements (in direct mode only) in the interface without the need to edit the interface by using the Interface Editor . The command is not available for deferred interfaces and when the interface element contains a nested interface. Updating the templates that reference the other role of the interface is required.
Go To	Opens a submenu that contains the name and path of other objects and/or their properties that are linked to the element through bindings. It lets you locate them (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) easily. When you select an entry, the object and the binding are highlighted in the workspace.
Copy	Copies the element for pasting in this template or another template.
Update, page 901	Allows you to update the version of the interface model that is used by the interface to the latest one, which exists in the Global Templates library.
Replace, page 907	Opens the Replace dialog box. It allows you to replace the interface model that is used by the interface by a different one, which exists in the Global Templates library.
Delete	Removes the element from the template.
Rename	Allows you to modify the identifier of the element.
Properties	Displays the properties of the element in the Properties pane.

Additional Information

The list indicates where you can find a detailed description of the contents of the various panes of the **Facet Editor**:

- Properties (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) and properties common definition (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) (shared among interface models, facet, and composite templates).
- Facet Elements (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
- Interfaces Rules (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
- Toolbox: binding functions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) and/or Supervision Participant elements, page 874.
- Parameter pane (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)

Saving Changes

Refer to Saving Changes in Global Templates, page 882.

Composite Editor

Overview

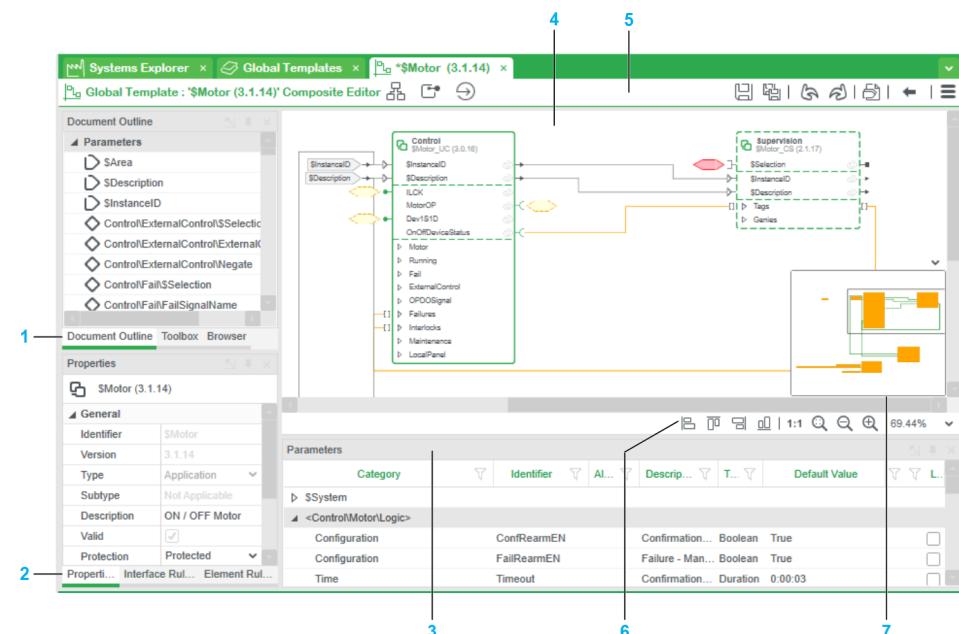
The graphical **Composite Editor** allows you to manage the entire composition and definition of composite templates.

For a description of panes, toolbars, and menus, refer to the topic describing common template editor components, page 867.

NOTE: This topic does not contain information on the definition of composite templates (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

Composite Editor

The following figure shows an example of the **Composite Editor** in editing mode.



Item	Description
1	Document Outline, Toolbox, and Browser panes.
2	Properties, Interface Rules, and Element Rules panes.
3	Parameters pane.
4	Workspace area displaying a graphical representation of the elements of the template that you are editing.
5	Editor toolbar, page 868 containing commands that apply to the template being viewed or edited.
6	Workspace toolbar, page 869.
7	Radar view, page 877.

Workspace Actions

For a description of the context menu that opens when you right-click an empty area of the workspace, refer to the description of the workspace context menu, page 872.

Element Actions

Right-click the header of an element in the workspace of the **Composite Editor** to open a context menu with the following commands.

Command	Description
Switch to Optional/Required	Optional: makes the element optional in the context of the template. The element is represented with a dotted outline. Required: makes the element mandatory in the context of the template. The element is represented with a solid outline.
Defer	Defers items to make them available in the top-level template referencing the composite that you are editing (for example, in the control module). Selecting the command displays a submenu with the following commands: <ul style="list-style-type: none"> • Unbound selection/parameters: Makes available any element selection (only optional elements) and parameters for which no binding exists. • Unbound interfaces: Makes available any interfaces for which no binding exists. • All unbound: Makes available any element selection (only optional elements), parameters, and interfaces for which no binding exists.
Optimize Bindings	Reroutes bindings of the element correctly.
Show/Hide Unchecked	The commands have been replaced by a filter menu, page 876 at the element level.
Show/Hide Unbound	
Inspect	The menu entry has been removed. Use the left-hand toolbar buttons instead to open the Used By , Dependencies Tree , and External References panes, page 874.
View	Opens the element in the corresponding editor in read-only mode.
Edit	Opens the element in the corresponding editor in edit mode.
Edit/Extend Interface	For interfaces only. Opens the Edit/Extend Interface window, page 917, which lets you edit, add, and remove elementary elements (in direct mode only) in the interface without the need to edit the interface by using the Interface Editor . The command is not available for deferred interfaces and when the interface element contains a nested interface. Updating the templates that reference the other role of the interface is required.
Go To	Opens a submenu that contains the name and path of other objects and/or their properties that are linked to the element through bindings. It lets you locate them (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) easily. When you select an entry, the object and the binding are highlighted in the workspace.
Copy	Copies the element for pasting in this template or another template.
Update, page 901	Allows you to update the version of the template that is used by the element to the latest one, which exists in the Global Templates library. In case of an interface element, lets you replace the interface model.
Replace, page 907	Opens the Replace dialog box. It allows you to replace the template that is used by the element by a different one, which exists in the Global Templates library. In case of an interface element, lets you replace the interface model.
Delete	Removes the element from the template.
Rename	Allows you to modify the identifier of the element.
Properties	Displays the properties of the element in the Properties pane.

Right-click the parameter of an element in the workspace of the **Composite Editor** to open a context menu with the following commands.

Command	Description
Create Deferred	Defers the parameter to make it available in the top-level template referencing the composite that you are editing.
The other commands are described in the table describing the context menu commands of properties of elements of facet templates, page 861.	

Right-click the interface of an element in the workspace of the **Composite Editor** to open a context menu with the following commands.

Command	Description
Create Deferred	Defers the interface to make it available in the top-level template referencing the composite that you are editing.
Create Extended	Shows the interface as an element of the template.
Explode Implode	<ul style="list-style-type: none"> Collapsed state: Makes a connector available that represents the complete interface. The connection is made by using an interface link (shown in orange). Expanded state: Makes each element of a multi-element interface available individually for connection to other elements by using bindings (shown in gray).
Inspect	<p>The menu entry has been removed. Use the left-hand toolbar buttons instead to open the Used By and Dependencies Tree panes, page 874.</p> <p>They allow you to view respectively:</p> <ul style="list-style-type: none"> The templates that reference it. Nested interfaces that it references.
View	Opens the interface in the corresponding editor in read-only mode.
Edit	Opens the interface in the corresponding editor in edit mode.
Go To	Opens a submenu that contains the name and path of other objects and/or their properties that are linked to the interface through bindings. It lets you locate them (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) easily. When you select an entry, the object and the binding are highlighted in the workspace.
Properties	Displays the properties of the interface in the Properties pane.

NOTE: Additional commands are described in the topic describing actions for properties of elements, page 861 in the **Facet Editor**.

Additional Information

The list indicates where you can find a detailed description of the contents of the various panes of the **Composite Editor**:

- Properties (Header) (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) and properties common definition (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) (shared among interface models, facet, and composite templates).
- Composite Elements (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
- Interfaces Rules (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
- Toolbox: binding functions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)
- Parameter pane (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)

Saving Changes

Refer to Saving Changes in Global Templates, page 882.

Common Template Editor Components

Overview

The **Facet Editor** and **Composite Editor** have several components in common, which are described in this topic. Items that are specific to either editor are pointed out.

Common Template Editor Toolbars and Menus

Overview

The **Facet Editor** and **Composite Editor** feature toolbars and menus, which allow you to access the various tools and functions that you require to create and edit Global Templates.

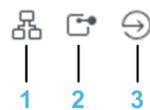
This topic describes toolbars and menus that the **Facet Editor** and **Composite Editor** have in common. Those that are specific to either editor are pointed out.

Certain menus and commands may be available only in editing mode.

NOTE: Unless otherwise mentioned, toolbars contain commands that apply to the template that is being viewed or edited. To interact with an element of a template, open the context menu of the element by right-clicking it.

Editor Toolbars

The figure shows the toolbar that is located in the top left corner of the editors.



The buttons apply to the template that is being edited. However, if an element inside this template is selected, they apply to this element.

Item	Description
1	Displays the Dependencies Tree pane, page 874.
2	Displays the Used By pane, page 874.
3	Displays the External References pane, page 874.

The figure shows the editor toolbar that is located in the top right corner of the editors.



Item	Description
1	Edit button. Switches the editor from read-only to editing mode.
2	Save button. Opens the Save dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.
3	Save as button. Opens the Save As dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.
4	Undo and redo buttons. Lets you undo and redo most actions.
5	Displays the Parameters pane, page 874.
6	Shows the parent from which you have opened the template and highlights the template. If you had opened an element inside a template, only the template is highlighted. The parent template must be open for the command to be enabled. The command is disabled when the child is opened from the Global Templates Explorer .
7	Opens the Select Variables or Select Genie window, which allows you to encapsulate constituents in Control and compatible Supervision facet templates respectively. It also allows you to modify encapsulated elements with the help of the corresponding Participant. If a Control facet contains no constituents, the Content Not Found dialog box opens, which allows you to start the encapsulation process. For more information, refer to the topic describing the configuration of facet templates (see EcoStruxure™ Process Expert, Global Templates, Reference Manual). NOTE: The command is available only in the Facet Editor in edit mode.
8	Opens the editor menu, page 870 that contains editor-specific commands.

Workspace Toolbar

The figure shows the workspace toolbar that is located in the bottom right-hand corner of the **Facet Editor** and **Composite Editor**.



Item	Description
1	Aligns the left edge of any selected item in the workspace with the left edge of the item that you have selected first. NOTE: The graphical alignment rules do not apply to parameters and input values, except for the three \$System parameters \$InstanceID, \$Description, and \$Area.
2	Aligns the top edge of any selected item in the workspace with the top edge of the item that you have selected first.
3	Aligns the right edge of any selected item in the workspace with the right edge of the item that you have selected first.

Item	Description
4	Aligns the bottom edge of any selected item in the workspace with the bottom edge of the item that you have selected first.
5	Reverts to the default zoom level (100%).
6	Adjusts the zoom level to fit the elements of the template inside the visible workspace. If the layout of the template is very large, some objects may not fit inside the visible workspace. You can use the radar view, page 877 to locate these objects.
7	Zooms out, page 107. Alternatively, use the minus (-) keyboard key. When you zoom out to 50%, the display switches to the simplified view, page 878. To return to the normal view, zoom in.
8	Zooms in. Alternatively, use the plus (+) keyboard key.
9	Current zoom level. Use a predefined level by clicking the arrow and selecting a value from the list. You can set a custom level by double-clicking the value, entering a new one, and pressing Enter . Range: 50 to 200.

NOTE: For information on commands that allow you to select multiple items in the workspace and move the workspace around, refer to the topic describing the engineering client workspace, page 106.

Editor Menu

The table describes the submenus and commands of the menu located in the toolbar of template editors.

Command		Description
New...		Opens the Global Templates dialog box, which contains the same base templates as the Toolbox of the Global Templates explorer. It allows you to create a new template in one or more existing folders of the Global Templates library and open it in editing mode in the corresponding editor. For more information, refer to the topic describing how to create templates (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
Edit	Copy	Copies the selected element for pasting in this template or another template. You can copy multiple elements by selecting them first.
	Paste	Pastes copied elements. When a naming rule applies to elements, the software uses the <i>_n</i> suffix (where <i>n</i> is an incremental number starting at 1) to create unique names.
	Select All	Selects any element in the workspace.
	Select None	Clears any selected element in the workspace.
	Invert Selection	Selects elements that are not selected and clears any selected element in the workspace.
	Find	Displays a search field, which allows you to find strings in any element of the workspace by entering entire or partial key words. Strings that contain the word you entered appear in a list with an indication of their path. Right-click a result and select Navigate to highlight it in the workspace. For example, entering <i>in</i> returns various results, such as: <ul style="list-style-type: none"> • The <i>\$InstanceID</i> system parameter. • The <i>\$InstanceID</i> parameter of an element. • The <i>In</i> input of a binding function. • <i>Interfaces</i> to indicate that an element references interfaces.
	Show Changes Log	The commands allow you to bring the corresponding pane to the front.
	Document Outline	If the pane is closed, the command opens it.
	Dependencies Tree	For a description of the panes, refer to the topic describing template editor panes, page 874.
	Used By	
	External References	
	Locations	
	Parameters	
	Interface Rules	
	Element Rules⁽¹⁾	
	Toolbox	
	Browser	
	Properties	
	Save	Save button. Opens the Save dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.
	Save As	Save as button. Opens the Save As dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.

Command		Description
Export	Standard Backup	Opens the Export window, which allows you to export the template definition to file (.sbk). You can only export a template if you have saved changes. For more information on export feature, refer to the topic describing template export, page 886.
	Various image file formats	Each command opens a Save dialog box, which allows you to create an image file with the corresponding file extension. The image captures the entire layout of the template independently of the zoom level. Panes and toolbars are not captured.
Print	Only Visible Content	Opens the Print dialog box, which lets you print on one page, the portion of the template layout that is visible in the current view.
	Editor Content On Single Page	Opens the Print dialog box, which lets you print the entire layout of the template on one page.
Close		Closes the template editor. If you have made changes, opens the Save dialog box. For more information, refer to the topic describing how to save changes in templates, page 882.
(1) The command appears only in the menu of the Composite Editor		

Workspace Context Menu

Right-click an empty area of the workspace of a template editor to open a context menu with the following commands.

Command	Description
Inspect	The menu entry has been removed. Use the left-hand toolbar buttons instead to open the Used By , page 855 and Dependencies Tree , page 855 panes.
Align Left Edges	Refer to the description of the corresponding commands in the workspace toolbar, page 869.
Align Top Edges	The commands are available only if two or more elements are selected in the workspace.
Align Right Edges	
Align Bottom Edges	
Copy	Copies the selected element for pasting in this template or another template. You can copy multiple elements by selecting them first.
Paste	Pastes copied elements. When a naming rule applies to elements, the software uses the <i>_n</i> suffix (where <i>n</i> is an incremental number starting at 1) to create unique names.
Show Changes Log	Opens the Changes Log pane, page 874, which lets you view the history of changes that were made to the template and saved.
Properties	Brings the Properties pane to the front to display properties of the template that is open in the editor. If the pane is closed, the command opens it.

Binding Context Menu

Right-click a binding or its connector (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) in the workspace of a template editor to open a context menu with the following commands.

Command	Description
Convert To Line	Lets you toggle between binding styles (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
Convert to Connector	
Go To	For bindings in line style, opens a submenu that contains the name and path of objects and/or their properties that are linked to this object through bindings. It lets you locate them (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) easily. When you select an entry, the object and the binding are highlighted in the workspace. For bindings in connector style, highlights the object that is connected to the other end of the binding.
Update Binding	Opens the Update Binding dialog box, which lets you move the destination (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of the binding to another property or element. The command is not available for source connectors of bindings.
Delete	Deletes the binding after you confirm the command.

Common Template Editor Panes

Introduction

The **Facet Editor** and **Composite Editor** use various panes to group properties and data that are related.

This topic describes panes that are common to both. Panes that are specific to either editor are pointed out.

Certain panes may be available only in editing mode.

For information on working with panes, refer to the topic describing the engineering client workspace, page 111.

NOTE: This topic does not describe how to define or modify the contents of the panes.

Editor Panes

The table describes the purpose of the panes that you can access in the **Facet Editor** and **Composite Editor**.

Pane	Description
Document Outline	Allows you to view and/or edit: <ul style="list-style-type: none"> • Parameters • Elements • Interfaces Click a parameter, element, or interface to display its properties in the Properties pane. The corresponding item is selected in the workspace. NOTE: To view and/or edit the properties of the template, click Parameters .
Interface Rules	Allows you to view and define the rules for the interfaces (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) referenced by the template.
Element Rules⁽¹⁾	Allows you to view and define the rules for the elements (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) referenced by the template.
Properties	Allows you to view and/or edit the properties of the selected element of the template.
Toolbox	Allows you to browse and select binding functions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) to be used in the template. For facet templates of the Supervision category, the Toolbox pane also contains the following elements (see EcoStruxure™ Process Expert, Global Templates, Reference Manual): <ul style="list-style-type: none"> • For facet templates of the Data subtype: <ul style="list-style-type: none"> ◦ Advanced Alarm ◦ Calculated Variable Tag ◦ Disk Variable Tag ◦ Digital Alarm ◦ Equipment ◦ Equipment Parameter ◦ Equipment Group of Messages ◦ Local Variable Tag ◦ Message ◦ Time Stamped Digital Alarm ◦ Trend Tag ◦ Variable Tag • For facet templates of the Genie subtype: <ul style="list-style-type: none"> ◦ Equipment ◦ Equipment Parameter ◦ Equipment Group of Messages

Pane	Description
	<ul style="list-style-type: none"> For facet templates of the Client Event subtype: Event. For facet templates of the Server Event subtype: Report. <p>These elements contain the same properties as the corresponding elements of the Supervision Participant and allow populating the respective databases.</p>
Browser	<p>In the Facet Editor, allows you to browse and select interfaces to be referenced by the template.</p> <p>In the Composite Editor, allows you to browse and select facet and composite templates, and interfaces to be referenced in the template.</p>
Dependencies Tree	<p>In the Facet Editor, allows you to view the interfaces that the template references.</p> <p>In the Composite Editor, allows you to view the interfaces, facet and composite templates that the template references.</p> <p>You can right-click an item to view and/or edit it in a new editor. You can also view the dependencies of the item or which other templates reference it.</p>
Used By	Allows you to view the templates that reference the template, which is open in the editor. You can right-click a template in the Used By pane to open it in a new editor. You can also view its dependencies or which other templates reference it.
External References	<p>Allows you to view:</p> <ul style="list-style-type: none"> In which systems an instance of the template exists. For each system, the identifier of the instances using the template. <p>You can right-click an application instance to:</p> <ul style="list-style-type: none"> Open the folder containing the instance in the Application Explorer. Open the Inspect Instance window, page 204.
Parameters (see EcoStruxure™ Process Expert, Global Templates, Reference Manual)	The Parameters pane allows you to: <ul style="list-style-type: none"> View and manage parameter data that is used during instantiation to customize elements of an instance. Customize the layout of the parameters that are visible when you edit an instance by using the Instance Editor: <ul style="list-style-type: none"> Change the position of parameters within their category. Change the position of categories. Assign parameters to another category.
Locations	Allows you to view the path to the template and its copies, page 885. Right-click an entry and select Open reference container to open the corresponding folder in the Global Templates explorer.
Changes Log	Keeps track of changes that are made to the template and saved. The pane indicates: <ul style="list-style-type: none"> The action that was performed. The state of the template after the change. The mandatory description of changes that is entered when the template is saved. The version of the template after the change. The user name of the user logged in to the engineering client.
(1) The pane appears only in the Composite Editor .	

Common Template Editor Filters

Overview

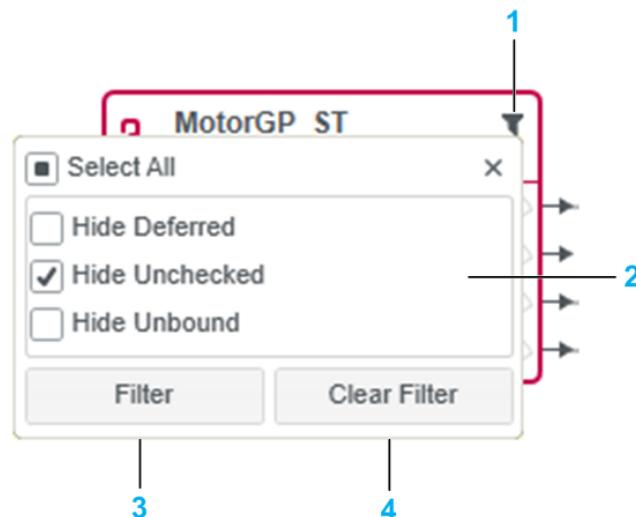
Starting with version 2021, the following context menu commands of elements, which appear in the **Facet Editor** and **Composite Editor** have been replaced by a filter menu.

- **Show/Hide Unchecked**
- **Show/Hide Unbound**

Also, the **Hide Deferred** filter criteria has been added.

Filter Menu Description

The following figure shows an example of the filter menu that is located in the header of an element with its default setting.



Item	Description	
1	Click the button to open the element filter menu. When at least one filter is applied, the button is shown with a black fill.	
2	Filter criteria to apply to the element representation. You can select any combination or none.	
	Hide Deferred	When selected, hides properties and interfaces that are deferred.
	Hide Unchecked	When selected, hides properties and interfaces whose eye icon that is located on the right-hand side is not visible (unchecked). To uncheck a property/interface, click its icon. The icon disappears. NOTE: A property/interface that has a binding connected to it cannot be hidden even if it is unchecked.
	Hide Unbound	When selected, hides properties and interfaces that have no binding connected.
3	Click the button to apply the selected filter criteria to the representation of the element.	
4	Click the button to clear the filter criteria selection and update the representation of the element.	

Using the Radar View

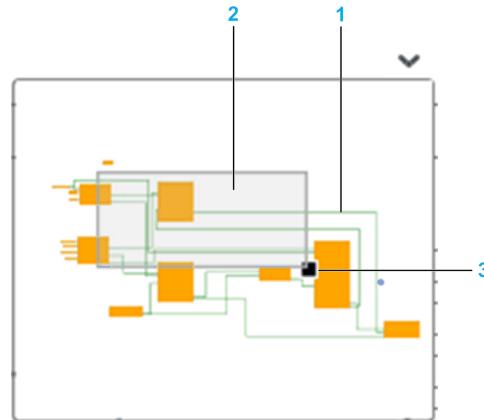
Overview

The radar view appears in the bottom right corner of certain graphical editors. It shows you which area of the layout of a template or workspace you are seeing in the editor.

You can also use it to navigate the inside the workspace and adjust the zoom level.

Description

The following figure shows an example of the **Radar View** in a template editor.



Item	Description
1	Shows all the objects that appear in the editor. The view is refreshed when you modify the layout of objects.
2	Area that is currently visible in the editor. You can move the translucent rectangle by dragging it with the pointer. The view inside the editor moves accordingly. Alternatively, click anywhere inside the Radar View to move the center of the translucent rectangle to where you have clicked. When you change the zoom level, the size of the translucent rectangle adjusts accordingly.
3	Zoom level adjustment. Dragging the corner inward or outward lets you zoom in or out respectively. NOTE: While the pointer is over the Radar View , you can also use the mouse wheel to zoom.

When the **Radar View** is collapsed, it appears as a button

Using the Simplified View in Template Editors

Overview

Zooming out to the lowest level in the **Composite Editor** or **Facet Editor** switches the display to the simplified view.

This view mode lets you see, at a glance, the elements of a template and the relation between them without showing the details of bindings and element properties.

Some template engineering functionality are not available.

To exit the simplified view, zoom in.

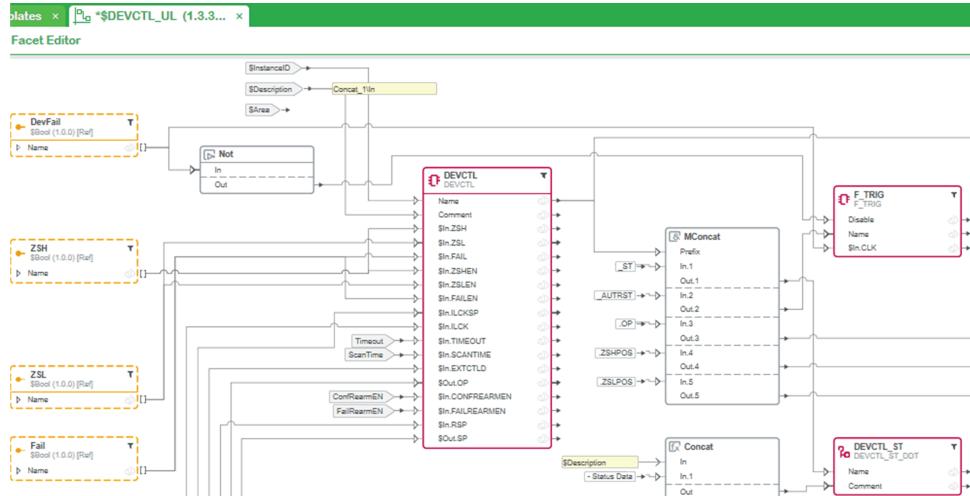
Restrictions of the Simplified View

The table describes the restrictions in the template editor while the simplified view is active.

Action or object	Restrictions
Actions related to bindings (such as creating, moving, deleting, switching styles).	Actions are not allowed
Creating parameters or platform inputs.	
Element properties and deferred inputs and outputs.	Are not shown.
Multiple bindings to and from an element.	Are shown as a single line.
Binding connectors.	Are not shown. Bindings are shown in line style instead.

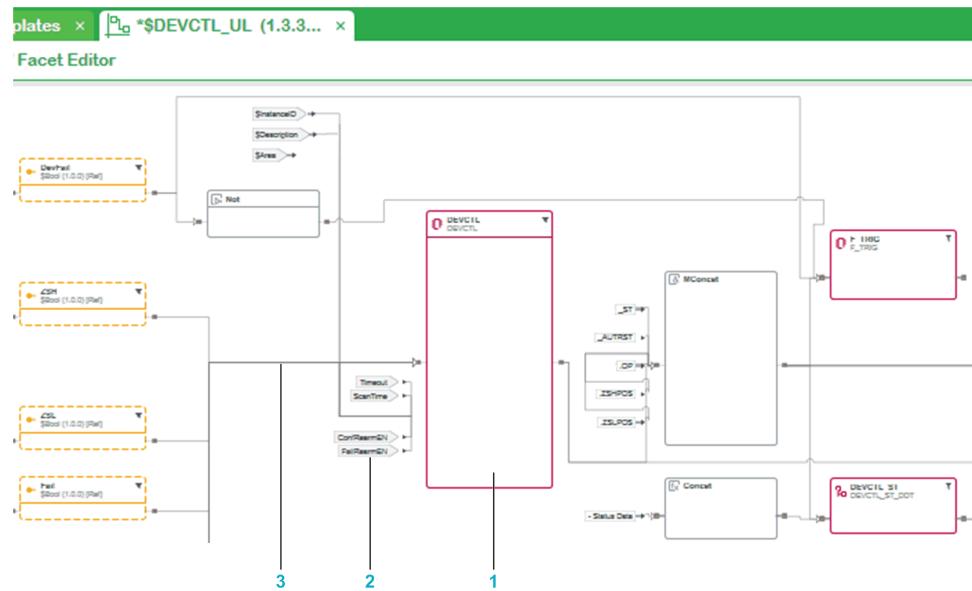
Description

The following figure shows an example of template as it normally appears in the **Facet Editor**.



Item	Description
1	Elements keep their relative size and position to preserve the aspect of the layout.
2	Parameters and platform inputs are shown in their actual position.
3	Bindings to and from each element are grouped into a single line.

The following figure shows the same template when simplified view is active (the view is enlarged for better visibility).



Managing Global Templates

What's in This Chapter

Managing Global Templates	881
Updating, Replacing, and Duplicating Global Templates	898

Managing Global Templates

Saving Changes In Global Templates

Overview

When Global Templates are open in their respective editor, page 853, you can save changes by using the following commands.

Save	Allows you to save changes to templates that you have created by using various save options. Use the command to rename a template or change its usability state. The command is available only if the template is not referenced by another template and not used by an instance of the application; otherwise the software opens the Save as dialog box.
Save as	Allows you to save changes to Schneider Electric Global Templates, create a new template, or a new version of a template.

When you select the **Save as** command, you can select one or more locations where to store the template.

If you have edited the template from within its parent, you have the possibility to update the parent template after closing.

NOTE: To save a template with a different identifier and/or version without editing it, use the **Duplicate** command, page 912.

Restrictions When Saving Schneider Electric Templates

Restrictions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) related to the template identifier apply when you save a Schneider Electric template.

Tracking Saved Changes

When you save a template, the software keeps a record of the save operation. You can view details in the **Changes log** pane, which is available in each template editor, page 874.

Save Options

The table describes the items of the **Save** and **Save as** dialog boxes that the software displays when you select the respective command.

Item	Description
Keep Version⁽¹⁾	Select this versioning scheme to save changes in a template without changing its version number. Selected by default.
New Build	For information on the usage of the different version components, refer to Version (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
New Minor	
New Major	By default, New Build is selected when you open the Save As dialog box.
Other	Selecting this versioning scheme allows you to edit the following parameters in the dialog box before saving the template: <ul style="list-style-type: none"> • The identifier. • The three version components (major, minor, build number). Select it to rename the template or create a new template based on the one you are editing. NOTE: By default, the version that is selected is the same as for New Build .
Version	For a description of the version format, refer to Version (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

Item	Description
Identifier	Within the Global Templates library, the software requires that the combination of identifier (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) and version be unique for each template.
Locations⁽³⁾	<p>Allows you to select one or more existing folders of the Global Templates library to store the template.</p> <p>Default value: The current location of the template. If linked copies, page 885 exist, their location is indicated also.</p> <p>Click the browse button to open the Select Template Location dialog box, which lets you change locations and create linked copies by selecting additional locations.</p>
Usability State⁽¹⁾	<p>For information on the parameter, refer to Usability State (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).</p> <p>Default value: Current state of the template.</p>
New Version Usability State⁽²⁾	<p>Allows you to define the usability state of the new template that you are creating.</p> <p>Default value: Not Approved</p> <p>For information on the parameter, refer to Usability State (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).</p>
Old Version Usability State⁽²⁾	<p>Allows you to define what will be the usability state of the template that you are currently editing after you create the new template.</p> <p>Default value: Approved</p> <p>For information on the parameter, refer to Usability State (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).</p>
Changes Description	<p>You must enter a description by using free-form text to be able to save changes.</p> <p>The description that you enter is visible in the Changes log of the template editor. Entering a detailed description of changes allows you to keep track of the version history of the template.</p>
<p>(1) The item is displayed only in the Save dialog box.</p> <p>(2) The item is displayed only in the Save As dialog box for templates that are already referenced or instantiated.</p> <p>(3) The item is displayed only in the Save As dialog box.</p>	

Saving Changes In Global Templates

Modifying Schneider Electric Global Templates or creating templates may affect the function of these templates and/or systems, and must be performed by qualified personnel. Before proceeding, refer to the Overview of this part, page 837.

To save changes in a Global Template, proceed as follows.

Step	Action
1	<p>In the template editor, click the Save or Save As button.</p> <p>Result: The software opens the appropriate dialog box.</p>
2	Select the versioning scheme that you want to use.
3	Edit the Identifier , Change Description , and/or Usability State as needed.
4	<p>Click the Save button.</p> <p>Result:</p> <ul style="list-style-type: none"> • For the Save dialog box: <ul style="list-style-type: none"> ◦ The dialog box closes. ◦ Changes are saved in the template. ◦ The template remains open inside the editor. • For the Save as dialog box: <ul style="list-style-type: none"> ◦ A new template, which includes your changes is created. It is opened in the editor.

Step	Action
	<ul style="list-style-type: none"> ◦ The source template is closed without saving. <p>NOTE: Click the Cancel button to close the Save or Save as dialog box without saving changes.</p>
5	Close the template. <p>Result: If you have edited the template from its parent and the parent is open in edit mode, the Update/Replace References dialog box opens. Otherwise, the template is closed.</p>
6	Click either button: <ul style="list-style-type: none"> • Yes: Closes the template and updates/replaces, page 899 all the references of it in the parent template. The parent template is shown. For each reference for which at least one binding cannot be recreated, a dialog box opens. It contains detailed information about the conflicts and lets you skip the update/replacement. • No: Closes the template. <p>NOTE: Other templates that also reference the template that you have edited are not updated.</p>

Copying and Pasting Global Templates and Folders

Creating Linked Copies of Templates

The **Copy** and **Paste** commands let you place copies of a template in other folders of the Global Templates library. The copies are linked to the source template. Any change that you make to the source template is also made to each copy that you have created.

NOTE: To create an independent copy of a template, in the same or a different folder, use the **Save as** or **Duplicate** command from the corresponding template editor or **Global Templates Explorer**.

Locating Copies

To locate the linked copies of a template, use the **Locations** context menu command, page 848.

Deleting Copies

To delete a linked copy of a template, use the **Remove** context menu command, page 848.

Copying Folders

You cannot paste an empty library folder.

When you copy a folder containing templates and you paste it on another folder, the software creates a subfolder with the same identifier, which contains a linked copy of each template of the source folders. Subfolders and their contents are created in the same way.

Exporting Templates

Overview

The **Export** command of the **Global Templates Explorer** lets you export templates of the Global Templates library.

The process creates an export file in **.sbk** format that contains:

- The selected templates.
- Referenced composite and facet templates, interfaces, and their parameter configuration.
- Local constituents.
- Supervision content.

User contents such as content containers, page 937 containing documents and URLs that are linked to a template are not exported.

The export functionality allows you to copy these templates to another system or platform, or to create a backup.

Aborting Export Tasks

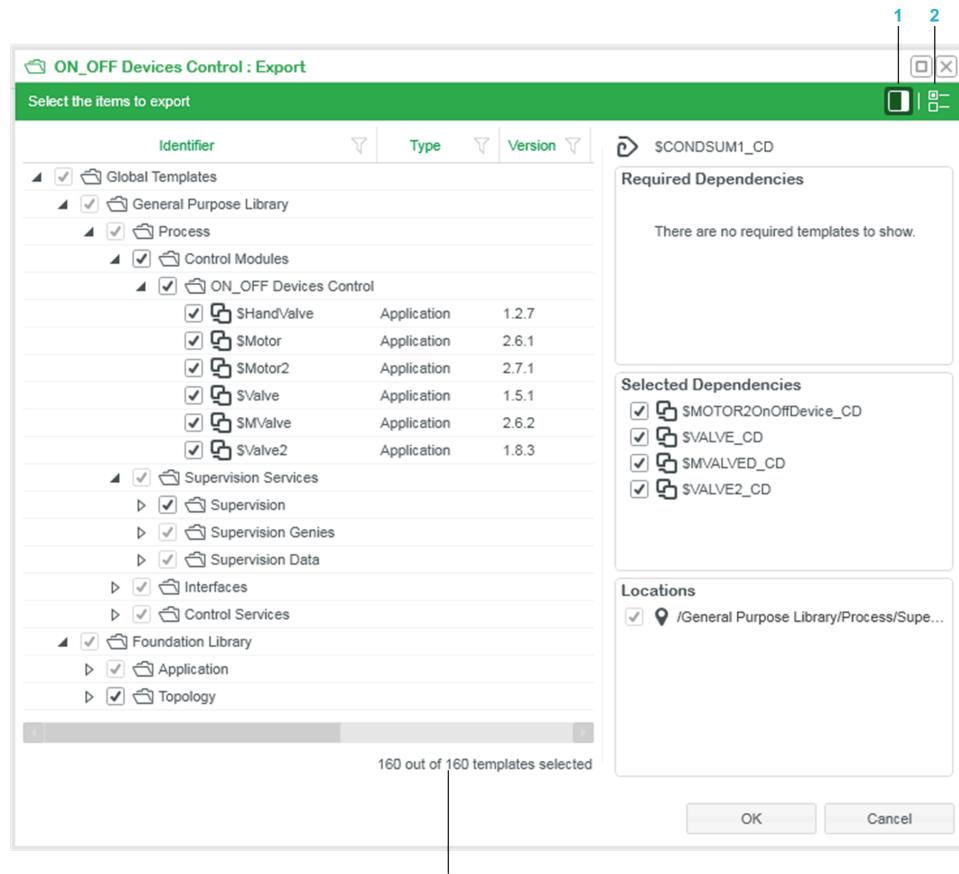
After confirming the export by clicking **Save** in the **Save** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. The export file is not created.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are exporting only a few templates, the task may complete before you are able to click the icon.

Template Export Window

The following figure shows an example of the template window that opens after selecting the **Export** command from the *ON_OFF Devices Control* sample folder.



Item	Description
1	Button to show or hide the details pane (shown on the right-hand side of the window). Select a template by clicking its row in the tree or grid view to view details.
2	Button to toggle between grid view and tree view. Tree view is the default view (shown here).
3	Indicates the number of selected templates and the total number of templates contained in the location from which you have selected the Export command. Linked copies, page 885 are not counted even if they are selected.

NOTE: To expand/collapse a node in the **Export** window, select the node and press **Enter**.

The table describes the information that appears in the columns of the **Export** window.

Column Header	Description
Identifier	<p>Displays:</p> <ul style="list-style-type: none"> The folder hierarchy in the Global Templates library (tree view mode only). The identifier of templates contained in the location from which you have selected the Export command. Check boxes that let you select the templates you want to export. <p>In grid view mode, only templates are shown, not linked copies. NOTE: By default, all templates are selected.</p>
Type	<p>Indicates the category of the template.</p> <p>Refer to the definition header of each template category for a description of the type, page 22.</p>
Version	Indicates the version of the template.
State	Usability state (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of the template (for example, Approved or Deprecated).
Description	Description of the template.

The table describes the details pane of the **Export** window.

Section Header	Description
—	<p>Identifier of the selected template. NOTE: You can view details even if the check box next to the template identifier is not selected.</p>
Required Dependencies	<p>Identifier of templates that are required to export the selected template.</p> <p>The check box of required dependencies is selected and disabled. To enable it, you need to clear the check box of the selected template.</p>
Selected Dependencies	<p>Identifier of templates that reference the selected template.</p> <p>When you clear the check box of a selected dependency, the template is removed from this section. Once you have cleared the check boxes of all the listed templates of this section, the check box of the selected template is enabled and you can clear it to exclude it from the export file.</p> <p>As long as selected dependencies are listed, the check box of the selected template remains disabled.</p>
Locations	<p>Path of the selected template and any of its linked copies in the Global Templates library.</p> <p>For each template and linked copy, the status of the check box in the Identifier column is duplicated.</p>

Creating the Export File

To export templates and create the export file, proceed as follows.

Step	Action
1	From the tree view of the Global Templates Explorer , right-click and select Export on: <ul style="list-style-type: none">• A template if you want to export the template only• A template folder if you want to export any templates of the folder, subfolders, and any templates contained in these subfolders Result: The Export window opens.
2	Select the templates that you want to export.
3	Click OK . Result: The Save dialog box opens.
4	Enter a name for the export file and select a location, and click Save . Result: The software exports the selected items and creates an export file in <i>.sbk</i> format in the location that you have selected.

NOTE: To export the templates of an entire library, create separate export files by selecting various subfolders in the **Export** window. This creates smaller export files that take less time to import.

Importing Templates

Overview

The **Import** command of the **Global Templates Explorer** allows you to import new templates, new versions of existing templates, and to modify or update templates or parts of these in the Global Templates library. Importing templates adds them to the database that is mounted.

When you select an export file (.sbk) containing templates, the software compares the information in the export file with the templates contained in the Global Templates library and applies rules to keep templates unique.

The **Import** window allows you to select which valid templates of the export file you import. It also allows you to import new and/or updated content that is required by the templates you are importing. Content are project files containing, for example, Supervision resources such as animated graphics (genies) used by Supervision runtime.

NOTE: Importing many templates may take time.

Export File Location

The export file containing the template information needs to be located in a folder that you can write to.

Import Rules

The table describes the rules that the software applies while comparing templates contained in the export file with those in the Global Templates library before importing templates.

Template identifier	Template version	Template signature	Action
Does not exist in the Global Templates library.	Indifferent	Indifferent	The template is imported.
Already exists in the Global Templates library.	Already exists in the Global Templates library.	Already exists in the Global Templates library.	The template is not imported.
		The signature is different.	The template is not imported and it displays a message about the detected conflict.
	The version is unique.	The signature is unique.	The template is imported.

Importing Supervision Included Projects

An included project (.ctz) that is used by a template is only imported if the template itself is imported.

For example, if you have modified the included project that is used by the *MyMotor* template v1.0.0 but without making any change to the template itself, you cannot add the modified included project to the content repository, page 936 by importing the template if *MyMotor* v1.0.0 already exists in the Global Templates library. Since the template already exists, it is not imported.

The table describes the methods that you can use to add a new included project to the content repository.

Method	Description
Modify the template that references the new included project so that it can be imported.	<p>Modify at least one Supervision genie facet template (_CG) of the template.</p> <p>For example, you can create a new version, page 882 of the genie facet template without changing its content by using the Save As command.</p> <p>Importing the template allows you to import the new included project and add it to the content repository.</p>
Add the new included project directly to the content repository.	<p>Refer to the procedure that describes how to add project files to the content repository (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).</p> <p>To follow the procedure, edit either facet template:</p> <ul style="list-style-type: none"> • A new Supervision genie facet template that you create by using the Genie template from the Toolbox of the Global Templates Explorer. • Any existing Supervision genie facet template (_CG) of the Global Templates library. <p>In both cases, you do not need to save the template to be able to add the included project to the content repository.</p>

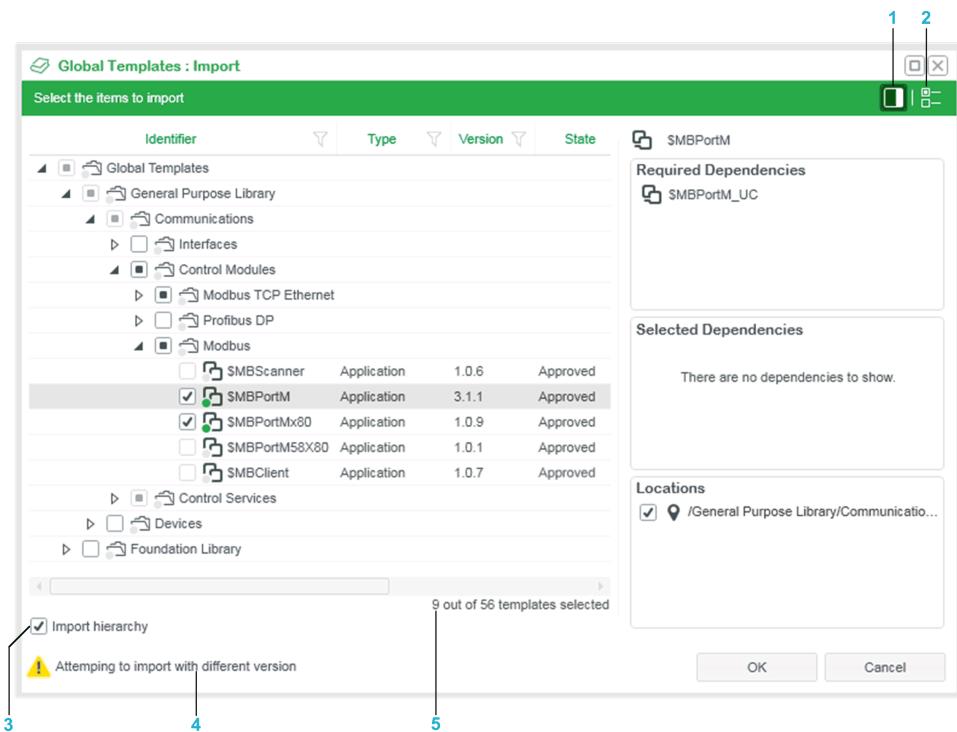
Accessing the Template Import Window

To access the **Import** window, proceed as follows.

Step	Action
1	<p>From the client toolbar, click the Global Templates explorer icon.</p> <p>Result: The Global Templates Explorer window opens.</p>
2	<p>Right-click the Global Templates root folder and select Import.</p> <p>Result: The Import dialog box opens.</p> <p>NOTE: If you know the exact location where you want to import the templates, you can right-click the corresponding folder in the tree view of the Global Templates Explorer and select Import.</p>
3	<p>Select the export file (.sbk) containing the templates that you want to import and click Open.</p> <p>Result: The software opens the Import window and displays the contents of the export file.</p>

Template Import Window

The following figure shows an example of the **Import** window.



Item	Description
1	Button to show or hide the details pane (shown on the right-hand side of the window). Select a template by clicking its row in the tree or grid view to view details.
2	Button to toggle between grid view and tree view, which is shown here. (default view).
3	Imports templates with the hierarchy of folders that existed at the time of export. Clear the check box to import templates in the same folder without hierarchy. The check box is selected by default. While in grid view, selecting or clearing the check box switches to tree view.
4	A message appears when the templates were exported from a version of the software that is different from the version you are using.
5	Indicates the number of selected templates and the total number of templates contained in the export file. Linked copies (templates marked with a blue dot) are not counted even if they are selected.

NOTE: To expand/collapse a node in the **Import** window, select the node and press **Enter**.

The table indicates the meaning of the colored dots that appear in the **Import** window next to template identifiers.

Color code	Description
Green	The template does not exist in the Global Templates library and can be imported. For a node, indicates that the elements of the node can be imported.
Gray	The template already exists in the Global Templates library and cannot be imported. For a node, indicates that at least one element of the node already exists and is identified by a gray dot.
Blue	The template already exists in the Global Templates library but a linked copy, page 885 is created if imported.
Red	Red dot: The template cannot be selected for import because of a detected signature conflict, page 889.

The table describes the information that appears in the columns of the **Import** window.

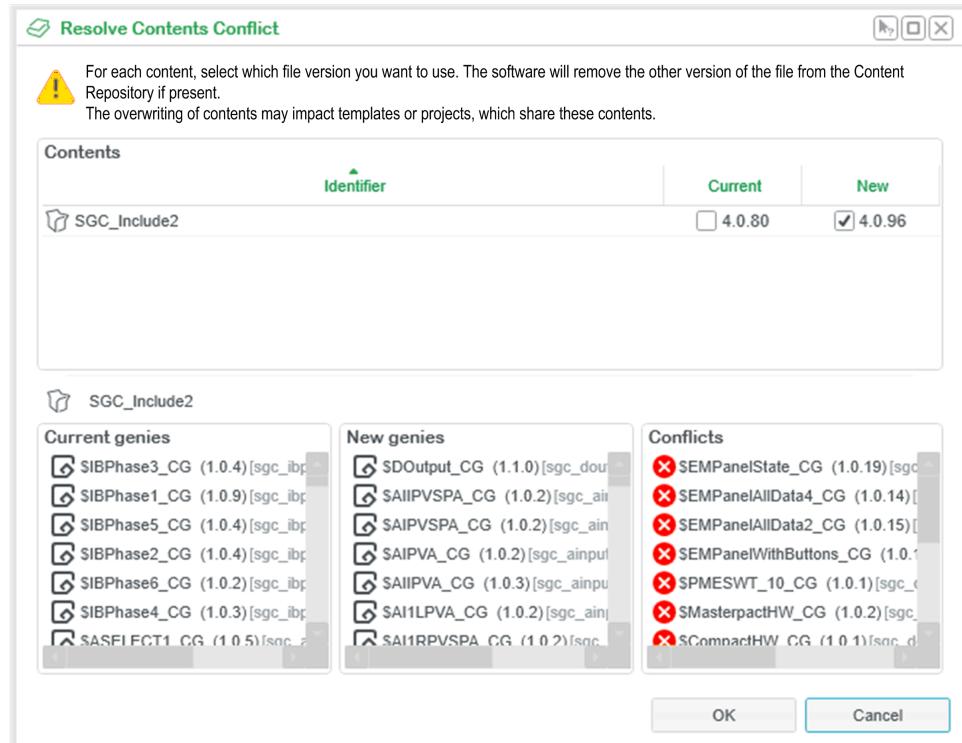
Column Header	Description
Identifier	<p>Displays:</p> <ul style="list-style-type: none"> The folder hierarchy at the time of export (when Import hierarchy is selected). The identifier of templates contained in the export file. A colored dot. Check boxes allowing you to select the templates that you want to import. <p>Selecting a template automatically selects any referenced templates that can be imported. It also selects linked copies of referenced templates if the referenced template does not exist in the Global Templates library yet.</p> <p>In tree view, when Import hierarchy is not selected, the software can only import one of each template and/or one linked copy of each existing template because any template or linked copy is imported to the same folder.</p> <p>In grid view, only templates are shown, not linked copies. If a template already exists, its check box is cleared and disabled. Templates that do not exist have their check box selected by default and enabled (you can clear it).</p> <p>NOTE: By default, new templates and new template versions are selected.</p>
Type	Indicates the category of templates in the export file. Refer to the definition header of each template category for a description of the type, page 22.
Version	Indicates the version of templates in the export file.
State	Usability state (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of the template (for example, Approved or Deprecated).
Description	Description of templates in the export file.

The table describes the details pane of the **Import** window.

Section Header	Description
–	<p>Identifier of the selected template.</p> <p>NOTE: You can view details even if the check box next to the template identifier is not selected.</p>
Required Dependencies	Identifier of templates that are referenced by the selected template.
Selected Dependencies	<p>Identifier of templates that reference the selected template (only if the selected template can be imported).</p> <p>When you clear the check box of a selected dependency, it clears the check box of any template and linked copies referencing the selected template. This allows you to clear the check box of the selected template, which would be disabled otherwise.</p>
Locations	<p>Path at the time of export of the selected template and any of its linked copies contained in the export file.</p> <p>For each template and linked copy, the status of the check box in the Identifier column is duplicated.</p> <p>NOTE: If you clear Import hierarchy:</p> <ul style="list-style-type: none"> Linked copies are not shown anymore. The path does not reflect the hierarchy at the time of export anymore

Resolving Contents Conflicts

When the software detects contents with the same name but a different version in the export file compared to the content that exists in the content repository, it displays the **Resolve Contents Conflict** dialog box. The following figure shows an example of the dialog box.



Header	Description
Identifier	Name of the content that conflicts with the current content located in the content repository.
Current	Version of the current content located in the content repository.
New	Version of the content in the export file.
Current genies	Animated graphics (genies) that are included in the current content located in the content repository.
New genies	Animated graphics that are included in the content in the export file.
Conflicts	Animated graphics that create a conflict if you import the new content.

In the **Resolve Contents Conflict** dialog box, verify the following before proceeding with the import:

- Refer to the library release notes. The documents contain information about changes that were made to templates of the corresponding library.
- Considerations when importing content:

Selecting new content	<p>Templates that you are importing and that require the new content will work correctly; however, the current content that is used by existing templates of the Global Templates library and instances thereof may be overwritten.</p> <p>Also, the status of executables of Supervision projects that use contents of which you are importing a new version may be set to Out Of Date if it is currently Built. In such case, you need to build these executables to integrate changes of the content in the corresponding Supervision projects.</p>
Selecting current content	The content that is used by existing templates of the Global Templates library and instances thereof is not affected; however, templates that you are importing will be missing the new content that they require to work properly.

NOTE: Refer to the **Conflicts** section of the **Resolve Contents Conflict** dialog box.

Aborting Import Tasks

After confirming the import by clicking **OK** in the **Import** window or in the **Resolve Contents Conflict** dialog box, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. No templates and no content are imported.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are importing only a few templates, the task may complete before you are able to click the icon.

Importing Templates

NOTICE

LOSS OF DATA

Resolve a content conflict that arises during the import of templates by selecting the correct version of the content to be imported.

Failure to follow these instructions can result in unusable Supervision projects.

To import templates from an export file (.sbk) into the Global Templates library, proceed as follows.

Step	Action
1	In the Import window, select the templates, page 891 that you want to import.
2	Click OK . Result: If software detects content with the same name but of a different version, it displays the Resolve Contents Conflict dialog box; otherwise the software proceeds with the import of selected templates from the export file. NOTE: Click Cancel to close the Import window without importing templates.
3	If the software displays the Resolve Contents Conflict dialog box, select the version of the content that you want to use, and click OK . Result: <ul style="list-style-type: none"> • The selected templates are imported. • The import settings that you have selected are applied. • It displays Completed in the notification panel when the import is complete. NOTE: Click Cancel to close the Resolve Contents Conflict dialog box without importing the template. This reverts to the Import window.

After Importing Templates

Depending on the number of templates that you are importing and your hardware configuration, the import process may affect the performance of your computer, requiring you to restart the computer.

NOTICE

LOSS OF COMMUNICATION

Stop the system server only when no client is running.

Failure to follow these instructions can result in unresponsive clients.

After importing templates, proceed as follows.

Step	Action
1	Quit clients, page 59.
2	Stop the system server.
3	Restart the computer.

Purging the Global Templates Library

Overview

The **Purge** context menu command of folders of the **Global Templates Explorer** allows you to remove unused, earlier versions of templates and interface models from the Global Templates library. The process also removes local constituents of deleted templates and unused global constituents, page 937.

You can select the templates to remove.

At the end of the purge process, the notification panel provides a summary with information on purged templates.

If the process ends abnormally (for example, by terminating the process), the tool rolls back the process.

Purge Rules

The tool deletes, using an iterative process, any facet, and composite templates, as well as interface models that satisfy the following rules:

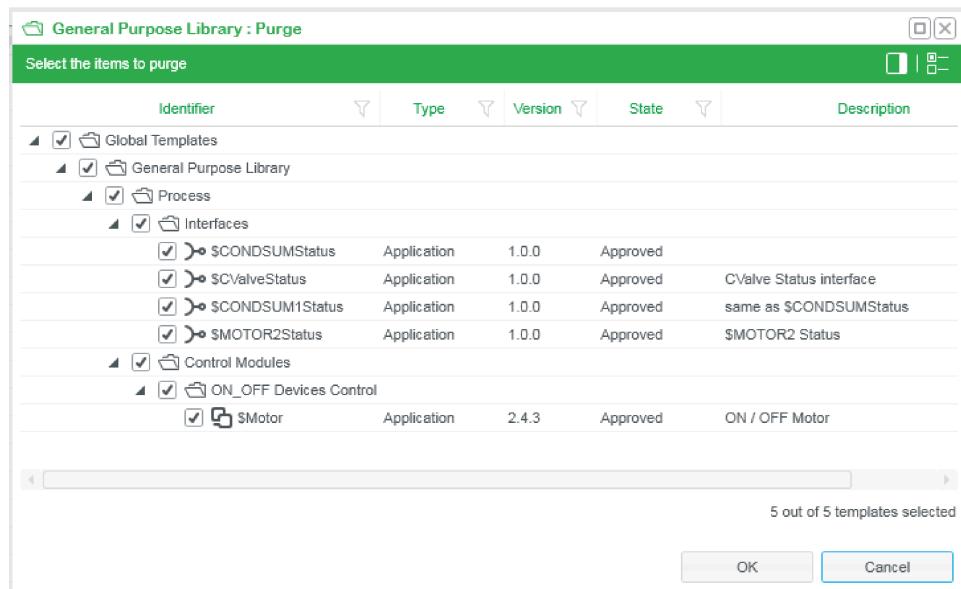
- A template with a higher version number, having the same identifier and the Usability State (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) **Approved** exists in the Global Templates library.
- The template or the interface model is not referenced (used) by any other template of the Global Templates library.
- The template is not used by any instance of the application.

NOTE: The tool does not delete a template if the template is used by a facet that is assigned to a project and the **Assignment** status of the facet is **Deleted**, page 411.

To delete such template, first generate the project or the container to which the facet is assigned.

Purge Window

The following figure shows an example of the **Purge** window when the **Purge** command is selected from the **General Purpose Library** folder.



Aborting Purge Tasks

After confirming the **Purge** command by clicking **OK** in the **Purge** window, the abort icon is displayed in the notification panel, page 73. Click the icon to cancel the task. No templates are purged.

Only the user who has selected the command is allowed to abort it.

NOTE: If you are purging only a few templates, the task may complete before you are able to click the icon.

Backing Up the Database

You cannot restore Global Templates that you delete from the Global Templates library by using the **Purge** command.

Before using the command, back up the database. Refer to the topic describing how to back up the database (see EcoStruxure™ Process Expert, Installation and Configuration Guide).

Purging Global Templates

NOTICE	
LOSS OF DATA	
Back up the database before using the Purge command. Failure to follow these instructions can result in unusable systems or database.	

To delete Global Templates that satisfy the purge rules, proceed as follows.

Step	Action
1	Back up the database.
2	Right-click the parent folder starting from which you want to proceed with the removal and select Purge . Result: The Purge window opens. After a few moments, it displays the templates that satisfy the purge rules in the selected location.
3	Select the templates that you want to delete and click OK . Result: The software: <ul style="list-style-type: none"> • Deletes applicable templates and interface models. • Displays information in the notification panel when the purge is complete. NOTE: Click Cancel to close the Purge window without purging Global Templates.

Updating, Replacing, and Duplicating Global Templates

Overview

This chapter describes the various commands that are available to modify Global Templates with or without editing them.

Template Modification Strategy

Overview

Various commands are available that allow you to modify Schneider Electric and/or user-created templates. Select the appropriate command depending on:

- Whether you want to edit a template or not.
- Whether the identifier of the template or one of its child templates starts with the \$ prefix (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
- The scope of changes to be performed.

Starting with EcoStruxure Process Expert 2021, when you edit a template from within its parent, you have the possibility to update its references, page 899 in the parent template after you save changes and close.

The table shows the commands that are available to modify Global Templates and their specifics.

Command	Use in	Use with	Purpose	Comment
Update	Global Templates Explorer	Templates	To update a template with the latest version of any child templates that exist in the Global Templates library.	<ul style="list-style-type: none"> • Cannot be used to update templates whose identifier starts with the \$ prefix. • The software finds the latest versions of templates with the same identifier automatically. • When used at the folder level, you can select which templates to update.
		Folders	To update the templates inside a folder, page 905, and optionally inside its subfolders, with the latest version of any child templates that exist in the Global Templates library.	
	Template editors	Elements	To update the template that is used by a specific element with the latest version that exists in the Global Templates library.	<ul style="list-style-type: none"> • Lets you update the template that is used by a specific element (instance). • The software finds the latest version automatically.
Replace	Template editors	Elements	To update the template that is used by a specific element with a different version or a different template that exists in the Global Templates library.	<ul style="list-style-type: none"> • Lets you replace the template that is used by a specific element (instance). • Can be used to undo a template update.
Duplicate	Global Templates Explorer	Templates	To create a new template from an existing one with the possibility to keep, duplicate, or replace each child template that is referenced.	<ul style="list-style-type: none"> • The replace action impacts all elements (instances) of a given template. • Can be used in place of the Update command with templates whose identifier starts with the \$ prefix.
Edit/Extend Interface	Facet and composite template editors	Interface references	To edit, add, and remove elementary elements in an interface, page 917 from within its parent template.	<ul style="list-style-type: none"> • Provides basic modification capability. • Updating templates that use the other role of the interface is required. • The command is not available for deferred interfaces and when the interface element contains a nested interface.

Updating References in Parent After Editing Child Templates or Interfaces

When you edit a template or interface by opening it from within its parent, after saving changes and closing it, page 882, the **Update/Replace References** dialog box opens. It lets you automatically update in this parent template all the references of the template/interface that you have edited. This requires that the parent template is open in a template editor in edit mode. If it is open in read-only mode, you can switch it to edit mode.

If you rename the template while saving changes, references are replaced instead of being updated.

Once you save changes and close the parent, you are again given the choice to update the next higher level template given it is open, and so on. Refer to the example, page 922 for details.

The impact of the template update/replacement is the same as when you use the **Update or Replace** command.

NOTE: Other templates that also reference the template that you have edited are not updated.

NOTE: The **Update/Replace References** dialog box opens also when you edit/extend an interface in-place, page 917 or edit a nested interface.

Impact of Template Update or Replacement

The table describes the impact on the various template components when you update or replace a template.

Current template component	In new template	Impact
Bindings to the element selection of composite references	Are the same as in current template.	The software maintains the bindings.
	Are not the same as in current template.	The software removes the bindings to the element selection of the composites that are not part of the new template.
Bindings and attributes of the element parameters	Are the same as in current template.	The software maintains the bindings and attributes.
	Are not the same as in current template.	The software removes the bindings and attributes of the element parameters that are not part of the new template.
Bindings to element interfaces	Are the same as in current template.	The software maintains the bindings.
	Are not the same as in current template.	The software removes the bindings to the element interfaces that are not part of the new template.

Applying Template Modifications to Existing Instances

Modifications to templates need to be propagated to instances of the application to take effect. The method to apply modifications to existing instances depends on the template identifier.

Result of template modification	Method to apply
A new version of the template (same identifier)	Update the template used by instances, page 212
A template with a new identifier	Replace the template used by instances, page 212

Updating Global Templates and Templates of Elements

Overview

The **Update** command lets you perform different actions depending on where you use it.

In the **Global Templates Explorer**, it lets you create a new version of your template, in the same location, by using the latest approved version of any of the child templates that are referenced by this template.

In the **Facet Editor** and the **Composite Editor**, it lets you update the template or interface model that is used by a specific element by the latest available version.

In both cases, the update process allows you to integrate in the composition of your template the latest versions of composite and facet templates, and interface models.

NOTE: The update of the template of an element impacts templates that reference this element and requires updating, page 899 all the higher-level templates.

Restrictions When Updating Schneider Electric Templates

Restrictions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) related to the identifier of the parent template apply when you update a Schneider Electric template.

You cannot use the **Update** command in the **Global Templates Explorer** on a template whose identifier contains the \$ prefix. You need to use the **Duplicate** command, page 912 instead.

Good Practices

- If you know which highest level templates reference templates of which a later version is available, proceed with the update from these highest level templates. Typically, the highest level template is the control module template.
- If you do not know which highest level templates need to be updated, use the **Inspect > Used By** context-menu command on the template of which you have created a new version. This allows you to determine which are the highest level templates that you need to update with the new version child template.

Working Principle of the Update Functionality

When the **Update** command is used on a template in the **Global Templates Explorer**, the software proceeds as follows:

- Searches the tree of dependencies of the template for any element with the same identifier and for which a later version exists in the Global Templates library.
- Displays in a window the latest version of templates that it has found in the dependencies and lets you select:
 - The usability state for the current version of templates after the new versions are created.
 - The versioning scheme (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) and usability state for the new version of templates that are created.
- Once you proceed, creates the new templates, proceeds with the update, and processes bindings (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).

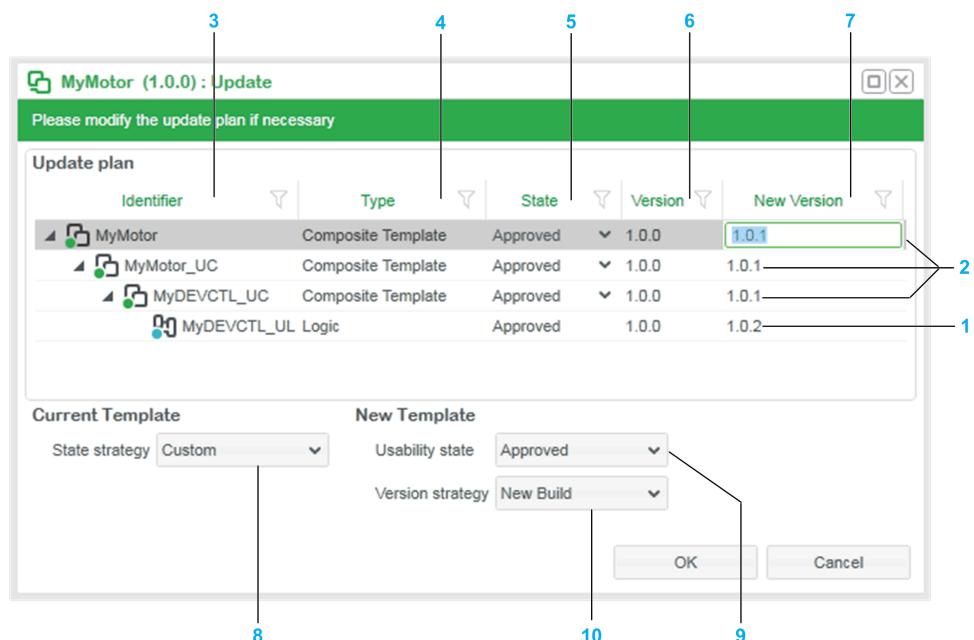
When the **Update** command is used in a template editor, the software proceeds as follows:

- Verifies if a template with a later version exists in the Global Templates library.
- If so, processes bindings and:
 - If any binding cannot be recreated, informs you and gives you the choice to proceed or cancel the update.
 - If bindings can be recreated, proceeds with the update of the template.

NOTE: For the template update process to complete, the latest version of the template or interface model must have the **Usability State Approved** or **Deprecated**.

Update Window

The following figure shows an example of the **Update** window in a scenario where the **Update** command was used on the *MyMotor* (1.0.0) control module template in the **Global Templates Explorer**. The template references *MyMotor_UC* (1.0.0). This UC composite template references *MyDEVCTL_UC* (1.0.0), which in turn references the *MyDEVCTL_UL* (1.0.0) facet template. *MyDEVCTL_UL* versions 1.0.1 and 1.0.2 exist in the Global Templates library.



Item	Description
1	Template of which the software has found a later version in the Global Templates library compared to the version that is currently used.
2	Templates of which the software will automatically create a later version because the version of their child template is being incremented.
3	Shows the hierarchy of the templates that will be updated and the templates of which a later version exists in the Global Templates library. Other child templates whose version is unchanged are not shown. At the top level, identifier of the template from which you have selected the Update command.
4	Type of the template.
5	Usability state of the current version of the template after the software has created the new version based on the value of the State strategy parameter. You can select it individually for templates of which the software creates a new version.
6	Current version of the template from which you have selected the Update command and current version of its child templates.
7	Version of the template after the update and latest version of child templates with the usability state <i>Approved</i> that the software has found in the Global Templates library. For parent templates, the version is created by the software during the update based on the value of the Version strategy parameter.

Item	Description
	<p>NOTE: You can modify the version number (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) manually for each template version that the software creates during the update. Modifying the version number manually sets the Version strategy parameter to Custom. You cannot enter the same value as in Version.</p> <p>NOTE: If Version for a template that the software will update is already 99.99.9999, New Version is blank and you need to enter a value manually.</p>
8	<p>Allows you to select the usability state of the current version (Version column) of the templates after the software has created a new version of them.</p> <p>Default value: <i>Custom</i></p>
9	<p>Allows you to select the usability state of the new version of the templates after the update.</p> <p>Default value: <i>Approved</i></p> <p>NOTE: The usability state applies only to new template versions that the software creates.</p>
10	<p>Allows you to select the versioning scheme (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) for new template versions that the software creates.</p> <p>Default value: <i>New Build</i></p> <p>NOTE: The version strategy applies only to new template versions that the software creates.</p>

Updating Global Templates in the Global Templates Explorer

Updating Schneider Electric or user-created templates may affect the function of these templates and must be performed by qualified personnel. Before proceeding, refer to the Overview, page 837.

To update Global Templates, proceed as follows.

Step	Action
1	<p>In the tree view of the Global Templates Explorer, right-click the template that you want to update and select Update.</p> <p>Result: The Update window opens.</p>
2	<p>For the current template version and its elements, select the usability state strategy that you want the software to apply from the menu of the State strategy parameter.</p> <p>NOTE: If you select Custom, select the usability state for each template in the State column.</p>
3	<p>For the new template version and its elements, select the usability state strategy that you want the software to apply from the menu of the Usability State parameter.</p>
4	<p>For the new template version and its elements, select the version strategy that you want the software to apply from the menu of the Version strategy parameter.</p> <p>NOTE: You can modify the version number manually for each template that the software will create by double-clicking the New Version field:</p> <ul style="list-style-type: none"> • If the version number that you have entered is invalid, the software applies the default versioning scheme for new templates. • To undo your version change, select New Build as the version strategy. This changes the version of new templates to the default value.
5	<p>Click OK.</p> <p>Result: The software:</p> <ul style="list-style-type: none"> • Proceeds with the update. • Displays the new templates that it has created in their respective folders within the Global Templates library. • Displays Completed in the notification panel when the update process is completed. <p>NOTE: Click Cancel to close the window without proceeding with the update.</p>

Updating Global Templates in Template Editors

Updating Schneider Electric Global Templates or user-created templates may affect the function of these templates and must be performed by qualified personnel. Before proceeding, refer to the [Overview](#), page 837.

To update the template or interface model that is used by an element of a facet or composite template, proceed as follows.

Step	Action
1	In the template editor, right-click the element whose template or interface model you want to update and select Update . Result: If a later version of the same template or interface model with the required usability state exists in the Global Templates library, the software processes binding information; otherwise, the software indicates that the template is already up-to-date.
2	Result: If any binding cannot be recreated, the Replace Conflicts dialog box opens; otherwise, the update of the template completes and bindings are recreated.
3	Verify the information that is displayed in the Replace Conflicts dialog box and click: <ul style="list-style-type: none"> • Yes: Proceeds with the update of the template, discards conflicting bindings, and recreates the other bindings. • No: Closes the Replace Conflicts dialog box without updating the template.
4	Recreate bindings as needed.
5	Save the change, page 882.

NOTE: To undo a template update that you have performed in a template editor, you can use the **Replace** command, page 907.

Updating Global Templates at the Folder Level

Overview

The **Update** command that appears in the context menu of folders of the **Global Template Explorer** lets you create a new version of templates contained in a folder, and optionally, in its subfolders.

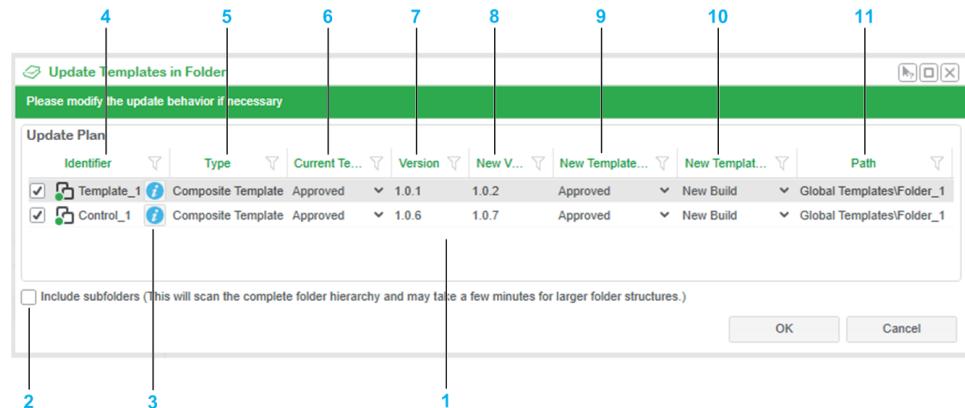
Restrictions When Updating Schneider Electric Templates

Restrictions related to the identifier (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) of the parent template and its elements apply when you update a Schneider Electric template.

When you use the **Update** command at the folder level and the folder contains a template whose identifier starts with the \$ prefix, the template is not updated. You need to edit the template and save it with a new identifier or use the **Duplicate** command, page 912.

Update Templates in Folder Window

The following figure shows an example of the **Update Templates in Folder** window.



Item	Description
1	List of templates in the folder for which a later version of at least one child template is available in the Global Templates library. NOTE: If a folder contains several versions of the same template and more than one can be updated, only the highest version will be shown and can be updated. This applies also if some of the versions that can be updated are in a subfolder and you have selected to include subfolders; only the highest version across all folders will be shown.
2	Lets you include templates located inside subfolders. Their location is indicated in the Path column.
3	Click the button to view details about which newer version child templates will be used to update the template.
4	Identifier of the templates that are candidate for the update. The operation is performed only on templates whose checkbox is selected.
5	Type of the template.
6	Usability state of the current version of the template after the software has created the new version.
7	Current version of the template.
8	Version of the template after the update. The version is based on your selection for New Template Version strategy .

Item	Description
	<p>NOTE: You can modify the version number (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) manually. This sets New Template Version strategy to <i>Custom</i>. You cannot enter the same value as it appears in Version.</p> <p>NOTE: If Version of a template to be updated is 99.99.9999, New Version is blank and you need to enter a value manually.</p>
9	<p>Allows you to select the usability state of the new version of the template after the update.</p> <p>Default value: <i>Approved</i></p>
10	<p>Allows you to select the versioning scheme (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) for the new template version.</p> <p>Default value: <i>New Build</i></p>
11	Location of the template in the Global Templates library.

Updating Templates at the Folder Level

To update the templates contained in a folder, proceed as follows.

Step	Action
1	<p>In the Global Template Explorer, right-click a folder and select Update.</p> <p>Result: The Update Templates in Folder window opens and displays the templates for which a child template with a later version exists in the Global Templates library.</p> <p>NOTE: The command is not available from the Global Templates library root folder.</p>
2	Select to display also templates of subfolders if needed.
3	Select which templates you want to update.
4	<p>For each one, select the appropriate usability states and versioning schemes for the current and new template versions and click OK.</p> <p>Result: The selected templates are updated and appear in the tree view of the Global Templates library. A message in the notification panel indicates the status of the operation.</p>

Replacing the Template of an Element of a Global Template

Overview

In the **Facet Editor** and **Composite Editor**, the **Replace** context-menu command allows you to replace the template that is used by an element with another template of the Global Templates library.

You can replace the template of the following types of elements:

- Composite templates
- Facet templates
- Interfaces

The software informs you if it detects binding-related conflicts before proceeding with the replacement of the template.

The replacement of the template of an element impacts templates that reference this element and requires updating, page 899 all the higher-level templates.

Restrictions When Replacing Schneider Electric Templates

Restrictions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) related to the identifier of parent templates apply when you replace a template.

Before Starting

Depending on the situation, consider the following before starting:

- To replace the template of an element by the latest version of the same template, use the **Update** command, page 901 instead.
- You can use the **Replace** command to undo the update of the template of an element, page 901. Undoing a template update does not restore bindings that the software was not able to recreate.

Replace Dialog Box

For a description of the template browser that opens, refer to the topic describing the **Replace** dialog box, page 916.

Replacing The Template That Is Used by an Element

Modifying Schneider Electric Global Templates or user-created templates may affect the function of these templates and must be performed by qualified personnel. Before proceeding, refer to the Overview, page 837.

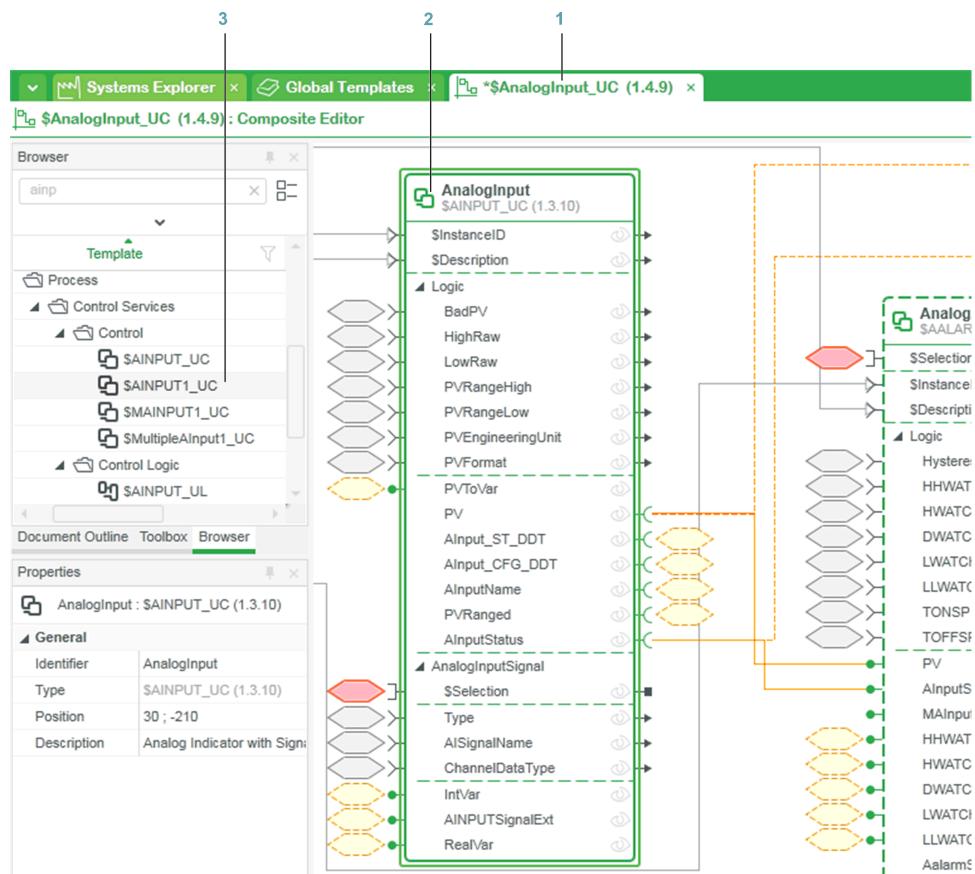
To replace the template that is used by an element of a Global Template, proceed as follows.

Step	Action
1	<p>In a template editor, right-click the element whose template you want to replace and select Replace.</p> <p>Result: The Replace dialog box opens.</p> <p>NOTE: Alternatively, you can select the template in the Browser pane and drag it onto the header of the element until Drop to update the reference with x is displayed in a tooltip (where x represents the name and version of the template that you want the element to use).</p>
2	<p>Select the template that you want the element to use instead and click OK.</p> <p>Result: The Replace confirmation dialog box opens.</p>
3	<p>Verify the information that is displayed and click Yes.</p> <p>Result: If any binding cannot be recreated, the Replace Conflicts dialog box opens; otherwise, the replacement of the template completes and bindings are recreated.</p>

Step	Action
4	Verify the information that is displayed in the Replace Conflicts dialog box and click: <ul style="list-style-type: none"> Yes: Proceeds with the replacement of the template, discards conflicting bindings, and recreates the other bindings. No: Closes the Replace Conflicts dialog box without replacing the template.
5	Recreate bindings as needed.
6	Save changes, page 882.

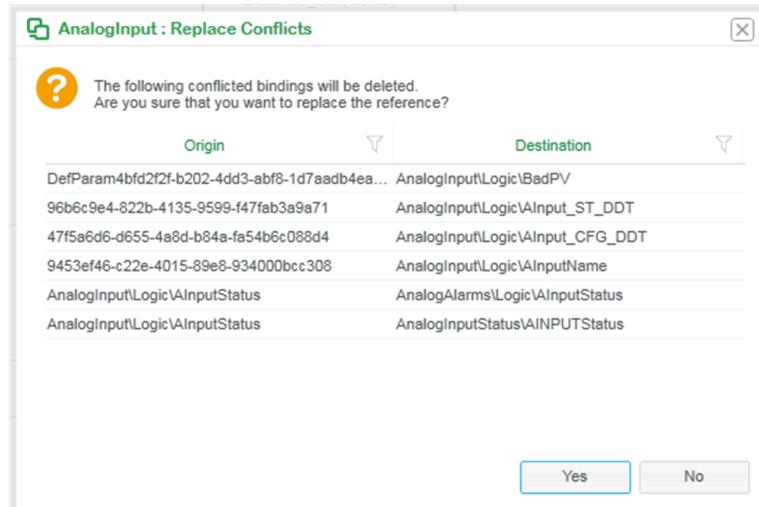
Example

The following example illustrates the replacement of the template of a composite element, which is referenced by a parent composite reference. The highest level composite template being **\$AnalogInput** (control module).

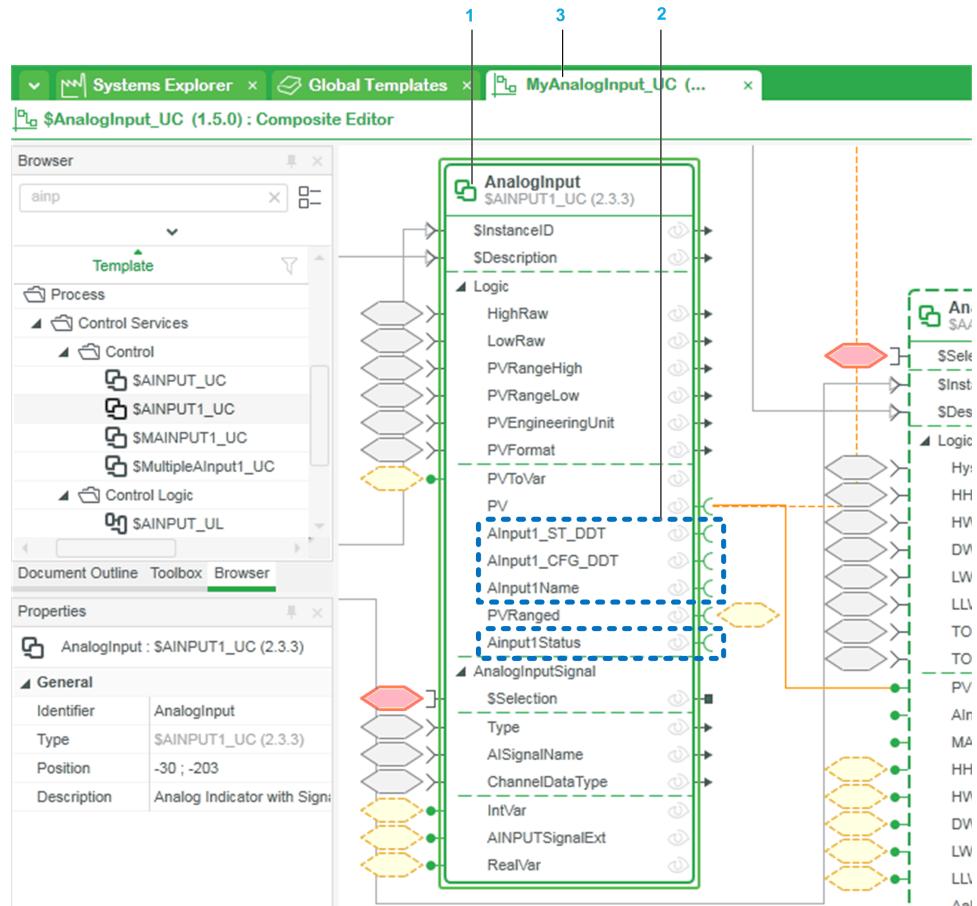


Item	Description
1	\$AnalogInput_UC [1.4.9], which references element <i>AnalogInput</i> is opened in edit mode in the Composite Editor .
2	Element <i>AnalogInput</i> uses template \$AINPUT_UC, which is going to be replaced.
3	Browser pane that allowed locating template \$AINPUT1_UC, which is going to replace \$AINPUT_UC.

The software displays the list of bindings that cannot be recreated after the template replacement. Write down these bindings so that you can recreate them later on if required.

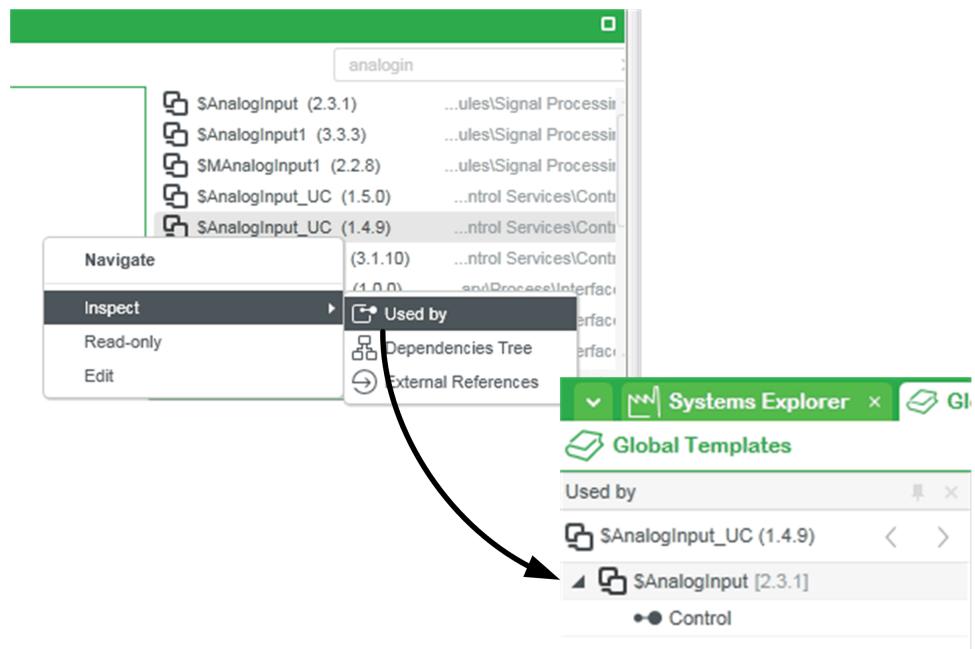


The figure shows the *AnalogInput* element after template replacement is complete and the template referencing it has been saved.

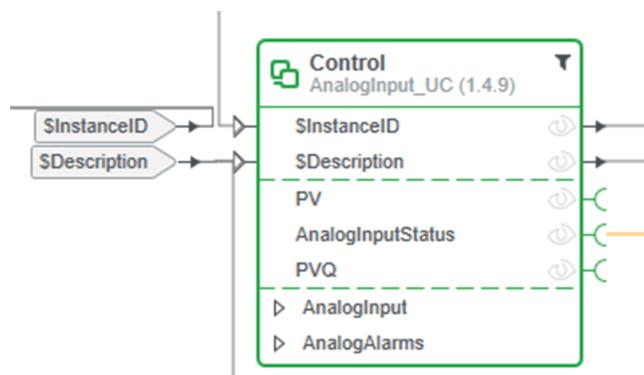


Item	Description
1	Element <i>AnalogInput</i> now uses template \$AINPUT1_UC. Its identifier is unchanged as shown in the element header and the Properties pane.
2	The bindings that were listed in the Replace Conflicts dialog box have not been recreated.
3	\$AnalogInput_UC [1.4.9] has to be saved with a different identifier (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) (for example, <i>MyAnalogInput_UC</i> [1.2.0]) with Usability state approved .

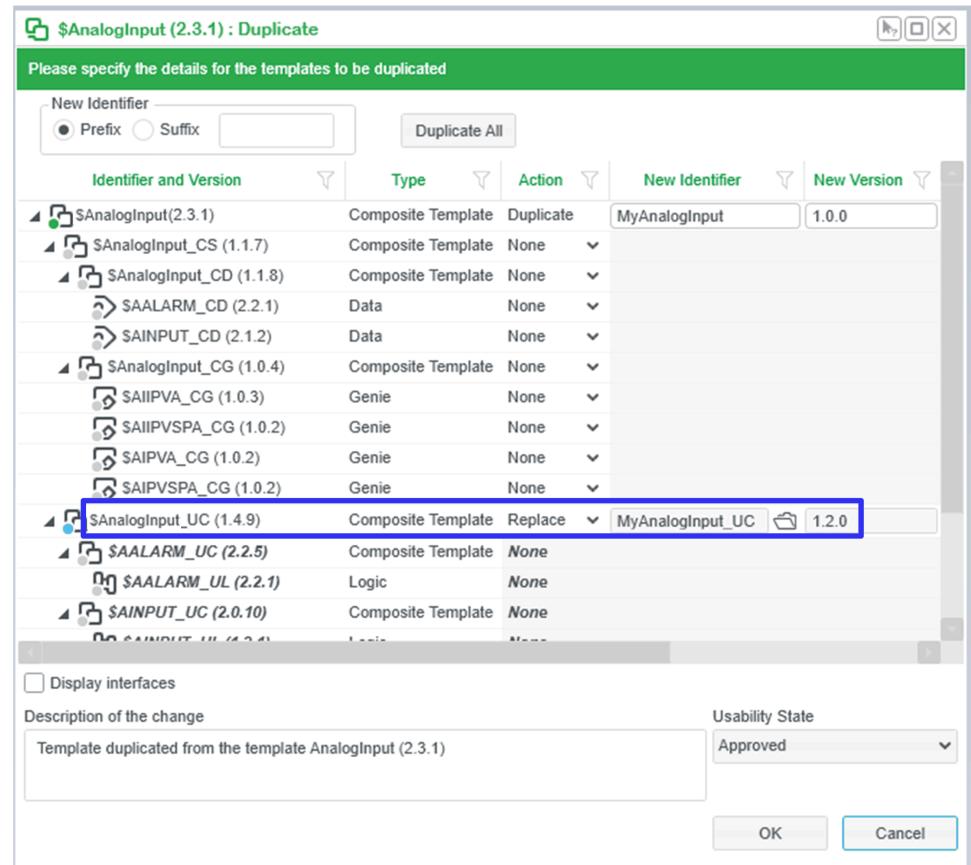
Once template replacement is complete, you need to update templates that reference `$AnalogInput_UC` [1.4.9] so that they use `MyAnalogInput_UC` [1.2.0] instead. To identify these templates, use the **Used by** command on `$AnalogInput_UC` [1.4.9]. In this example, the command returns `$AnalogInput` [2.3.1], which uses `$AnalogInput_UC` [1.4.9] for element *Control*.



The following figure shows in a partial view, element *Control* referenced by `$AnalogInput` [2.3.1].



To update `$AnalogInput [2.3.1]` with `MyAnalogInput_UC [1.2.0]`, navigate to `$AnalogInput [2.3.1]` and use the **Duplicate** command, page 912. This process creates a copy with a different identifier, which includes `MyAnalogInput_UC [1.2.0]`. In this example, duplicating `$AnalogInput [2.3.1]` creates `MyAnalogInput [1.0.0]`. (The template that is replaced is highlighted for the example.)



NOTE: You may need to update application instances that currently use `$AnalogInput [2.3.1]` and/or `$AnalogInput_UC [1.4.9]` to use the new templates `MyAnalogInput [1.0.0]` and/or `MyAnalogInput_UC [1.2.0]` respectively. Use the **Replace** command, page 212 to perform the update.

Duplicating Global Templates

Overview

The **Duplicate** command that is available in the context-menu of templates in the **Global Templates Explorer**, lets you create a copy or a new version of any template with the original and/or modified child templates.

For each child template, you can select to keep the original, duplicate (rename or change the version), or replace it by another existing template.

The software informs you if it detects binding-related conflicts because of a template replacement before proceeding with the duplication of the template.

NOTE: This command can be used instead of the **Update** command for templates whose identifier contains the \$ prefix. However, you need to know for which child templates a new version already exists and select it manually.

Restrictions When Duplicating Schneider Electric Templates

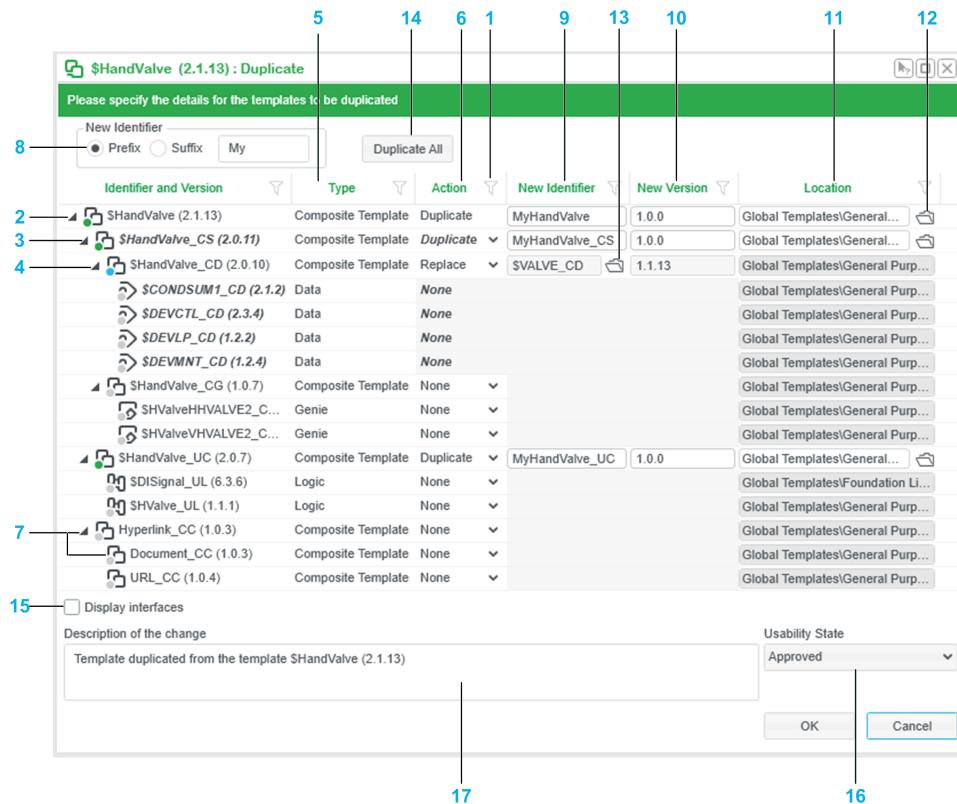
Restrictions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) related to the template identifiers apply when you duplicate a Schneider Electric template .

Good Practices

- Use the **Duplicate** command as a one-step solution to create a copy of a Schneider Electric template whose child templates you can edit and save without using the \$ prefix (see EcoStruxure™ Process Expert, Global Templates, Reference Manual).
- Avoid duplicating templates of the Foundation library and interface models because these are low-level resources that are common to all Schneider Electric templates to make them widely compatible. You can identify foundation templates by their location.

Duplicate Window

The following figure shows an example of the **Duplicate** window for a *\$HandValve* sample template in which duplication and replacement actions for child templates have been selected.



Item	Description
1	You can sort and filter, page 118 data.
2	Identifier and version of the template from which you have selected the Duplicate command.
3	Identifier and version of child templates that the template references. At each level of the template composition, each template is shown only once even if several instances of it exist. By default, interface models are not shown.
4	When you replace a template, the child templates of the replacement are shown under the template to be replaced in read-only mode.
5	Type, page 25 of the template.
6	Action to be performed for each referenced child template. Each type of action is identified by a colored dot: <ul style="list-style-type: none"> Duplicate (green): Creates a new template. If the template references child templates, they are reused without change unless you duplicate all or some of them as well. None (gray): No action is performed. The template is reused in the duplicated parent template. For replacement templates and their children, it is the only setting that is allowed. Replace (blue): Opens the Replace dialog box, page 916, which lets you select an existing template in the Global Templates library to be used in place of the current template and its child templates. The replacement template is used for all instances of the current template. The template must have the usability state (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) Approved.
7	Duplicating or replacing a template automatically sets the action of all parent templates to Duplicate . Entries in italic indicate a change that is the result of an action that you have selected.

Item	Description
8	Free form text fields to enter either a prefix or suffix that is added to the identifier of each duplicated template.
9	Identifier of the template that is proposed by the software for duplicated templates (editable) or identifier of the replacement template that you have selected.
10	Version that is proposed by the software for duplicated templates (editable) or version of the replacement template.
11	<p>Location in the Global Templates library of:</p> <ul style="list-style-type: none"> • An original template (None action). The location is read-only. • The duplicated template. You can select a different location and/or add additional ones. <p>The location that you select must exist. When you select more than one location, the software creates linked copies, page 885 of the template.</p> <p>In the location selection dialog box, a check box lets you set the same location for all duplicated templates.</p> <ul style="list-style-type: none"> • The replacement template. The location is read-only. • Child templates of a replacement template (None action). The location is read-only
12	Button to modify the location of duplicated templates within the Global Templates library.
13	Button to open the Replace dialog box, page 916 to select a different template.
14	<p>Sets the action of all child templates to Duplicate. Applies to interfaces only if Display interfaces was selected when the Duplicate All button was clicked.</p> <p>This action overrides changes that you have made for child templates for which you have selected the Replace action.</p> <p>NOTE: You cannot revert this action for all templates at once unless you cancel and discard changes.</p>
15	<p>When selected, shows the interfaces that are referenced by child templates. You can select an action for each interface. The changes that you have made to interfaces are retained when you clear the check box.</p> <p>When you select the Duplicate command for an interface, select it to show nested interfaces.</p>
16	Lets you select the usability state (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) that is set for all duplicated templates.
17	Free-form text field to enter a description of the changes.

Duplicating Global Templates

Duplicating Schneider Electric Global Templates or user-created templates may affect the function of these templates and must be performed by qualified personnel. Before proceeding, refer to the [Overview](#), page 837.

To duplicate a global template, proceed as follows.

Step	Action
1	<p>In the tree view of the Global Templates Explorer, right-click the template that you want to duplicate and select Duplicate.</p> <p>Result: The Duplicate window opens.</p>
2	Select an action for referenced child templates as needed.
3	Configure the parameters for duplicated templates as needed.
4	Select a usability state for duplicated templates.
5	<p>Click OK.</p> <p>Result: If the software detects no binding-related conflicts, it duplicates the template based on your selections; otherwise, the Replace Conflicts dialog box opens.</p>

Step	Action
6	Verify the information that is displayed in the Replace Conflicts dialog box and click: <ul style="list-style-type: none">• Yes: Proceeds with the duplication of the template and discards conflicting bindings.• No: Closes the Replace Conflicts dialog box and reverts to the Duplicate window.
7	Recreate bindings as needed.

Replace Dialog Box

Overview

The **Replace** dialog box opens when you select the corresponding command in the **Duplicate** window, page 912 or in the context menu of an element in a template editor, page 907.

It lets you browse the Global Templates library to select a replacement template.

Description

The browser of the **Replace** dialog box behaves like the templates browser, page 840 of the **Global Templates Explorer** with the following exceptions:

- The scope of the search is context-sensitive.
- Facet templates are shown by default.
- Only templates with the usability state **Approved** are shown.

Editing and Extending Interfaces In-Place

Overview

The **Edit/Extend Interface** command lets you edit, add, and remove elementary elements of an interface that is referenced in a template without the need to edit the interface by using the **Interface Editor**.

You can add elements in direct mode only.

The operation creates a new version of the interface and updates the reference. It also allows you to update other references of this interface (any role) that exist in this template.

The command is not available for deferred interfaces and when the interface element contains a nested interface.

NOTE: The update of an interface reference impacts the other role of this interface and requires updating, page 899 the templates that reference the previous version of the other role. Refer to the example, which describes the recommended workflow, page 919.

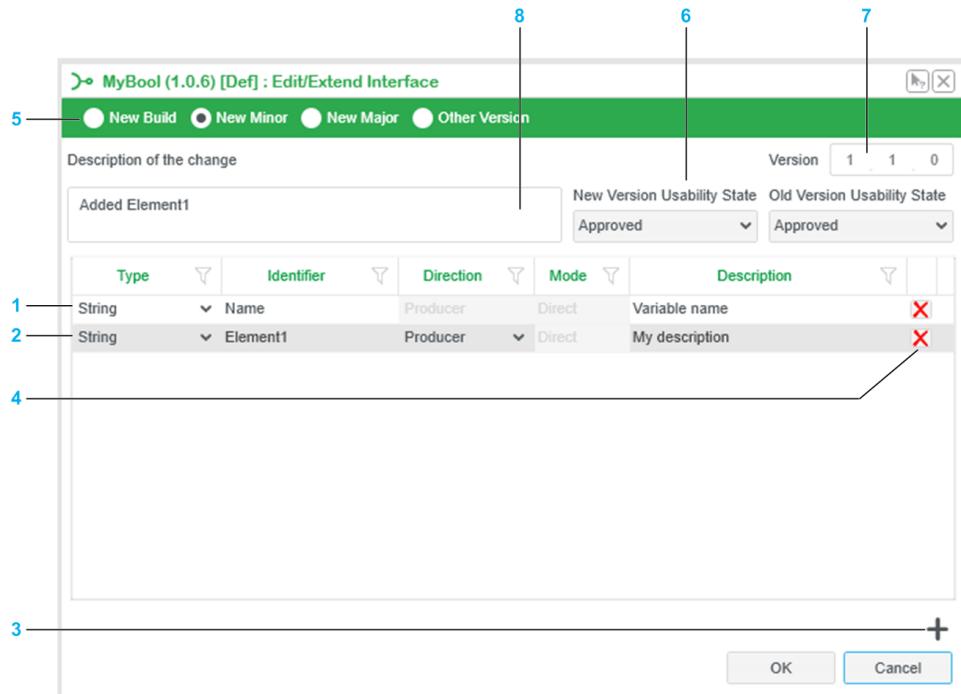
Restrictions When Updating Schneider Electric Templates

Restrictions (see EcoStruxure™ Process Expert, Global Templates, Reference Manual) related to the identifier of the parent template and its elements apply when you update a Schneider Electric template.

You cannot use the **Edit/Extend Interface** command on an interface whose identifier starts with the \$ prefix. You need to edit the interface from within the template and save it with a new identifier or use the **Duplicate** command, page 912.

Edit/Extend Interface Window

The following figure shows an example of the **Edit/Extend Interface** window.



Item	Description
1	<p>Element that was originally part of the interface.</p> <p>Once an element has been added and the interface saved, you cannot change its Direction and Mode properties anymore.</p> <p>NOTE: Modifying Type does not discard existing bindings. However, modifying Identifier does.</p>
2	<p>Element that is being added to the interface by using the Edit/Extend Interface command.</p> <p>The Mode interface property (see EcoStruxure™ Process Expert, Global Templates Reference Manual) is read-only; only direct data propagation is supported when you use the command.</p>
3	Button to add a new element.
4	<p>Button to remove the element.</p> <p>NOTE: If you remove a bound element and recreate it with identical identifier and configuration, the binding with this element is maintained when you update the interface.</p>
5	<p>Lets you select the versioning scheme that you want to use for the new version of the interface.</p> <p>Select Other as version scheme to edit the value.</p>
6	<p>Lets you select the following:</p> <ul style="list-style-type: none"> • Usability state that the new version of the interface will have. <p>NOTE: Only <i>Approved</i> or <i>Deprecated</i> allow using the new version of the interface right away in the template.</p> <ul style="list-style-type: none"> • Usability state of the interface (old) that you are editing/extending after you create the new version of interface.
7	Version of the interface that you are creating.
8	<p>Mandatory change description.</p> <p>The description is visible in the changes log of the interface.</p>

Editing/Extending Interfaces

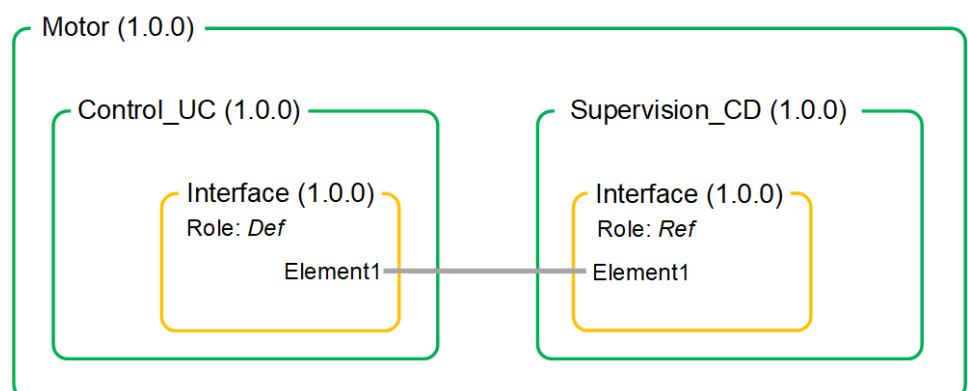
To edit/extend an interface from within its parent template, proceed as follows.

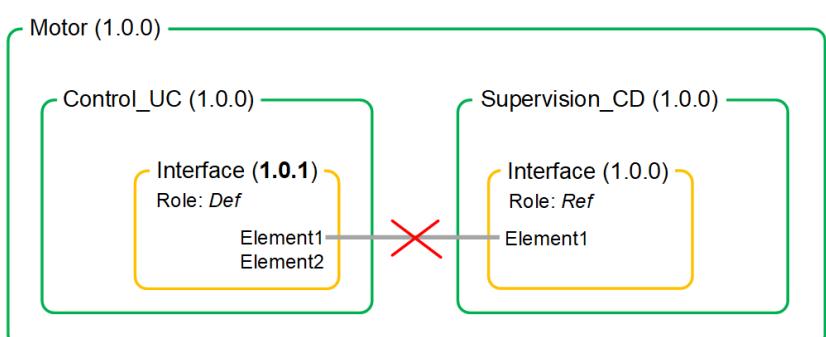
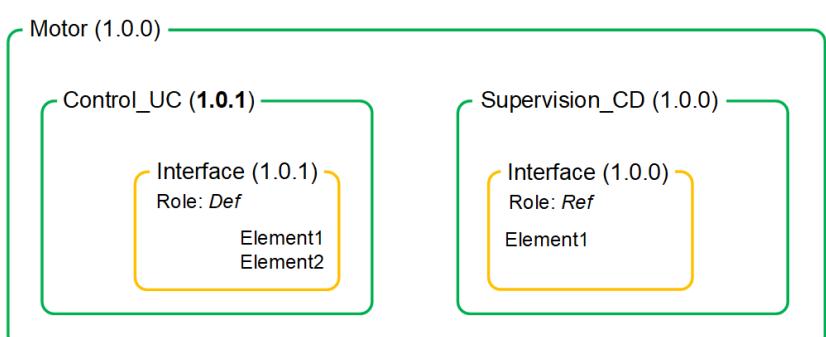
Step	Action
1	in the Facet Editor or Composite Editor , right-click the header of an interface element and select Edit/Extend Interface .
2	In the Edit/Extend Interface window, perform the required changes, enter a change description, and click OK . Result: The window closes and a new version of the interface with your configuration is created. NOTE: If New Version Usability State is <i>Not Approved</i> or <i>Obsolete</i> , a new version of the interface is created but the interface role element is not replaced in the template.
3	If other references (any role) of this interface exist in this template, the Update References dialog box, page 899 opens. Click either button: <ul style="list-style-type: none"> • Yes: Updates all references of the interface in the template if existing bindings can be recreated. • No: Only updates the interface that you have edited if existing bindings can be recreated.
4	If in the new interface version at least one element that was bound has been removed, a Replace Conflicts dialog box opens in sequence for each reference that is updated. In each dialog box, verify the information that is displayed about bindings that cannot be recreated and click either button: <ul style="list-style-type: none"> • Yes: Proceeds with the replacement of the reference without recreating bindings with elements that are removed. • No: Does not replace the reference.
5	Save changes in the template that you are editing.
6	Update the other templates that reference other role of each interface reference. You can identify these templates by using the Inspect > Used By context menu command on the previous version of the interface in the Global Templates Explorer or using the Update command at the folder level.

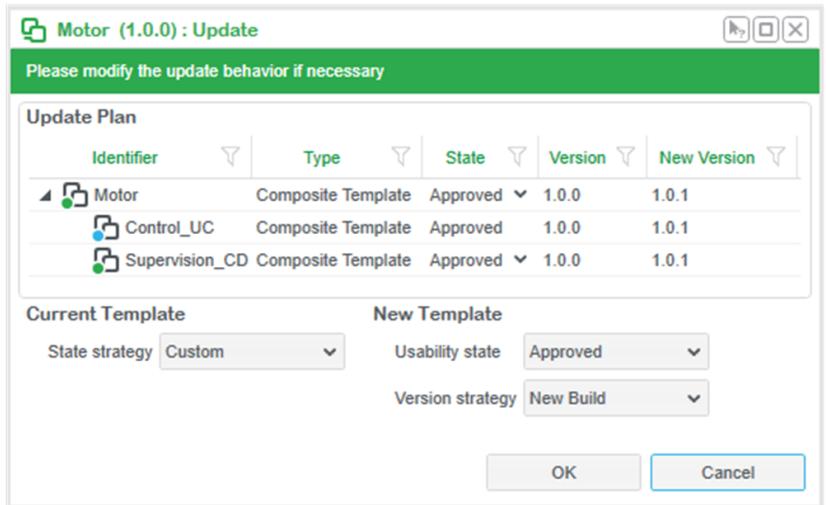
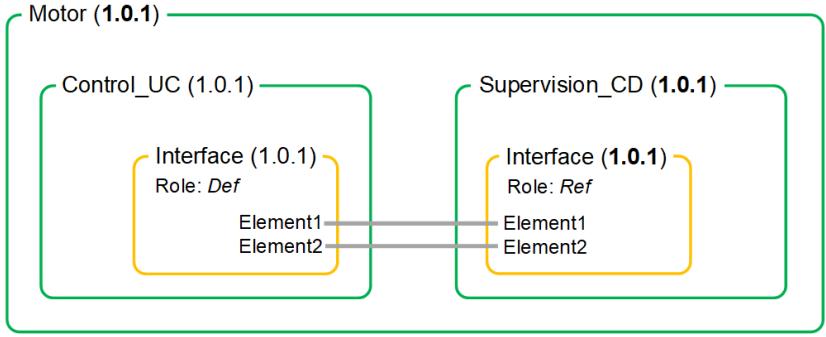
Recommended Workflow When Editing/Extending an Interface in a Control Module Template

This following table describes the workflow to propagate changes that were made to an interface by using the **Edit/Extend Interface** command to the entire composition of the template referencing this interface. A simple template is used as example.

The control module template *Motor (1.0.0)* contains two child templates. Each of them references a role of *Interface (1.0.0)*. The interface consists of one element to propagate data.



Step	Action
1	<p>From the Global Templates Explorer, edit the template that contains either role of the interface.</p> <p>(In this example, <i>Control_UC</i>. However, you can also edit <i>Supervision_CD</i> instead. The overall workflow remains the same.)</p>
2	<p>Select the Edit/Extend Interface command on the interface and modify it.</p> <p>(In this example, <i>Interface</i> and adding <i>Element2</i>.)</p>
3	<p>Save changes to this role of the interface with a new version and usability state <i>Approved</i>.</p> <p>(In this example, <i>Interface (1.0.1)</i>, role <i>Def</i>.)</p> <p>NOTE: Do not rename the interface because this will prevent the propagation of the changes throughout the template. If the identifier of the interface starts with the \$ prefix, use the Duplicate command, page 912 to remove it first.</p>  <p>NOTE: At this point, data is not propagated anymore to other templates referencing the original version of other role of the interface (In this example, to <i>Supervision_CD (1.0.0)</i> referencing role <i>Ref</i> but of <i>Interface (1.0.0)</i>).</p>
4	<p>Save the template (Save As) that contains the modified interface role with a new version, usability state <i>Approved</i>, and close it.</p> <p>(In this example, <i>Control_UC (1.0.1)</i>.)</p> 

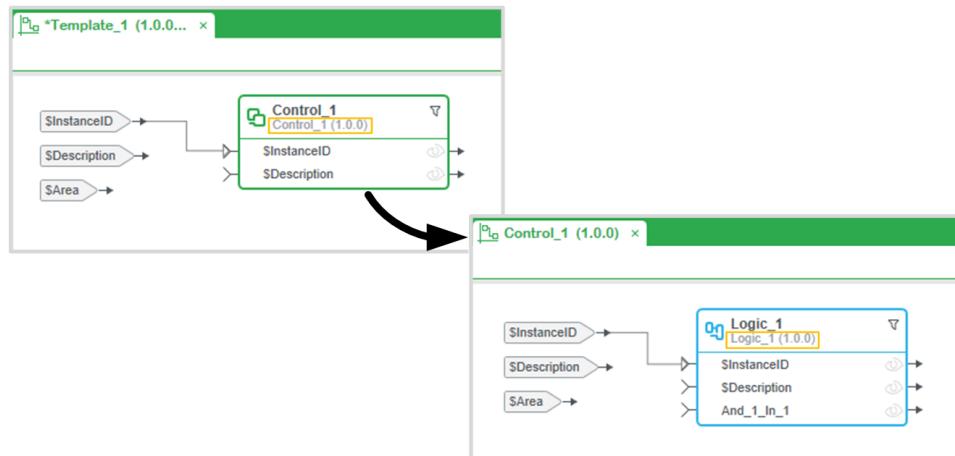
Step	Action
5	<p>In the Global Templates Explorer, right-click the highest level template that contains the interface and its parent and select Update.</p> <p>(In this example, <i>Motor (1.0.0)</i>, which contains <i>Interface</i> and its parent <i>Control_UC (1.0.0)</i>. However, in your template, it could be several levels up.)</p> <p>Result: The Update window opens and lists, among other templates, all the child templates that contain a reference of either role of the original interface version, which will be updated to the new version containing your changes.</p>  <p>NOTE: The interface that you have modified does not appear.</p>
6	<p>In this example, clicking OK creates <i>Motor (1.0.1)</i>, which is updated with the following:</p> <ul style="list-style-type: none"> Existing <i>Control_UC (1.0.1)</i> (which already contains <i>Interface (1.0.1)</i>, <i>Def</i>) <i>Supervision_CD (1.0.1)</i> created by the software (in which <i>Interface (1.0.0)</i>, <i>Ref</i> is updated to <i>Interface (1.0.1)</i>, <i>Ref</i>) <p>As a result, the binding between <i>Element1</i> is automatically re-established and a new binding between <i>Element2</i> created.</p> 

Updating References in Parent After Editing Child Element – Example

This topic describes the update of references inside the parent template by using a simple three-level template as an example. It shows how to update the parent after the lowest-level template has been modified.

Starting Point – Parent Template

To illustrate the automatic update process, page 899, this example uses parent *Template_1* version 1.0.0, which references composite *Control_1* version 1.0.0, which references facet *Logic_1* version 1.0.0.



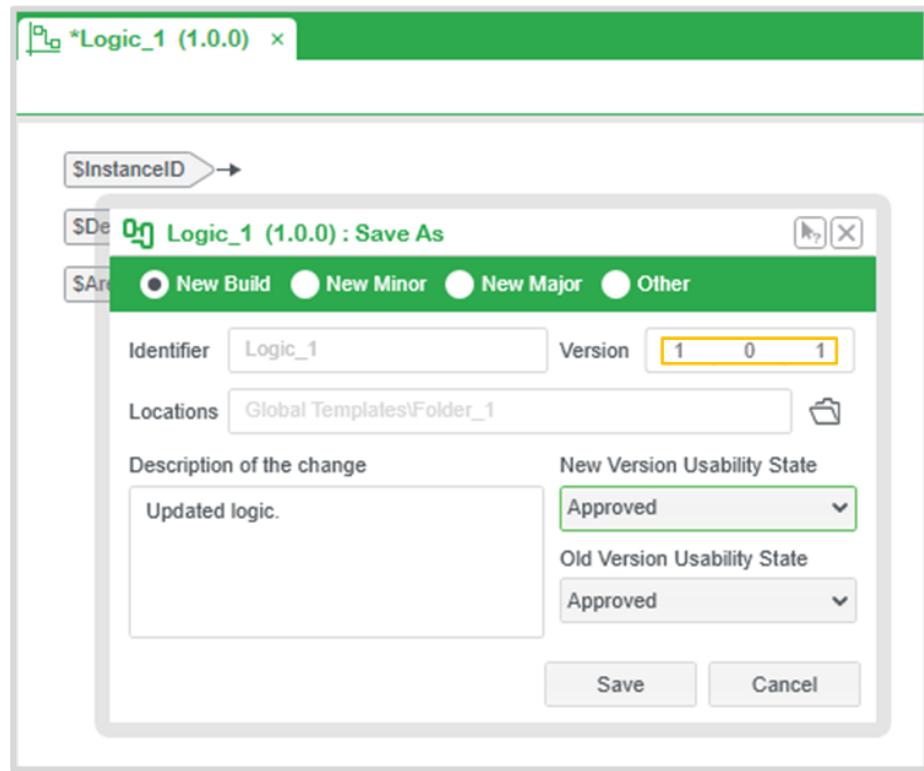
NOTE: Template identifiers and versions are highlighted with an orange outline for illustration purposes only.

Modifying the Lowest-Level Template and Saving Changes

Template_1 is edited.

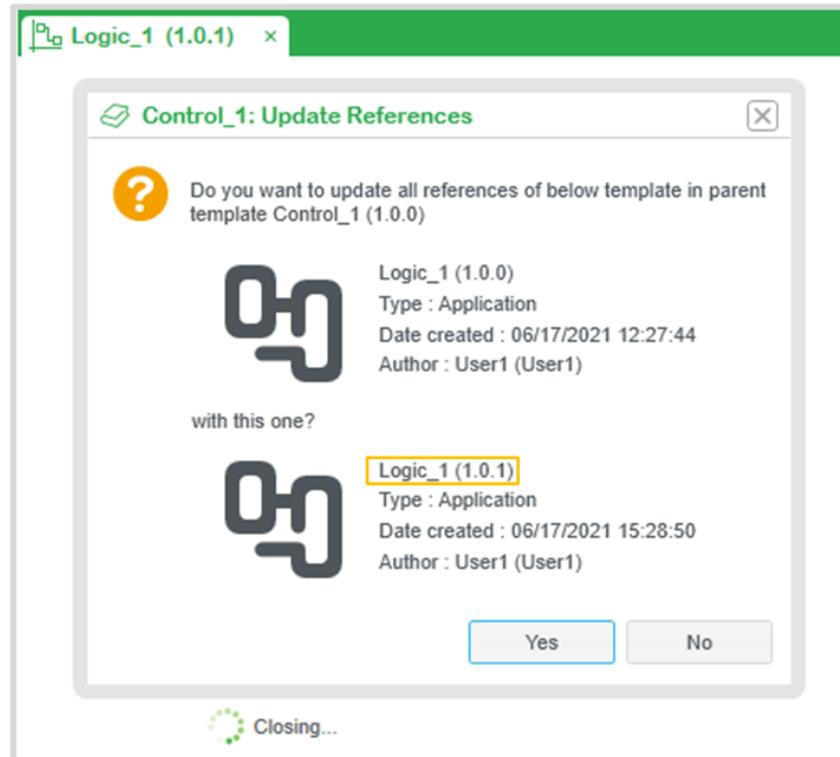
From within *Template_1*, *Control_1* is edited (by using the context menu command). All templates must remain open.

Then, from within *Control_1*, *Logic_1* is edited, modified, and saved with a new version (1.0.1) and usability state *Approved*.



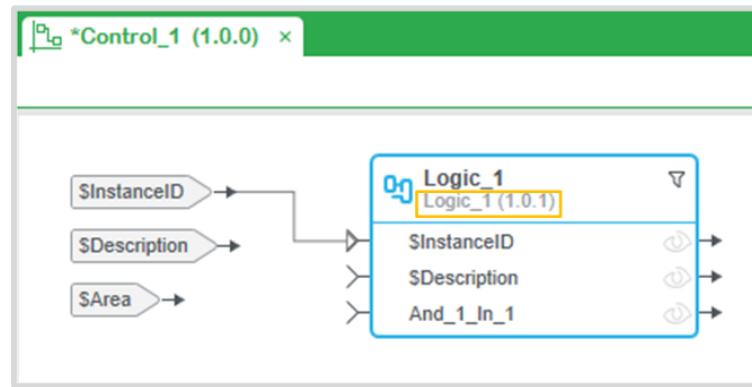
Closing the Lowest-Level Template and Updating the Reference in Parent (Mid-Level Template)

After closing *Logic_1*, the **Update References** dialog box opens, which lets you update reference *Logic_1* v1.0.0 with version 1.0.1 inside *Control_1*.



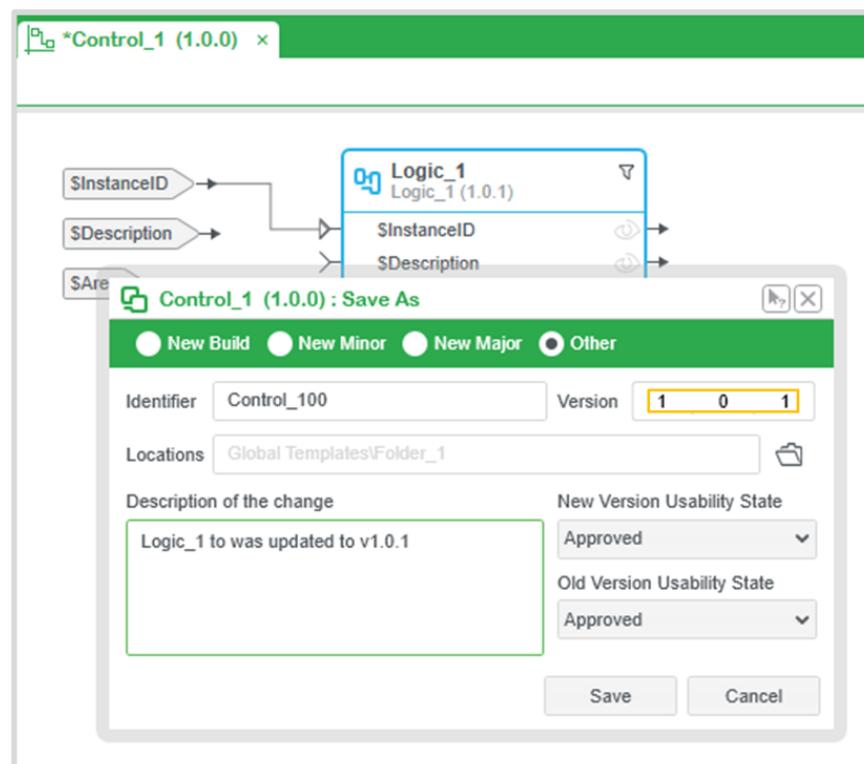
NOTE: If *Control_1* contains several references of *Logic_1* v1.0.0, clicking **Yes** replaces them all. Bindings are managed by the software and a **Replace Conflicts** dialog box opens for each reference for which at least one binding cannot be recreated. You can select not to proceed with the update.

The following figure shows the result in *Control_1* after clicking **Yes** in the dialog box.



Saving Changes in Mid-Level Template

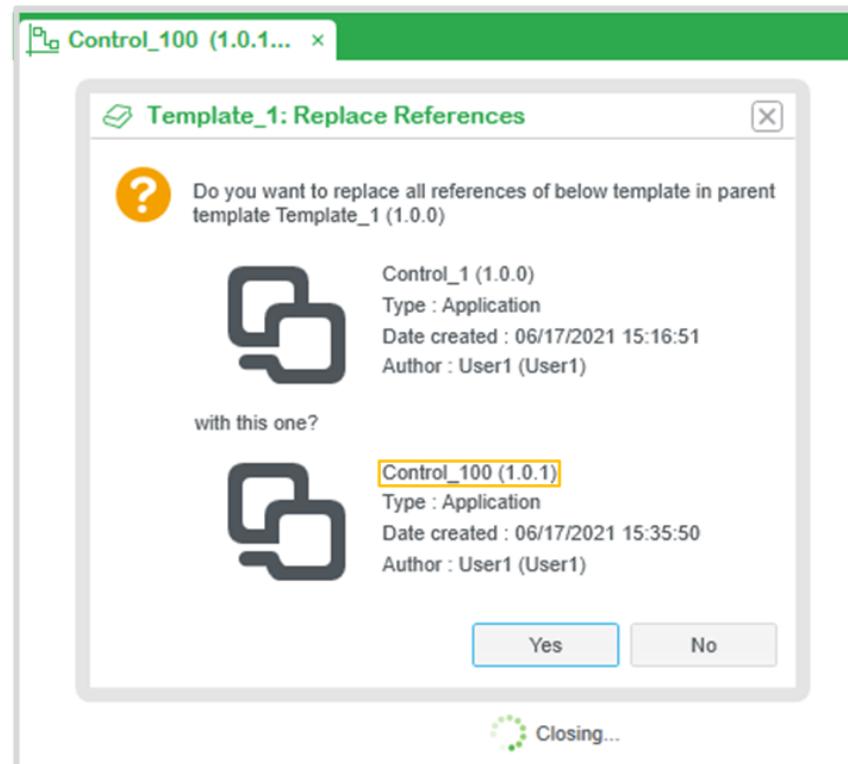
To apply the update of *Logic_1*, *Control_1* needs to be saved with a new version (1.0.1) and usability state *Approved*. In this example, *Control_1* is renamed *Control_100* to show the impact of renaming a template when saving it.



NOTE: You can perform additional modifications in *Control_1* before saving it.

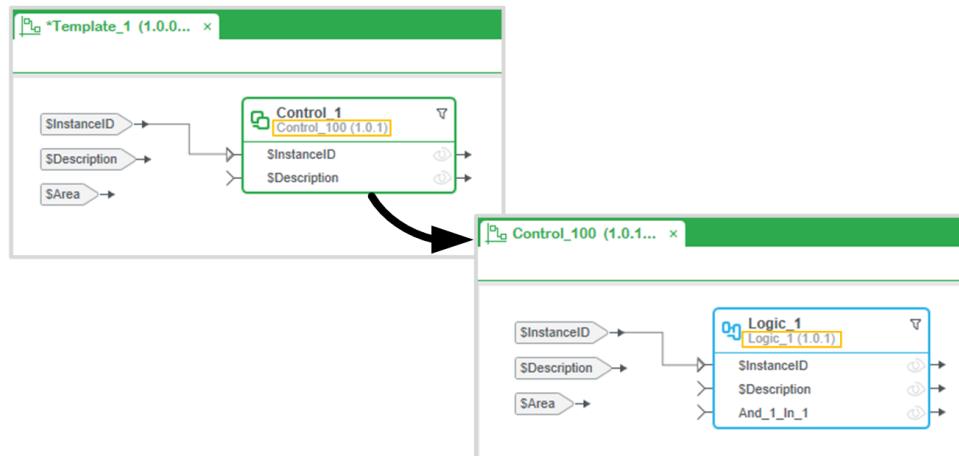
Closing the Mid-Level Template and Updating the Reference in Parent

After closing *Control_100*, the **Replace References** dialog box opens, which lets you replace reference *Control_1* v1.0.0 with *Control_100* version 1.0.1 inside *Template_1*.



Result of the Update Process

The following figure shows the result in *Template_1* after clicking **Yes** in the dialog box. The modifications in *Logic_1* and *Control_100* have been propagated to the highest level. To finalize the update process, *Template_1* needs to be saved.



Software Services

What's in This Part

System Documentation.....	927
Content Repository.....	935

Overview

This part describes the services that are provided by the software to facilitate the engineering of systems.

System Documentation

What's in This Chapter

About System Documentation	927
System Documentation Report Content	928
Generating System Documentation Reports	934

About System Documentation

Scope of System Documentation Reports

You can generate, page 934 a system documentation report that contains information on either one:

- A system (all domains)
- One or more domains of a system. The domains are:
 - **Application**
 - **Projects** (Control and Supervision)
 - **Topology**
 - **Global Templates**

The report encompasses the detailed engineering data of a system or its domains including:

- Instance custom configuration.
- Application hierarchy.
- Control and Supervision project content (includes Participant-generated documentation), mappings, properties, and settings.
- Topological entity configuration and physical connections.
- All templates of the Global Templates library or only templates that are used by objects of the system.

Details on how to configure the report to obtain the desired information are given in the following pages.

Report Layout

You can customize the information that appears in the report as well as the report layout in the first **Generate Report** window. You can also preview the report cover page.

The following are always included in a system documentation report.

Information	Comment
Product name	Name of the application that is used to engineer the system. Appears on each page of the report. For example, <i>EcoStruxure Process Expert Report</i> .
Identifier of the system	For example, <i>System_1</i> . Appears on the first page of the report.
Schneider Electric logo	Appears in the header and footer of each page.
Creation date and time	Current computer date and time.
Blank pages	Five blank pages are added at the end of the report for your own use.

The following information and report properties can be customized.

Information	
Header and footer	Customizable. Free-form text. Appear on each page of the report.
Page orientation and size	Applies to all pages of the report.
Report format	PDF or Microsoft Word.
Report file name and location	Default values: <ul style="list-style-type: none">• Name: <System identifier>• Location: The Windows Documents folder of the user logged in to the computer. NOTE: Generate each report, page 934 in a separate folder because the software generates subfolders with the same name.
Enduser details	<ul style="list-style-type: none">• Customer name and site name: Free-form text. Appear on the cover page only.• Company logo: Appears in the header of each page. Accepted formats: JPG, PNG, BMP. The size of images is automatically adjusted to fit the report (approximately 350 x 175 pixels).
Report description and author	Free-form text. Appear on the cover page only.

System Documentation Report Content

System Information

System properties, page 137 are not included in a report.

Domain Information

For each domain of a system that you select, a corresponding section appears in the report. The following pages describe the content that is included in each section by default and on-demand.

Application Domain Report Data

Object Selection

The **Application** tab reproduces the hierarchy of the application and the instances that exist in the system. You can exclude objects from the report by clearing their checkbox.

NOTE: Clearing the checkbox of the root node excludes the entire domain from the report.

Default Information

The *Application Domain* section of the report contains the following:

- *Folders* table
- *Instances* table

NOTE: Element selections and default parameter values are not shown.

Optional Information

The following table describes the information that is added to the *Application Domain* section of the report when you select the corresponding option.

Optional selection	Description
Include Modified Parameters	Adds the <i>Modified Parameters</i> table. For each instance, lists the parameters whose value is different from the default value of the template. <i>Parameters</i> column: <ul style="list-style-type: none"> • <i>Name</i>: Name of the modified parameter. • <i>Value</i>: Modified value.
Include Interface Links	Adds the <i>Interface Links</i> table. For each instance that is the source of an interface link, it lists the following: <ul style="list-style-type: none"> • Source interface • Destination instance • Destination interface
Include Workspaces	Adds the <i>Workspaces</i> table, which lists the workspaces that exist in the application. For each workspace, it lists the following: <ul style="list-style-type: none"> • The instances it contains and the template the instances use. • Information about their interfaces

Project Domain Report Data

Object Selection

The **Projects** tab with its **Control Project** and **Supervision Project** child tabs reproduces the projects that exist in the system. You can exclude projects from the report by clearing their checkbox.

NOTE: Clearing the checkbox of the root node in the Control or Supervision child tab excludes the respective sub-domain from the report.

Default Control Project Information

The *Project Domain* section of the report consists of a sub-section for each Control project, which contains the following:

- *Properties* table: Indicates the controller platform of the project.
- *Settings* table: Indicates the settings that are configured for the project.
- *Control Executable* sub-section: A subsection exists for each Control executable of a Control project. It contains the following information:
 - *Service Mappings* and *Communication Mappings* tables: A table for each type of mapping that exists.
 - *Hardware Mappings* table: A table lists the hardware mapping interfaces of the mapped controller and, if they have been mapped, the interfaces of instances whose facets are assigned to the project.

Optional Control Project Information

The following table describes the information that is added to each Control project sub-section of the report when you select the corresponding option.

Optional selection	Description
Include Assignments	<p>Adds the <i>Assignments</i> table to each Control project sub-section.</p> <p>For each FBD project section, lists the instances and their facets that are assigned to it. Also shows the facet validity status and the assignment and generation statuses.</p>
Include Section Content and Controller Configuration	<p>Adds documentation that is generated by the Control Participant in a technical documentation folder.</p> <p>Documentation is generated only if a Control project contains at least one section and has been generated.</p> <ul style="list-style-type: none"> • Documentation file names: <<i>Control project identifier (time stamp)</i>>.pdf (for example, ControlProject_1 (20221025143510).pdf) • Technical documentation folder name: <i>UnityReports</i>. • Location of technical documentation folder: Same as the system documentation report. <p>For details on the content of the Control Participant documentation, refer to the <i>Topics in Documentation</i> topic in the help of the Control Participant (<i>Operating Modes</i>).</p> <p>NOTE:</p> <p>The information about the configuration does not correspond to the configuration (topology) of the controller that is mapped to executable of the Control project but to the default configuration that exists for each Control Participant project.</p> <p>To view information about the configuration part of the Control project, refer to the <i>Engine</i> column of the <i>Service Mappings</i> table of its executables and to the <i>Topology Domain</i> section of the system documentation report.</p>

Default Supervision Project Information

The *Project Domain* section of the report consists of a sub-section for each Supervision project, which contains the following:

- *Cluster* sub-section: A subsection exists for each cluster of a Supervision project. It contains the following information:
 - *Tag Container Properties* table: Lists the tag containers that exist.
 - *Server* tables: A table for each type of server that exists (such as, alarm, report, I/O).
 - *Boards and Ports* tables: A table for each, which lists the boards and ports that exists.
 - *IODevices* table: Lists the I/O devices that exist.
- *Supervision Executable* sub-section: A subsection exists for each Supervision executable of a Supervision project. It contains the following information:
 - *Service Mapping* table: Lists the service mappings that exist.
 - *Communication Mapping* table: Lists the communication mappings that exist.

Optional Supervision Project Information

The following table describes the information that is added to each Supervision project sub-section of the report when you select the corresponding option.

Optional selection	Description
Include Assignments	<p>Adds the <i>Assignments</i> table to each Supervision project sub-section.</p> <p>For each tag container, lists the instances and their facets that are assigned to it. Also, shows the facet validity status and the assignment and generation statuses.</p> <p>If the Supervision project contains more than one cluster, a table exists for each cluster.</p>
Include Genie Assignments	<p>Adds the <i>Genies</i> table.</p> <p>For each page, lists the instances and their facet that are assigned to it. Also, shows the facet validity status and the assignment and generation statuses.</p>
Include Pages	<p>Adds a graphical output of each page as a separate file in a technical documentation folder.</p> <ul style="list-style-type: none"> • Page file names: <Supervision project identifier>_<Page identifier>.pdf (for example, Supervision_1_Page_1.pdf) • Technical documentation folder name: <i>CitectPageReports</i>. • Location of technical documentation folder: Same as the system documentation report.
Include Tag Data	<p>Adds the content of the various DBF files of the Supervision Participant project as a separate files in CSV format in a technical documentation folder.</p> <ul style="list-style-type: none"> • File names: <Supervision project identifier>_<Table>.csv (for example, Supervision_1_variable.csv) • Technical documentation folder name: <i>CitectTagReports</i>. • Location of technical documentation folder: Same as the system documentation report.

Topology Domain Report Data

Object Selection

The **Topology** tab reproduces the hierarchy of the topology and the entities that exist. You can exclude objects from the report by clearing their checkbox.

NOTE: Clearing the checkbox of the root node excludes the entire domain from the report.

Default Information

The *Topology Domain* section of the report contains a pair of tables for each type of topological entity that exists and has been configured (for example, controllers, station nodes, I/O devices):

- <*Topological entity type*> table: For each type of topological entity, lists the entities that exist and the devices/services they consist of.
- <*Topological entity type*> *Properties* table: Lists some of the properties of each topological entity. For station nodes, the *Station Node Services* table is shown instead.

Control Participant Documentation for Controllers, STB Islands, and PRMDevices

For each controller, STB island, and PRM device that is configured, additional documentation generated by the Control Participant (project documentation) is included in a technical documentation folder with a generic name (for example, *ControllerReports*, *AdvantysReports*).

Documentation is generated only if the topological entity is included in the system documentation report.

- Documentation file name: <*Topological entity identifier (time stamp)*>.pdf (for example, *Controller_1 (20221025143510).pdf*)
- Location of technical documentation folders: Same as the system documentation report.

For details on the content of the Control Participant documentation, refer to the *Topics in Documentation* topic in the help of the Control Participant (*Operating Modes*).

Optional Information

The following table describes the information that is added to the *Topology Domain* section of the report when you select the corresponding option.

When a entity that corresponds to an option (for example, a controller, a station node) exists in the topology, the option is automatically selected. Conversely, clearing an option excludes entities of the corresponding type from the report.

Optional selection	Description
Include Controller Information	Adds the <i>Controllers</i> and <i>Controller Properties</i> tables as well the Control Participant documentation.
Include Station Node Information	Adds the <i>Station Nodes</i> and <i>Station Node Services</i> tables.
Include Ethernet Network Information	Adds the <i>Ethernet Networks</i> , <i>Ethernet Network Properties</i> , and <i>Physical Connections</i> tables.
Include STB Island Information	Adds the <i>STB Islands</i> and <i>STB Island Properties</i> tables as well as the Control Participant documentation.
Include Device Information	Adds the <i>Device/Os</i> and <i>Device/O Properties</i> tables.
Include PRM Profibus DP Information	Adds the <i>PRM Devices</i> and <i>PRM Device Properties</i> tables as well as the Control Participant documentation.

NOTE: When you select an option, a table is added only if a corresponding entity exists in the topology.

Global Templates Domain Report Data

Object Selection

The **Global Templates** tab reproduces the hierarchy of the entire Global Templates library and the templates it contains. You can exclude templates from the report by clearing their checkbox.

NOTE: Clearing the checkbox of the root node excludes the entire domain from the report.

Default Information

The *Global Templates Domain* section of the report contains the following:

- *Global Templates* table: Lists the templates of the Global Templates library grouped by library and location/type. It also indicates the template version, usability state, description, and whether it is used in the system or not.

Optional Information

The following table describes the information that is added to the *Global Templates Domain* section of the report when you select the corresponding option.

Optional selection	Description
Include Change History	Adds the <i>Change History</i> table. For each template, lists the information that appears in the changes log. NOTE: The <i>Change History</i> table contains only templates that are used in the system if the Include Only Used Templates option is selected.
Include Only Used Templates	Limits the content of the <i>Global Templates</i> table to templates that are used in the system. Otherwise, includes the data of the templates that exist in the Global Templates library. The option is selected by default. Clearing the option may generate a very large report and take time.

Generating System Documentation Reports

Procedure

To generate a report containing information of a complete system or some of its domains, proceed as follows.

Step	Actions
1	In the Systems Explorer , select a system and verify that no other window is open in the engineering client.
2	Click Generate Report in the toolbar or context menu of the system. Result: The first Generate Report window opens.
3	In the first window, fill out the information as required, select a format, name, and location for the report, and click Next . NOTE: Create each report in a separate folder; otherwise, technical documentation folders, page 932 and their content are overwritten without notification.
4	In the second window, select the scope of the report by selecting one or more domains and click Next . NOTE: At any time while the Generate Report window is open, you can go back to a previous window and modify your selection. Your selection in the current window or tab is retained.
5	In the third window, wait until the objects that exist in the various domains appear. Then, for each of the selected domains (tabs), refine the scope of the report and click Generate . Result: The Generate Report window closes and the report is generated. A dialog box informs you of the completion of the process and the report location that you have specified opens. NOTE: Do not interact with the Save Print Output As dialog box and Participant windows, which may open several times during report generation if the report contains Participant information. The dialog box and windows close automatically. NOTE: Once the Generate Report window closes, your selections are reset.

Content Repository

What's in This Chapter

Content Repository Explorer	936
Temporary Contents Repository Files	940

Content Repository Explorer

Overview

The content repository is the place where the software stores:

- Global files used by Global Templates and that are required by the software platform to provide various services (for example, global constituents referenced by templates). These files are associated to the library that is installed during the installation of the software and to optional application libraries.
- System-related files (for example, logical Participant projects or user files attached to a Supervision project or to instances). These files are created during the different stages of the system engineering life cycle.

You cannot add to or remove files directly from the content repository. The software removes from the content repository system-related files such as Participant project files when they are not required anymore (for example, when you delete a Participant project). An exception exists for documents that you can manage yourself (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).

Each system server possesses one content repository in which the global and system-related files are stored.

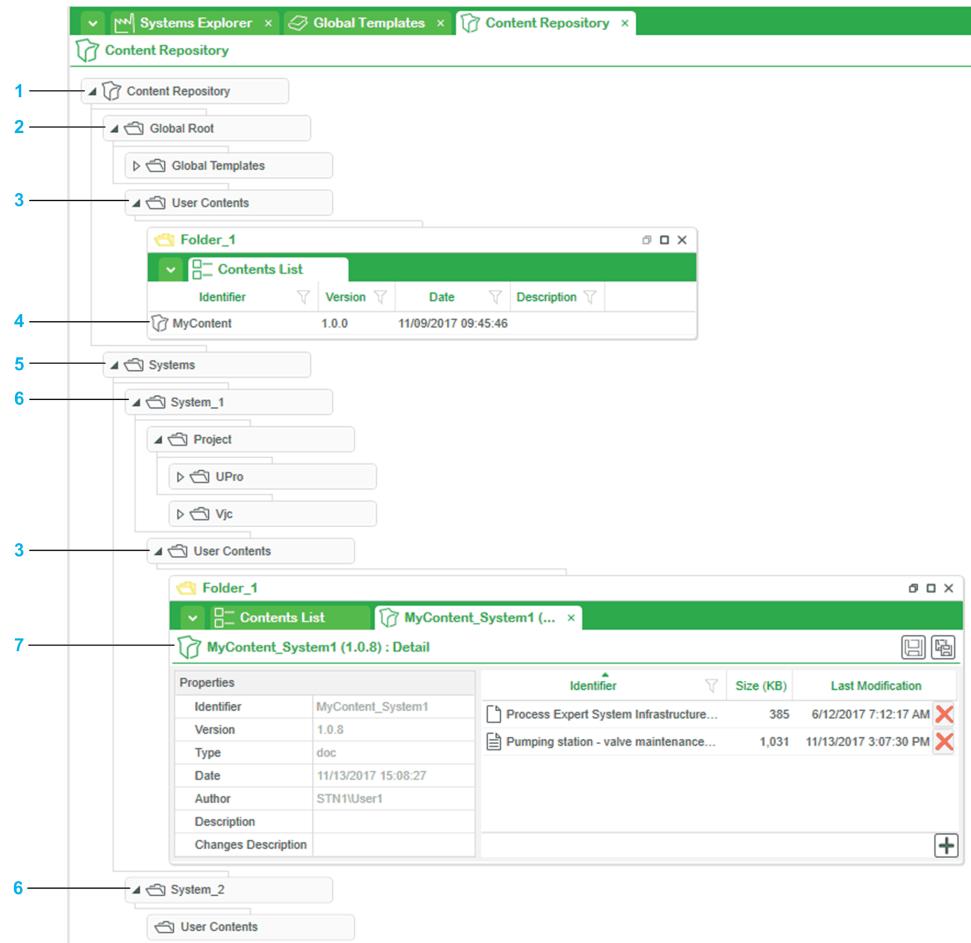
The **Content Repository** explorer is the tool, which lets you view contents in the form of a folder structure and work with the files that are stored in the content repository.

Opening the Content Repository Explorer

To open the **Content Repository** explorer click the **Content Repository** icon in the **toolbar**, page 101.

Content Repository Explorer Window

The following figure shows an example of the **Content Repository** explorer window.



Item	Description
1	Content repository root folder.
2	Root folder for global content repository items that are available at the platform level (system independent).
3	User Contents folders that are the parent folder of the folder structure and content containers that you can add depending on your role.
4	User-created content container.
5	Root folder for content repository items that are available to systems.
6	Root folders for content repository items that are available only to one given system.
7	User-created content container open in the content editor.

Description of the Contents of the Repository

The following data is stored in the content repository under the **Global Root \Global Templates** parent folder.

Node	Contains
Global Constituents	Global Control and Supervision Participant constituents (such as variables, DFBs, included projects).
Local Constituents	Control facet templates of the Global Templates library and for each facet, the referenced local constituent sources and the referenced global constituents
User Contents	<p>Documents (files) that can be managed by users having the appropriate role (see EcoStruxure™ Process Expert, Installation and Configuration Guide). They are delivered with the software and organized in a predefined folder structure and content containers (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).</p> <p>These documents can be opened when viewing instance information by using runtime navigation services.</p> <p>They can be associated either to templates or instances.</p>

The following data is stored in the content repository under the **Systems** parent folder.

Node	Contains
<system identifier>	<p>For each system that exists in the database of the system server, the complete set of files defining the various existing Participant projects, their components, and topological entities.</p> <p>Data is grouped by Participant.</p>
Project	<p>The logical Participant projects under the folder with the name of the project. If you update the logical Participant project by using the Update Project command, the project file is updated, reflecting the changes.</p> <p>For each Participant project, data of the various components of the logical Participant project such as:</p> <ul style="list-style-type: none"> • Section export files under the folder with the name of the section. • The built Participant projects under the folder with the name of the executable.
Topology	<p>Configuration files and profiles of the entities that exist in the topology of the system (under the folder with the name of the entity).</p> <p>For entities that you configure by using the Control Participant, the corresponding project files.</p>
Upro	<p>For each entity of the system, the deployed Participant project. If you do online refinement, the file is updated, reflecting the changes.</p>
VJC	<p>For each Supervision Participant project of the system, the various project files such as database and configuration files.</p>
User Contents	<p>User documents (files) that can be added and managed by users having the appropriate role (see EcoStruxure™ Process Expert, Installation and Configuration Guide). They are organized in a user-created folder structure and content containers (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide) at the system level.</p> <p>These documents can be opened when viewing instance information by using runtime navigation services.</p> <p>They can be associated either to templates or instances.</p>

Content Repository Root Folder Actions

Right-click the **Content Repository** root folder to open a context menu with the following commands.

Command	Description
Export User Contents	Opens the Export User Contents window, which lets you export content containers (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide) with the documents they contain as a single file (.cbk).
Import User Contents	Opens the import window, which lets you select an export file (.cbk) to import (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide) the folders and content containers it contains.

Content Repository File Actions

Starting out from one of the previously mentioned parent folders:

1. Drill down to the lowest level folder of a node until you reach the **Contents List** tab (this requires opening the folder).
2. Right-click a content and select **View Content Detail**.

In the **Detail** tab, right-click a file to open a context menu with the following commands.

Command	Description
View	Opens the file by using the program that is associated to it. The program must be installed on the computer. NOTE: You cannot save changes in the content repository when you open a file by using this command.
Save As	Opens a dialog box, which lets you save a copy of the file by selecting a location and a file name.

NOTE: To be able to open a Control Participant project file (.stu) in a standalone installation of Control Expert, the version of Control Expert and of the installed EcoStruxure Control Expert DTM Library need to be the same as the Control Participant. For information on the installed version, refer to the platform release notes.

NOTE: For Control Participant project files (.stu), an application password is set and file encryption is enabled if a system access password, page 138 for the corresponding system or a Control constituent application password, page 849 is set.

Temporary Contents Repository Files

Overview

The system server stores temporary files from the content repository inside a folder structure to access the files it needs faster. If needed the software creates the temporary storage folders when you start the system server.

Temporary File Folder Location

The folder is named Process Expert x, where x corresponds to the major version of the software that is installed (the R• version suffix does not appear). It is located on the computer on which the system server is installed, at the path C:\Users\<UserName>\AppData\Local\Temp\Schneider Electric.

Reclaiming Disk Space

With time, the size of this temporary file folder increases.

You can reclaim disk space by deleting the subfolders and files inside the Process Expert x folder when the software is not running.

Appendices

What's in This Part

Linking Instances by Using the Links Editor	942
---	-----

Linking Instances by Using the Links Editor

What's in This Chapter

Links Editor.....	943
Linking Instances by Using the Links Editor	946

Links Editor

Overview

The **Links Editor** lets you link the exposed application interfaces of an instance to the interfaces of other instances that are compatible and respect the cardinality of the roles (for example, 1 to 1, or 1 to *n*). The editor also allows you to change or delete existing application links between instances.

You can use the **Links Editor**, for example, to map a signal to an application object such as a limit switch signal to the corresponding input of a valve motor.

Some instances require that you link them to other instances depending on the interface rules of their templates (for example, you need to link the instance of a motor starter template \$TesysTMB to a communication instance such as a port \$MBPortQ); otherwise the status of the **Link**, page 171 of the device instance remains **Invalid**.

Link any instance with an **Invalid Link**, page 197 status with the **Links Editor** to change the status to **Valid** since it is prerequisite to proceed further in the system engineering life cycle.

NOTE:

- Linking instances by using the **Links Editor** helps avoid having to create such links later by refining the Control Participant project and provides a level of consistency in terms of Control resource names.
- Linking an instance to itself is not allowed. If this is required, edit the template used by the instance or refine the Control project to which facets of the instance are assigned.

Naming Rules for Variable Names

The name of the variable that is created by linking two instances needs to satisfy the naming rules, page 89 for the subsequent generation process to complete successfully.

Instance and Links Appearance

In the **Links Editor**, instances appear as follows:

- **With a green outline:** Applies to the instance for which you have selected the **Edit Links** command and, which is referred to as *active*. Only one instance can be active in the same **Links Editor** tab.
- **With a gray outline:** Applies to instances that you can link, which are referred to as *inactive*.

NOTE: Active and inactive instances can be selected in which case a green border appears around them.

The following colors are used for application interface links:

- Orange: Default color.
- Blue: When an instance is selected to highlight links that are connected to it.

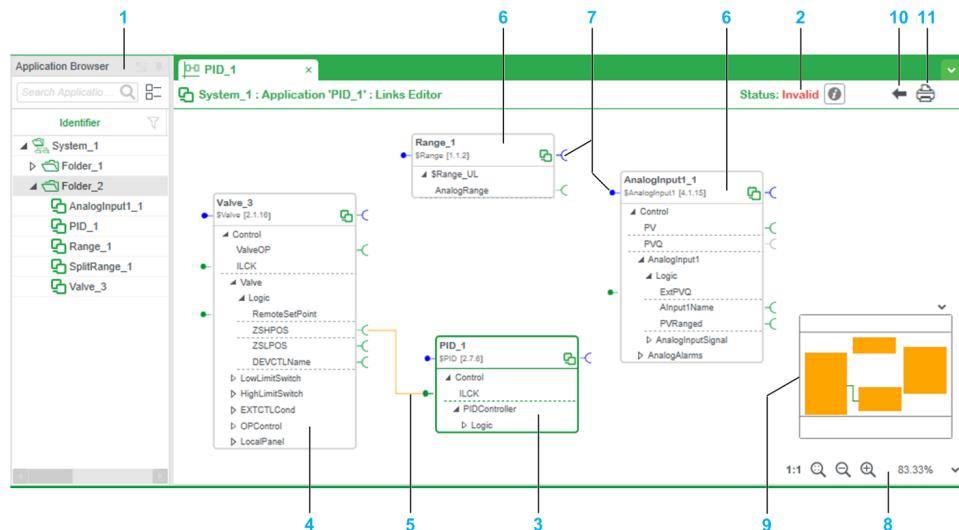
Opening the Links Editor

To open the **Links Editor**, proceed as follows.

Step	Action
1	<p>Right-click the instance that you want to link to one or more other instances, and select Edit Links.</p> <p>Result: The Links Editor tab opens inside the work area, page 105 of the Application Explorer and displays the selected instance in the center of the editor. The instance appears as <i>active</i>, page 943.</p> <p>NOTE: If the active instance is already linked to other instances, these are shown and appear as <i>inactive</i>.</p>

Links Editor

The following figure shows an example of the **Links Editor**, which is open in the work area of the **Application Explorer**.



Item	Description
1	Application Browser pane of the Application Explorer from which you can drag instances that you want to link.
2	Indication of the Link status of the active instance and button to display the Validity Diagnosis window if the status is Invalid.
3	Active, page 943 instance from which you have selected the Edit Links command.
4	Inactive instance, which was already linked with the active instance or which you have linked.
5	Application interface link.
6	Inactive instances, which you have dragged from the Application Browser pane to the editor and on which you can link with the active instance.
7	RTNS interfaces, page 201 (runtime navigation services). Appear in blue and are located in the header section of the instance.
8	Zoom tools, page 869.
9	Radar View, page 877
10	Reverts the view to the section in the Assignment Editor from where you have opened the Links Editor. The command is available only when you open the Links Editor from a Control facet. NOTE: The command does not close the Links Editor, which is required to perform the automatic generation of the section, page 394.
11	Lets you print the visible portion of the workspace.

NOTE: You can also move the editor background, page 106.

Impact of Modification of Instance Properties

When you edit an instance by using the **Instance Editor** and saving the change disables an interface that was linked, the link is automatically removed.

This may set the **Link** status, page 197 of either linked instances to **Invalid**.

Viewing Interfaces and Interface Description

Not all interfaces of instances that are displayed in the **Links Editor** are visible by default. You may need to expand nodes of the interface to show enabled (green) or disabled (gray) interfaces.

To enable an interface, edit the instance and select the element that contains the interface, and/or configure the element parameter that enables it.

Move the pointer over any interface of an active or inactive instance in the **Links Editor** to display a description of the interface in a tooltip.

NOTE: An interface may be compatible, unbound, and enabled but cannot be connected, page 946.

Viewing Linked Instances

Right-click a link between two instances and select **Go to**. A submenu opens and displays:

- The identifiers of the two instances that are linked.
- For each instance, the name of the element that exposes the interface and its path.

Click an entry to view the instance and highlight the element in the editor.

Linking Instances by Using the Links Editor

Creating Interface Links

To create an interface link between two instances, proceed as follows.

Step	Action
1	<p>From the Application Browser pane, drag the instance that you want to link next to the active instance, page 943, which is displayed in the Links Editor.</p> <p>Result: The instance is shown next to the active instance as inactive.</p> <p>NOTE: You can drag additional instances to the editor.</p>
2	<p>From the active or inactive instance, position the pointer over an interface and when the pointer turns into a hand, click and move the pointer away from it while holding down the left mouse button.</p> <p>Result: A green line appears and follows the pointer. You can release the mouse button.</p> <p>NOTE: Press ESC to return to the normal pointer function.</p>
3	<p>Move the pointer over the other instance.</p> <p>Result: The software automatically highlights compatible and enabled, page 946 interfaces of the instance and snaps the green line to the nearest matching connector.</p>
4	<p>Click to confirm the connection or move the pointer to another highlighted interface until the line snaps to it.</p> <p>Result: When you click, the software:</p> <ul style="list-style-type: none"> Links the interfaces of both instances with a line. Changes the status of the active instance to Valid if the interface rules are satisfied (if applicable). <p>NOTE: To delete the link, right-click the line and select Delete.</p>
5	<p>Verify that the status of the active instance is Valid; otherwise click the validity status information button  to display the Validity Diagnosis window and make the required links until the instance status becomes Valid.</p> <p>NOTE: You can only make links between an active and an inactive instance. To link the inactive instance with another inactive instance, right-click the header of the inactive instance and select Edit Links. The inactive instance becomes active.</p>
6	Repeat steps 1...5 until you have made the required links.
7	Close the Links Editor tab.

Removing Instances from the Links Editor

You can remove inactive instances from the **Links Editor**.

To do so, right-click the header of the inactive instance that you want to remove and select **Remove Instance**. This also removes links that may exist with the active instance after you confirm the command.

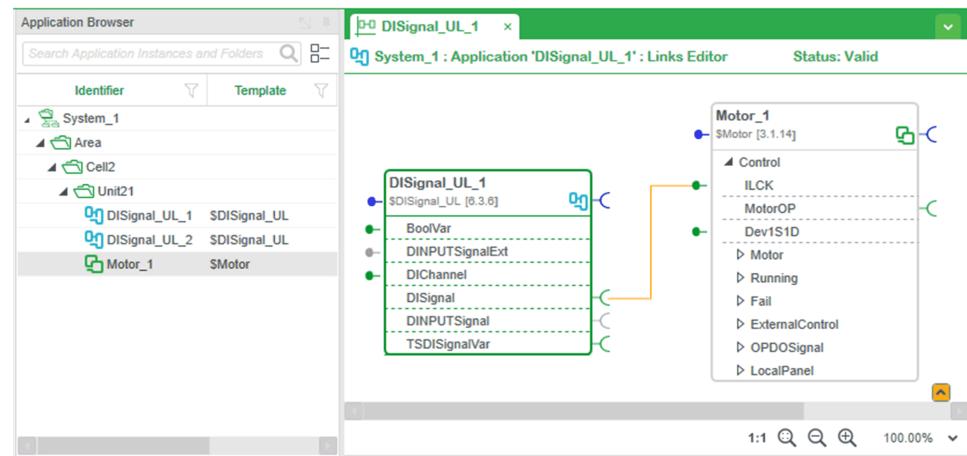
NOTE: The software does not remove interface links that the instance that you are removing may have with other instances, and which are not displayed in the current **Links Editor** tab.

Enabled Interfaces That Cannot Be Connected

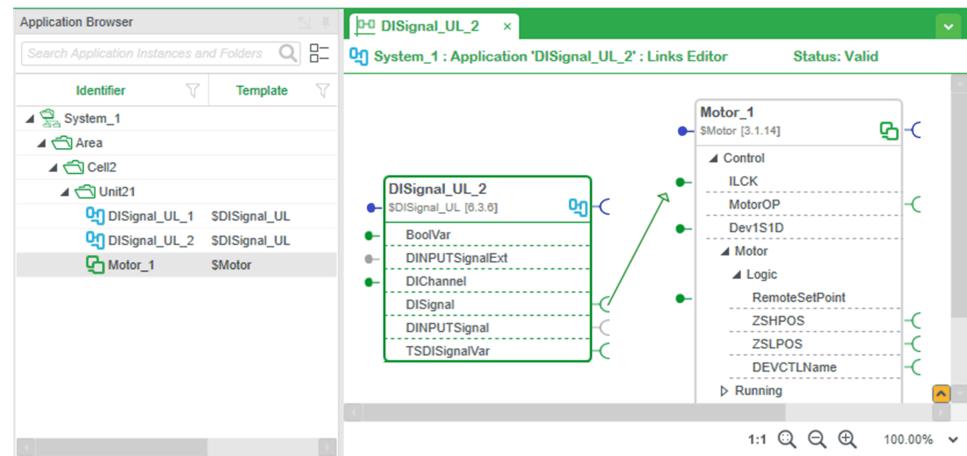
The interface of an inactive, page 943 instance may appear as enabled, unbound, and be compatible but the link does not snap to it. This can be the case if a link with the interface exists but is not visible and the interface accepts only one connection at a time.

To view the links of an instance, open it in the **Links Editor** so that it is the active instance or open the **Inspect Instance** window, page 204.

In the following example, interface *ILCK* of *Motor_1* is connected to *DISignal_UL_1* and the **Links Editor** is closed.



When you open the **Links Editor** for *DISignal_UL_2* and drag *Motor_1* to the editor, its *ILCK* interface appears enabled. Yet, the link being created from *DISignal_UL_2* does not snap to *ILCK*. This is because *Motor_1* is an inactive instance and its links to other instances (*DISignal_UL_1*) are not shown in this view.



Glossary

A

application interface:

Mechanism that allows sharing data and manage dependencies between 2 instances/references.

The *application interface* allows you to link:

- applications instances/references to each other
- topological instances/references to each other

application link:

Links describing connections between instances of the application that are made by using application interfaces.

application template:

Global Template contained in the Global Templates library that models an object of the application of a system and its associated functions. The template encapsulates the necessary Participant capabilities.

Application templates are instantiated and can be configured to create the application of a system.

application:

The *application* models the complete functionality provided by the system through an application folder structure and instances of application templates.

area:

The *area*, when referring to access control, defines a topological, functional, or another user-based rule to restrict access in the scope of an automation system.

B

binding:

The *binding* is the key mechanism that establishes relations between the following items of the facet and composite templates:

- Parameters
- Interfaces
- Elements

built Participant project:

The *built Participant project* is a generated, optionally refined, and built project, which is created by the corresponding Participant and that exists at the platform level in the form of an executable.

For example, files with *.stu* and *.ctz* extensions that are represented by Control and Supervision executables respectively are built Participant projects.

C

cardinality:

The *cardinality*, in the context of the interface model, is the number of connections that are supported by the objects playing the other role of the interface model.

communication channel:

A *communication channel* is the logical representation, at the platform level, of communication between controllers or between a controller and devices.

For example, a controller I/O scanner row is modeled by a *communication channel* for Modbus TCP communication in the executable of the Control project.

communication interface:

Mechanism that allows sharing data and manage dependencies between 2 instances or references.

Communication interfaces allow the platform to link topological instances/ references at the logical level (for example, I/O scanner, OPC Factory Server, Supervision I/O devices).

communication link:

Links describing logical connections between topological instances that are made by using communication interfaces.

For example, the communication link between 2 controllers exchanging data.

communication mapping:

The *communication mapping* process defines the communication aspects of a logical Participant project after being mapped to the topology.

composite template:

The *composite template* combines the capabilities of 1 or more facet templates, each encapsulating functionalities provided by one software Participant, and/or of other *composite templates*.

composite:

The *composite* is an instance of a composite template.

constituent encapsulation:

Process that is performed by using the **Facet Editor** and that allows you to:

- Select constituents that are created with a software Participant.
- Optionally, modify constituents inside the corresponding Participant.
- Include the selected constituents inside a facet template.

constituent:

Set of data provided by a software Participant, which can be global or local.

contents repository:

The *contents repository* is a global storage provided by the platform to manage global constituents and other files, such as Participant projects, used by the libraries and the systems.

D

deferring mechanism:

Mechanism that allows you to make available in the composite template that is at the highest level optional element selections, parameters, and/or interfaces of its references.

deployed Participant project:

A *deployed Participant project* is a built project that has been transferred to a controller or other entity of the topology acting as an engine, and that you can execute.

DFB type:

The *derived function block type* is a programming language element that consists of:

- A data structure definition partitioned into input, output, public, and internal variables.
- A set of operations written in IEC 61131 programming languages to be performed upon the elements of the data structure when an instance of the function block type is started.

E

EcoStruxure Process Expert:

Third generation name of the software platform. Formerly, StruxureWare Process Expert then, EcoStruxure Hybrid DCS.

element:

Element is the term that is used to describe the contents of templates at the next lowest level as well as the contents of interface models:

- For facet templates, elements are the constituents that the facet encapsulates.
- For composite templates, elements are:
 - Facet references
 - Composite references
- For interface models, elements are the data that is shared. See *interface elements*.

You can define properties and rules for elements during the Global Templates definition stage.

engine:

An *engine* is the projection of the hardware and software defined in the topology that is able to execute the deployed Participant project.

engineering client:

EcoStruxure Process Expert client connecting to the system server that you use to engineer and maintain systems. It can run on the system server and/or an engineering station.

Engineering clients use software Participants.

engineering station:

The *engineering station* is a computer running an EcoStruxure Process Expert engineering client to design, and maintain systems that exist on the server to which the engineering client connects.

executable:

The *executable* is a project component that represents the built Participant project. It contains the mapping information.

execution domain:

Property of the executable of a Participant project, which serves as a filter for selecting the scope of applicable services.

For example, it is used to define the boundaries for runtime navigation services or to restrict the exchange of data through peer to peer communication between projects that have executables with the same execution domain.

F

faceplate:

The *faceplate* is a component of a Supervision genie that provides a user interface to interact with the object that it represents.

facet element:

The *facet element* is the contents that can be accessed at the system level. Depending on the type of facet template, such elements can be either:

- The visible part of the constituents stored in the facet templates.
- A set of data defined by the facet template.

facet template:

The *facet template* is the smallest available template that encapsulates a specific capability provided by 1 software Participant only.

facet:

The *facet* is an instance of a facet template.

G**genie:**

Animated graphic that is assigned to Supervision pages and that represents functionalities of instances of the application during operation.

Genies are encapsulated in Supervision facets.

global constituent:

Set of data provided by a software Participant that is a common resource referenced by several Global Templates. Global constituents are stored in the contents repository.

Global Templates:

Global Templates encapsulate one or more functions that can be customized and reused in any system created with EcoStruxure Process Expert. They are stored in the Global Templates library.

H**hardware mapping:**

The *hardware mapping* process links the logical projection of the hardware that is defined in the application and assigned to projects to the hardware defined in the topology of the system.

For example, linking Control variables representing I/O signals to I/O channels of an I/O module.

HMI:

Abbreviation for *human machine interface*. It is a graphical operator interface for industrial equipment.

I**instance element selection:**

The *instance element selection* is a mechanism of the instantiation stage that allows you to customize an instance by selecting services that are provided by the template that the instance uses.

instance parameter:

Instance parameters are properties of the elements of an instance that you may be able to customize.

instance:

Abbreviation for object instance. It is the result of the instantiation of a template.

instantiation naming convention:

The *instantiation naming convention* defines the naming strategy that the platform applies when you create instances.

interface element:

Represents the data that will be shared through an interface. Interface elements are defined during in the interface definition. Interface elements can be transformed by the interface by using expressions.

interface link:

Links describing logical connections between instances that are made by using interfaces.

The following types of links exist:

- Physical links
- Communication links
- Application links

Each type of link is made by using the corresponding interface (physical, communication, or application interface).

interface model:

The *interface Model* is a type of template that is available in the Global Templates library and that you can configure.

interface:

The *interface* is a reference, inside a composite or facet template, of an interface model. Interfaces are exposed by the facets of an instance, allowing you to make different types of links with other instances to share data.

In the context of the template definition, interfaces are a mechanism to define the links between references by declaring compatibility and/or requirement rules.

I/O:

Abbreviation for *Inputs/Outputs*.

IEC:

The *International Electrotechnical Commission* is a non-profit and non-governmental international standard organization that prepares and publishes international standards for electrical, electronic, and related technologies.

L

local constituent:

Set of data that is provided by a software Participant, encapsulated inside a facet template but not used at the system level. The local constituent is used to generate the contents of the logical Participant project. An example is Control Participant code that is encapsulated in a Control facet template.

logical Participant project:

The *logical Participant project* is a generated and refined project, which is created by the Participant but, which is not associated to the topology.

NOTE: The refinement of the project is optional.

M

mapping interface:

Mechanism that allows sharing data between two facets.

Mapping interfaces allow you to perform the hardware mapping. It is the process whereby you link facets assigned to projects to facets representing the hardware defined in the topology through matching mapping interfaces that these facets expose.

MES:

The *Manufacturing Execution Systems* is a control system for managing and monitoring work-in-process in a factory plant.

N**network variable:**

The *network variable* is a peer to peer communication mechanism allowing you to share data between 2 or more Control projects.

O**object template:**

An *object template* is a generic term that covers several reusable templates such as facet or composite templates.

OFS:

Abbreviation for OPC Factory Server.

operation client:

An EcoStruxure Process Expert client connecting to the system server that you can use during runtime to monitor and troubleshoot a system.

operator station:

The *operator station* is a computer running a Supervision client software.

P**Participant services:**

Participant services are the functions provided by a software Participant when interacting with EcoStruxure Process Expert.

peer to peer communication:

Peer to peer communication is a data exchange mechanism between 2 or more Control projects, which uses the I/O scanner function of the controller acting as client.

physical (interface) link:

Links describing logical connections between topological instances that are made by using physical interfaces.

For example, the connection of a controller to an Ethernet network.

physical connection:

Link between topological entities representing the physical connections between controllers, station nodes, devices, and communication networks.

physical interface:

Physical interfaces allow the platform to link, at a physical level, topological instances to model the topology of the system.

platform:

Abbreviation for system platform. Represents the services that are provided by EcoStruxure Process Expert apart from the software Participants.

privilege:

Defines groups of functions that are provided by an application and granted to users through access control.

project container:

The *project container* is an organizational structure of a project to organize the facets that are assigned to it. Such a structure models the ones managed by the corresponding software Participant, and that are visible at the system level.

project facet:

Facet that is assigned to a project.

project:

Component of a system associated to a software Participant. Its structure contains elements that are managed by the software participant.

R

redundant controller:

Generic term that is used to refer a Quantum Hot Standby controller and/or an M580 redundant controller.

reference:

Defines facet and composite templates, which are contained inside other facet or composite templates in order to distinguish templates, which are used in the composition of other templates from the highest level templates, such as control modules.

role:

The *role* can have 2 different meanings,

- For access control:
The *role* groups functionalities to grant different levels of user rights, which combine areas and privileges to fulfill a set of services.
- For interface models:
The *role* defines the 2 sides of an interface, role A and role B.

runtime navigation services:

Describes the complete set of functionalities that are provided by EcoStruxure Process Expert operation client in runtime, such as process monitoring, viewing of instance information, diagnostics.

S

service mapping:

The *service mapping* links the execution capabilities of a project represented by the executable to an engine of the topology.

For example, it can link:

- The I/O server of a Supervision project to a station node representing the operation server.
- The executable of a Control project to a controller.

service:

In the context of projects, the *project service* is an organizational structure of a project to organize the execution capabilities. It models the structures that are managed by the corresponding software Participant, and that are visible and configured at the system level.

In the context of templates, a *service* represents a functionality provided by a Participant under the form of a facet referenced by the template.

software Participant:

An external tool that is installed on the local computer and which collaborates with platform components. For example, the Control Participant.

software:

Refers to EcoStruxure Process Expert.

station node:

A *station node* represents a computer that can act as an engine to execute a Control project by using software emulating a simulator, or a Supervision project by using Supervision software.

subnet mask:

A 32-bit value that indicates the number of available host addresses on a subnet, which uses TCP/IP knowing that the first and last addresses are reserved (for example, subnet mask 255.255.192.0 allows 317 addresses for classless devices). It also indicates whether addressing is classful or classless. For classful addressing, it indicates the class of the network (for example, 255.255.255.0 is the subnet mask for class C networks).

Supervision client:

Client connecting to the Supervision server that provides runtime services to operate and monitor automation systems. The client runs on an operator station and/or operation server.

Supervision server:

Server running on the operation server or on the system server, and that provides runtime data to Supervision clients.

system engineering life cycle:

The *system engineering life cycle* represents the stages to engineer a system in EcoStruxure Process Expert.

system server:

EcoStruxure Process Expert server that hosts:

- Software Participants
- The database containing template libraries and system data

The system server manages requests from the EcoStruxure Process Expert clients.

system:

Models a physical automation system.

The system consists of the following components:

- Topology
- Application
- Participant projects

T

topological entity:

A *topological entity* is the representation of piece of hardware infrastructure.

For example, a controller.

topology:

Models the hardware and software infrastructure of a system through topological folders and entities.

Index

\$EGenericDeviceHW template	generic device mapping interfaces	603
\$GenericDevice	implementing communication.....	781
\$GenericNOC_UL	implementing communication.....	781
\$GenericPBSlave template	generic device mapping interfaces	603
\$GenNOCDevice	implementing communication.....	781
\$PRMMgt templates	managing Modbus TCP communication (I/O scanning).....	785
\$System	\$System parameters of instances.....	184
A		
aborting		
abort task icon	73	
application import command	248
back up command.....	143	
build command (control)	633	
build command (supervision)	645	
Deploy Built Project command (supervision station node).....	700	
Deploy Built Project command (supervision)	695	
Deploy Changes command (supervision).....	703	
device export command	565	
device import command.....	571	
export command	221	
export command (control).....	269	
export command (supervision)	310	
export command (Topology).....	553	
folder delete command	165	
Generate and Build command for projects	386	
Generate command for project containers.....	415	
Generate command for projects	414	
import command (control)	282	
import command (supervision)	319	
import command (Topology).....	561	
instance delete command	173	
paste command	207	
Purge command	897	
ReGenerate command for projects	422	
restore command	148	
template export command	886	
template import command	894	
Update Control Project command.....	756	
About Process Expert		
About dialog box	104	
opening the about dialog box	101	
actions		
on assigned facets	377	
on generated facets	431	
activity		
log files	80	
notification panel.....	73	
addresses		
generating network addresses for supervision components	644	
identifying engines	651	
Advantys		
selecting templates	504	
alarm server redundancy		
implementing	305	
analysis		
analyzing configuration changes (M580 safety) ..	770	
analyze issues		
troubleshooting control Participant projects	439	
validity indicators	632	
Analyze Project		
automatic generation		
automatic generation setting	263	
control project settings.....	263	
generation		
automatic generation setting	263	
analyzing		
control projects during generation.....	388	
application		
generating system documentation reports.....	929	
import rules for control project-related objects	272	
import rules for supervision project-related objects	314	
modeling the application of systems	152	
Application Browser pane		
presentation	161	
using and customizing	161	
Application Explorer		
accessing	155	
opening	101	
presentation	155	
application export files		
using the invariant culture	228	
application folders		
actions	165	
changing folder structure	209	
changing the application hierarchy	206	
copying and pasting	207	
creating	165	
managing	203	
moving	209	
properties	166	
application objects		
exporting and importing to XML or CSV	219	
application passwords		
changing the application password of control projects	662	
clearing the application password of control projects	662	
managing passwords of control projects	658	
resetting a forgotten application password	663	
setting the application password of control projects	661	
application structure		
tree view description in Containers pane	371	
tree view when refining Control projects	438	
tree view when refining Control projects online ..	728	
application templates		
description	28	
Asset Workspace Editor		
using the radar view	877	
using the simplified view	196	
Asset Workspace pane		
asset workspaces		
exporting	219	
assigned facets		
generating automatically assigned facets	386	
managing assigned facets	376	
unassigning generated facets	432	
assigning		
Assignment Viewer description	359	
defining container names.....	350	
facets	349	

facets (control).....	362
facets (supervision).....	364
Assignment Editor	
presentation	330
assignment stage	
completing.....	328
concept.....	44
Assignment Viewer	
description.....	359
View Assignments window.....	379
assignments	
automatic assignment of remaining facets.....	368
automatic assignment of Supervision facets.....	370
viewing by instance	379
audit trail	
functionality description	76
authentication	
logging in to server and clients	60
automation system	
description.....	33
automation systems	
description.....	23
B	
back up command	
aborting.....	143
backing up	
backing up systems.....	142
scheduling system backups	145
backing up data	
backing up controller data	733
backup	
password information and settings in system	
backups.....	143
backup files	
managing data backup files (control)	286
batch	
modeling the equipment hierarchy.....	154
BMENUA0100	
variable tag addresses	643
boards	
creating communication boards (supervision)....	302
bold formatting	
editing instance properties	182
browsers	
Templates Browser presentation	159
build command	
aborting (control).....	633
aborting (supervision).....	645
build stage	
completing.....	628
concept	53
control projects	629
supervision project	642
build states	
when deploying (supervision).....	695
build status	
actions changing the build status (control).....	636
building the logical supervision project.....	645
changes affecting the build status (control).....	634
verifying before deploying (control).....	667
when deploying from station node (supervision).....	698
building	
building safety control projects	774
creating the built Control Participant project	630
network addresses of supervision components...	644
building changes (control)	
building with device DDT variables	637
building control projects	
building with device DDT variables	631
Generate and Build command.....	632
project settings	632
building supervision projects	
Generate and Build command.....	645
built Participant projects	
changing (control)	636
changing (supervision)	647
C	
calculated variable tags	
I/O device properties	300
canceling	
canceling generation for version conflict (control).....	389
capabilities	
hardware.....	83
programming	83
CCOTF	
refining projects offline.....	436
using CCOTF with S908 architectures	505
certificate validity	
About dialog box	104
changes	
actions changing the build status (control).....	636
deploying changes in hardware mappings.....	686
deploying changes to engines (control).....	682
managing hardware mappings after deploying ...	608
saving changes in Manage Network Variables dialog box.....	461
changes log	
accessing	874
interfaces	855
saving changes in Global Templates.....	882
CHM help	
HTML help for user-created templates	96
using the HTML help for several versions of templates.....	94
circular links	
generating	389
citect.ini files	
deploying	693
deploying from station node	698
deploying to station nodes from executable.....	695
clients	
connection to the server	64
creating (supervision)	297
engineering client UI description.....	101
logging in and out.....	60
notification panel	73
quitting	68
restarting after closing unexpectedly.....	67
supervision client properties.....	298
working with several clients.....	70
closing	
closing the software	68
clusters	
actions	295
creating	295
properties	295
collapsing	
nodes in the Application Browser pane	163
nodes in the instance pane	179
color coding	
importing application objects	250
importing devices from CSV.....	573

color indicators	25
instance color icons	188
project color icons	321
communicating	
with devices through Ethernet explicit	
messaging	789
with devices through EtherNet/IP explicit	
messaging	795
with devices through Modbus explicit	
messaging	799
communication	
changing EtherNet/IP device properties	535
changing Modbus TCP device properties	535
changing STB islands properties	524
communicating with the computer for deployment	
(control)	657
EtherNet/IP explicit messaging	794
implementing I/O scanning	780
implementing I/O scanning with devices	781
implementing I/O scanning with PRMs	785
implementing I/O scanning with STB islands	784
implementing peer to peer communication	787
managing	779
mapping (control)	582
Modbus explicit messaging	798
Modbus TCP explicit messaging	788
task workflow	124
Variable Mapping Editor window	594
communication boards	
creating (supervision)	302
driver selection	302
properties (supervision)	302
communication channels	
actions	585
adjusting properties	591
creating	590
moving	592
properties	586
properties server side	588
unmapping	591
communication mapping	
Communication Mapping Editor	583
prerequisites (control)	582
with Modbus serial devices	624
with Modbus TCP devices	624
communication module	
removing from the configuration	511
communication networks	
connecting a station node to the Ethernet	543
connecting an STB island to the Ethernet	545
creating	538
disconnecting a station node from the Ethernet	544
disconnecting an STB island from the Ethernet	546
communication ports	
creating (supervision)	302
creating additional ports (supervision)	303
properties (supervision)	303
components	
creating supervision project components	294
Composite Editor	
changes log	874
menus and toolbars	868
panes	874
parameter pane	874
presentation	864
using the radar view	877
using the simplified view	878
composite templates	
changes log	874
description	25
modification strategy	899
configuration	
changing EtherNet/IP device properties	535
changing Modbus TCP device properties	535
changing STB islands properties	524
changing the controller configuration	511
changing the PRM configuration	529
configuring M580 safety controllers	769
deploying hardware changes	685
updating configuration after deploying changes	
online	727
updating controller configurations	748
updating hardware changes	750
viewing instance configuration	204
configuration change on the fly	505
configuration file	
OFS/OPCUA configuration file description	643
configuration stage	
completing	498
concept	49
configuring	
configuring HART modules	536
instance parameters	182
connection	
client-server connection	64
physical connection import rules	558
redundant configuration connection scenarios	744
consistency checks	
after import	271
before updating control projects	754
code lock feature	442
fixing inconsistencies	451
performing (control)	450
rules	449
consistency management	
generating inconsistent facets	419
refining control projects	449
consumer	
about network variables	452
managing network variables	459
owner and consumer projects	452, 459
consumer project	
description	452
Container Naming Rules	
window description	350
containers	
creating and managing	329
Containers pane	
displaying sections in tree view	333
content	
resolving contents conflicts on import	893
content repository	
actions	939
content description	937
deleting temporary files	940
explorer	936
opening	101
root folder actions	938
User Contents folder	936
Content Repository explorer	
opening	936
Control Constituent Password	
managing Control facet template passwords	849
Control Expert projects	
importing a Control Expert project	284
Control facet templates	
managing Control facet template passwords	849
Control Facet Templates	

Global Template settings	849
control modules	
description.....	25
template modification strategy.....	899
Control Project Browser	
description.....	258
control project sections	
actions	335
control projects	
actions	260
build stage.....	629
changing built Participant projects	636
communication mapping import rules.....	275
consistency management.....	449
creating	260
creating FBD and non-FBD sections.....	264
creating FBD sections	335
creating M580 safety control projects.....	761
creating the built Participant project.....	630
data backup file import rules	279
deleting	262
deploying	649
deploying changes to engines	682
description of tree view mode.....	371
executable import rules	275
executing	707
export settings	268
exporting	267
facet import rules	274
generating after changes	417
generation specifics	388
hardware mapping import rules	275
import settings	271
importing	271
importing a Control Expert project	284
managing	265
managing passwords of control projects	658
mapping stage	577
peer to peer communication export settings	269
peer to peer communication variable import rules.....	277
prerequisites.....	260
Project Explorer	259
Project Explorer and Browser.....	255
project import window.....	279
properties	261
refining.....	435
refining locked code	442
regenerating	421
related project data import rules	272
section import rules	273
selecting type version changes	396
service mapping import rules	275
starting the simulator with your control project	655
starting the simulator without password protection	655
task quick reference	128
task workflow.....	124
tree view mode when refining.....	438
tree view mode when refining online	728
updating projects after deploying changes online	727
Control projects	
managing system access passwords.....	138
Control service	
creating in station node.....	517
ControlExecutables	
actions	578
assigning execution domains	579
creating	578
properties	579
ControlExpert services	
actions for station nodes.....	522
properties for station nodes.....	522
controller passwords	
resetting a forgotten controller password.....	663
controller stop	
when deploying changes	692
controllers	
about network variables.....	452
actions	507
assigning IP addresses	511
backing up controller data.....	733
changing the application password of control projects	662
changing the configuration	511
clearing the application password of control projects	662
communicating with the computer	657
configuring	511
configuring the I/O scanner	512
connecting to Ethernet networks	542
creating	507
creating the system topology.....	499
deleting	514
deploying built projects	667
deploying changes (control).....	682
deploying data	677
deploying to redundant controllers.....	665
disabling the I/O scanner	511
disconnecting from Ethernet networks	542
execution with redundant controllers.....	711
exporting	552
import rules	557
importing	555
managing data backup files	286
managing network variables	459
mapping (control).....	580
memory size configuration	509
properties	509
redeploying built control projects	672
representation	512
selecting templates	504
setting the application password of control projects	661
starting and stopping	713
supported platforms	83
copying	
copy-paste application folders	208
copy-paste instances	207
Global Templates and folders	885
copying instances in Control projects	377
counting	
counting instances	174
CRA modules	
connection with device I/Os	780
Create Instance dialog box	338
creating	
Global Template editors	853
creating M580 safety control projects	761
CSV	
application export file description.....	223
editing the application export file	227
exporting and importing application objects to XML or CSV	219
exporting and importing devices	564
importing CSV files in the topology	571
importing CSV files into the application	246

csv device export files	775
using the invariant culture	568
csv export files	698
using the invariant culture	228
ctz files	741
changing (supervision)	647
creating (supervision)	642
deploying	693
deploying changes to station nodes	703
deploying from station node	698
deploying to station nodes from executable	695
redeploying to engines from executable	696
redeploying to engines from station node	702
D	
data	651
backing up controller data	733
deploying controller data to engines	677
instance validity status	180
managing data backup files (control)	286
sorting data	118
using filters	118
data backup	733
with redundant controllers	733
data backup files	286
import rules for data backup files (control)	279
managing data backup files (control)	286
data deployment	649
to redundant controllers	677
data types	693
conversion to Participant types	86
data types used by the software	86
validating instance parameters	182
DDDT	631
building changes in control projects	637
building control projects	631
creating during build	630
deploying changes to control projects	687
for peer to peer communication	631
managing device DDT variables	445
naming convention	630
naming convention for P2P communication	631
DDT	507
device DDTs in the configuration	507
deferring parameters automatically	849
delete command	165
aborting folder delete	165
aborting instance delete	173
deleting	514
control projects	262
controllers	514
deleting FBD sections during update Control project	754
PRMs	530
sections	335
STB islands	526
unmapping hardware mappings	607
Deploy Built Project command	700
aborting (supervision station node)	700
aborting (supervision)	695
Deploy Changes command	703
aborting (supervision)	703
deployed control projects	727
editing online	727
deploying	602
controller data to engines	677
deploying changes to engines (control)	682
deploying to M580 safety controllers	775
from station node (supervision)	698
main redundant configuration scenarios	741
preparing for deployment	694
to engines (control)	667
to redundant controllers	665, 667
to station nodes from executable (supervision)	695
deploying changes	666
changes requiring to stop the controller	692
deploying device DDT variable changes	687
hardware mappings	686
main redundant configuration scenarios	741
selecting IP addresses	683
to redundant controllers	683
to redundant controllers (control)	666
to station nodes (supervision)	703
deploying control projects	651
supported engines	651
deployment	650
communicating with the computer (control)	657
process flowchart	650
deployment stage	55
completing	648
concept	649
control projects	649
supervision project	693
device DDT	637
building changes in control projects	637
building control projects	631
creating during build	630
deploying changes to control projects	687
device DDTs in the configuration	507
for peer to peer communication	631
naming convention	630
naming convention for P2P communication	631
device DDT variables	445
managing device DDT variables	445
Device Editor	514
hardware module properties	514
device export files	568
using the invariant culture	568
DeviceConfig.xml file	642
creating	642
deploying	693
saving	619
devices	532
actions for EtherNet/IP devices	532
actions for Modbus TCP devices	532
assigning IP addresses to topological I/O device entities	532
changing IP addresses in explicit messaging	793, 797
communicating through Ethernet explicit messaging	789
communicating through EtherNet/IP explicit messaging	795
communication through Modbus explicit messaging	799
connecting to Ethernet networks	544
connectivity with supervision	831
creating EtherNet/IP devices	531
creating Modbus TCP devices	531
disconnecting from Ethernet networks	545
exporting	564
importing CSV files in the topology	571
integrating PROFIBUS devices	836
managing Modbus TCP communication (I/O scanning)	781
mapping I/Os (control)	602

properties of EtherNet/IP device entities	532
properties of Modbus TCP device entities	532
resolving import conflicts	574
selecting topological templates	504
diagnostic messages	
during generation	427
DIO scanner	
implementing communication by I/O scanning	780
disk variable tags	
I/O device properties	300
docking menus	
creating additional work areas.....	108
documentation	
application data	929
generating system documentation reports.....	927
Global Templates data.....	933
project data	929
topological data	931
downloading	
controller data to engines.....	677
deploying changes to engines (control)	682
deploying to engines (control)	667
main redundant configuration scenarios.....	741
redeploying to engines (control)	672
to redundant controllers.....	665
dragging	
moving the workspace around.....	106
drivers	
creating communication boards and ports	302
DTM	
naming convention	630
naming convention for P2P communication.....	631
duplicating	
Global Templates	912

E

Edit/Extend Interface	
command description	917
editing	
deployed control projects online	727
editing instance properties	
bold formatting.....	182
italic formatting	182
validity status.....	182
editing instances	
changing element selection after assignment	367
editing interfaces	
changes log	855
editing online (control)	
changes requiring to stop the controller.....	692
overview.....	723
editors	
Assignment Editor.....	330
Composite Editor	864
Facet Editor	860
Interface Editor	854
Links Editor	943
opening Global Template editors	853
elements	
definition	25
replacing templates of elements	907
updating templates of elements.....	901
End User License Agreement	
viewing the End User License Agreement	104
engineering client	
engineering client work areas.....	105
engineering client toolbar	101
engineering clients	
engineering client UI description.....	101
engines	
backing up controller data.....	733
deploying built control projects	667
deploying data (control)	677
deploying to redundant controllers.....	665
execution with redundant controllers.....	711
managing data backup files (control)	286
mapping (control).....	580
redeploying built control projects	672
refreshing engine status and connecting	657
supported engines for deployment (control).....	651
entities	
creating and configuring topological entities	498
entity identifiers	
naming rules.....	88
entity nodes	
description.....	500
enumerations	
exporting parameter values to CSV	228
exporting parameter values to XML	238
viewing values in the Instance Editor	176
equipment	
equipment hierarchy.....	154
generating equipment (supervision).....	424
equipment generation	
troubleshooting	427
equipment hierarchy	
import rules for equipment (supervision)	315
errors	
detected errors during generation	389
Ethernet	
conflicting Ethernet connections	541
connecting controllers to Ethernet networks	542
connecting entities to an Ethernet network	540
disconnecting controllers from Ethernet	
networks.....	542
managing EtherNet/IP explicit messaging	794
managing Modbus TCP explicit messaging	788
viewing Ethernet connections.....	540
Ethernet IP devices	
import rules	557
Ethernet networks	
actions	538
connecting devices	544
connecting PRMs.....	546
connecting station nodes	543
connecting STB islands	545
creating	538
disconnecting devices	545
disconnecting PRMs	547
disconnecting station nodes	544
disconnecting STB islands	546
exporting	552
importing	555
properties	538
selecting templates	504
Ethernet RIO drops	
connecting to Ethernet networks	542
disconnecting from Ethernet networks	542
EtherNet/IP devices	
changing properties.....	535
creating and configuring devices	531
EtherNet/IP explicit messaging	
communicating with devices	795
managing communication.....	794
EULA	
viewing the End User License Agreement	104
Exclude command	

description.....	861
executable files	
changing (control)	636
creating (control).....	630
creating (supervision).....	642
deploying (supervision).....	693
modifying (supervision).....	647
executables	
actions (control)	578
actions (supervision)	619
assigning execution domains (control)	579
assigning execution domains (supervision)	621
backing up controller data.....	733
creating (control).....	578
creating (supervision).....	619
deploying changes to engines (control).....	682
deploying changes to station nodes (supervision).....	703
deploying controller data to engines (control)	677
deploying from station node (supervision)	698
deploying to engines (control)	667
deploying to redundant controllers	665
deploying to station nodes (supervision)	695
executable import rules (control)	275
executable import rules (supervision).....	316
executing with redundant controllers.....	711
managing data backup files (control)	286
properties (control).....	579
properties (supervision).....	620
redeploying to engines (control)	672
redeploying to engines (supervision)	696, 702
executables (supervision)	
status change when importing new content	
version	893
executing	
deployed participant projects (control)	713
to redundant controllers.....	711
execution	
facet execution order.....	381
FBD section execution order	266
process flowchart.....	708
synchronizing redundant controllers	719
execution domains	
assigning to executables (control)	579
assigning to executables (supervision)	621
execution order	
after first generation	390
when regenerating	422
execution stage	
completing.....	706
concept	56
control projects	707
supervision projects	757
expanding	
nodes in the Application Browser pane	163
nodes in the instance pane	179
explorers	
Application Explorer	155
Content Repository explorer	936
creating work areas in explorers	109
Global Templates Explorer.....	839
opening	101
opening the Project Explorer	256
overview.....	35
Systems Explorer.....	133
task workflow	124
Topology Explorer	500
export command	
aborting.....	221
aborting (control).....	269
aborting (supervision).....	310
aborting (Topology)	553
aborting device export	565
aborting template export.....	886
export files	
editing the CSV application export file	227
editing the XML application export file	238
editing XML to manage application links	243
using the invariant culture	228
using the invariant culture (devices).....	568
export settings	
control projects	268
peer to peer communication export settings	269
supervision projects	309
exporting	
CSV application export file description	223
editing exported hardware mappings	612
exporting application objects to XML or CSV	219
exporting control projects.....	267
exporting hardware mappings	610
password related information for topological entities	552
peer to peer communication data	267
supervision projects	307
templates	886
topological entities	552
XML application export file description	236
exporting asset workspaces	219
exporting control projects	
software version compatibility	267
exporting devices to CSV file	564
exporting M580 safety controllers	772
exporting safety control projects	763
exporting supervision projects	
software version compatibility	307
expressions	
Container Naming Rules window.....	350
importing expressions for control sections	271
extracted Participant windows	
troubleshooting	117
extracting	
software Participant windows	116

F

Facet Editor	
changes log.....	874
description.....	860
menus and toolbars.....	868
panes.....	874
parameter pane	874
using the radar view	877
using the simplified view	878
facet template passwords	
changing the password for Global Templates	850
clearing the password for Global Templates	851
facet templates	
changes log.....	874
description.....	25
modification strategy	899
facets	
actions after assigning	377
actions after generation	431
assigned facet description	366
assigning	349
assigning (control).....	362
assigning (supervision).....	364
automatic assignment of remaining facets.....	368

automatic assignment of Supervision facets	370
changing order inside section	381
element selection changes of assigned facets	367
generating automatically assigned facets	386
import rules for control project facets	274
managing assigned facets	376
moving	378
moving generated facets	433
navigating to the instance	382
reassigning	432
relinking	433
unassign	378
unassigning generated facets	432
unlinking generated facets	432
viewing assignments per instance	379
FBD	
using FBD language with Process Expert	83
FBD sections	
actions	335
changing execution order	266
creating during generation	390
creating FBD sections	335
deleting	335
import rules for FBD sections	273
inserting FBD sections	335
properties	336
when regenerating	422
file encryption	
applying to deployed control projects	141
files	
adding external files to supervision projects	497
filters	
filtering data	118
template editor element filters	876
Templates Browser	158
to display instance parameters	180
floating Participant windows	
troubleshooting	117
folder structure	
changing application folder hierarchy	166
changing the position of folders	211
folders	
changing the application hierarchy	206
copying and pasting application folders	207
copying and pasting library folders	885
creating Global Template folders	845
exporting and importing to XML or CSV	219
moving application folders	209
moving folders	166
moving topological folders	549
naming rules	88
resolving application import conflicts	252
S88 representation	170
searching application instances and folders	156
function blocks	
layout after first generation	390
layout when generating after changes	420
layout when regenerating	422
G	
gateways	
using for Modbus explicit messaging	799
Generate and Build command	
aborting	386
Building control projects the first time	632
Building supervision projects the first time	645
description	386
Generate command	
aborting generation of project containers	415
aborting generation of projects	414
generated code	
identifying software-generated code	442
generated facets	
unlinking	432
generating	
analyzing control projects	388
automatically assigned facets	386
configuring settings (control)	392
control project specifics	388
detecting errors during generation	389
diagnostic messages	427
entire projects	414
equipment (supervision)	424
facet order	381
FBD section size	390
first-time generation	413
function block layout during first generation	390
function description	417
general description	384
Generate and Build command	386
generating invalid facets again	419
generating safety control projects	767
importing sections after generation	416
invalid facets	387
part of a project	415
projects after changes	416
rules for generating graphical links	389
satisfying naming rules	385
section order	266
statuses of facets	411
generation	
canceling generation for version conflict (control)	389
changing type versions during generation (control)	396
function block layout after changes	420
managing generated facets	431
skipping facets during generation	387
generation name issues	
troubleshooting	427
generation stage	
completing	383
concept	46
generating inconsistent facets	419
GenerationSettings	
configuring generation XML file (control)	392
generic devices	
communicating through Ethernet explicit messaging	789
communicating through EtherNet/IP explicit messaging	795
communication through Modbus explicit messaging	799
connectivity with supervision	831
creating EtherNet/IP devices	531
creating Modbus TCP devices	531
managing Modbus TCP communication (I/O scanning)	781
mapping interfaces	603
genies	
editing supervision pages	495
Global Template editors	
opening	853
Global Templates	
actions	848
classification	25
copying and pasting	885

description.....	25, 847
duplicating.....	912
generating system documentation reports.....	933
Global Templates Explorer.....	839
managing	881
managing Control facet template passwords	849
naming convention.....	840
purging the Global Templates library.....	896
renaming.....	882
replacing templates of elements	907
saving changes.....	882
updating	901
updating at folder level	905
Global Templates definition	
concept	36
Global Templates Explorer	
folder actions	846
opening	101, 839
presentation	839
Toolbox	841
Global Templates library	
viewing in the Global Templates Explorer	839
Global Templates passwords	
resetting a forgotten Global Templates password.....	851
Go To Instance command	
navigating to the instance	382
graphical links	
generation rule.....	389
H	
HAL DFBs	
managing	639
HAL variables	
managing	639
HAL variables and DFBs	
properties	640
viewing.....	641
hardware	
configuration stage.....	498
creating the system topology.....	499
deploying hardware changes	685
mapping	602
supported components	499
updating hardware changes.....	750
hardware abstraction layer variables	
managing	639
hardware mapping	
managing HAL variables.....	639
Hardware Mapping Editor	
presentation	603
hardware mappings	
editing exported hardware mappings	612
exporting hardware mappings	610
importing hardware mappings	616
managing changes after deploying	608
unmapping hardware mappings	607
HART	
HART communication	536
HART modules	
configuring HART modules	536
representation	537
help	
help button	101
HTML help for user-created templates	96
opening the help of software Participants	93
using the HTML help for several versions of templates.....	94
Hide Deferred filter	876
Hide Unbound filter.....	876
Hide Unchecked filter.....	876
hierarchical naming	183
hot standby	
managing data backup files	286
hot standby controllers	
backing up controller data.....	733
connecting to Ethernet networks	541
deploying built control projects	667
deploying changes to redundant controllers	683
deploying controller data.....	677
deploying to redundant controllers	665, 667
disconnecting from Ethernet networks	542
execution	711
identifying.....	507
main deployment and execution scenarios	741
physical connection import rules.....	558
redeploying to redundant controllers	672
reservation and connection scenarios	744
HTML help	
HTML help for user-created templates	96
using the HTML help for several versions of templates.....	94
I	
icons	
assigning icons to folders.....	170
identifiers	
changing the identifier of Global Templates	882
naming rules.....	88
identifying engines	
using physical addresses	651
import	
accessing the import window	558
import command	
aborting (control).....	282
aborting (supervision).....	319
aborting (Topology)	561
aborting application import command	248
aborting device import command.....	571
aborting template import	894
import rules	
executable import rules (control)	275
executable import rules (supervision).....	316
for control project facets	274
for control project-related objects	272
for data backup files (control)	279
for equipment (supervision)	315
for included projects	313
for mapped peer to peer communication variables	277
for pages of supervision projects	315
for program sections (control)	273
for supervision page templates.....	315
for supervision project facets	315
for supervision project-related objects.....	314
for topological entities	557
for topological folders	557
mapping import rules (control)	275
mapping import rules (supervision)	316
physical connection import rules.....	558
import settings	
control projects	271
supervision projects	312
importing	
application import settings	250

checking consistency after importing control projects	271
color coding when importing application objects	250
color coding when importing devices from CSV ..	573
Control Expert projects	284
control projects	271
expressions for control project sections	271
importing application objects to XML or CSV	219
importing CSV files in the topology	571
importing devices from CSV file.....	564
importing hardware mappings	616
importing XML or CSV files into the application...	246
password related information for topological entities	558
resolving contents conflicts for templates	893
sections during online refinement	738
supervision projects	311
templates	889
topological entities	555
importing control projects	
software version compatibility	271
importing Control projects	
peer to peer communication.....	475
importing devices	
resolving conflicts	574
importing instances	
resolving conflicts	252
importing M580 safety controllers	772
importing safety control projects	763
importing supervision projects	
software version compatibility	311
included projects	
adding or removing	496
downloading	646
importing	313
importing with templates	889
saving (supervision)	619
inconsistent facets	
generating	419
layout of blocks after generation.....	420
infrastructure	
creating the system topology.....	499
supported components	499
instance count	
About dialog box	104
instance link status	197
instance path	
viewing instance location information.....	204
instance properties	
naming rules.....	88
instances	
accessing the Instance Editor	176
actions	173
changing the application hierarchy	206
color indicators	188
configuration parameters	184
configuring parameters.....	182
copying and pasting	207
counting for licensing.....	174
Create Instance dialog box	338
creating.....	171
creating in-place	329
data status	180
editing in-place	377
element selection changes of assigned facets....	367
exceeding count of a license	174
exporting and importing to XML or CSV	219
hierarchical naming	183
import rules for control project-related objects	272
import rules for supervision project-related objects	314
Inspect Instance window	204
managing	203
moving	209
navigating from facets to the instance	382
of templates.....	31
order when creating	171
resolving import conflicts	252
searching	161
searching application instances and folders	156
validating parameters	182
viewing facet assignments	379
instantiating	
templates	153
instantiation	
order when creating instances.....	171
structure of the application	154
instantiation stage	
completing.....	152
concept	37
interface links	
editing the XML export file.....	243
exporting and importing to XML or CSV	219
interface models	
types.....	30
interfaces	
changes log	855
editing/extending interfaces	917
generic device mapping interfaces	603
mapping hardware interfaces	602
modification strategy	899
invalid	
instance data validity status	180
invalid facets	
generating	387
generating invalid facets again	419
invalid link status	197
invariant culture	
editing the application export file	228
editing the device export file.....	568
editing the XML application export file.....	240
I/O device addresses	
Build stage	643
I/O devices	
communication mapping import rules.....	275
creating I/O devices	299
I/O devices for calculated variable tags	300
I/O devices for disk variable tags	300
properties (supervision)	299
I/O scanner	
configuring for controllers	512
disabling.....	511
I/O scanner lines	
memory allocation.....	632
I/O scanning	
connecting devices to Ethernet networks	544
connecting PRMs to Ethernet networks	546
creating and configuring Modbus TCP device.....	531
disconnecting devices from Ethernet networks	545
disconnecting PRMs from Ethernet networks	547
implementing communication.....	780
managing communication with devices	781
managing communication with PRMs	785
managing communication with STB islands.....	784
managing peer to peer communication	787
I/O server redundancy	
implementing	304

IL	using IL language.....	83
Instance Editor	accessing.....	176
	presentation	176
Interface Editor	pane description	855
	presentation	854
IO servers	communication ports	296
	properties	296
IP address	assigning to controllers.....	511
	assigning to EtherNet/IP devices.....	532
	assigning to Modbus TCP devices.....	532
	assigning to PRMs	528
	assigning to station nodes	521
	assigning to STB islands	523
	changing EtherNet/IP device properties	535
	changing for explicit messaging with devices....	793, 797
	changing Modbus TCP device properties	535
	changing STB islands properties	524
	deploying changes to redundant controllers	683
	online changes with redundant controllers.....	728
	physical connections to Ethernet networks	540
	refreshing engine status and connecting	657
IPSimPortAssign.xml	port simulation file location.....	464
italic formatting	editing instance properties	182
K		
keyboard shortcuts	using.....	121
L		
layout of function blocks		
after first generation	390	
when generating after changes	420	
when regenerating	422	
LD	using LD language	83
libraries	viewing in the Global Templates Explorer	839
licensing	counting instances	174
	exceeding license instance count	174
link validity status	197	
Linked Production Assets section	creating links to view instances	201
linking instances.....	193, 199	
	using the Asset Workspace Editor	193
links	creating links to view instance information in RTNS	201
	editing the XML export file.....	243
	exporting and importing to XML or CSV	219
	modifying links between instances.....	199
	rules for generating graphical links	389
Links Editor	Asset Workspace Editor	193
	Links Editor	943
	presentation	944
LL984	updating logical control projects	750
	using LL984 language	83
location	viewing instance location information.....	204
locked code	code lock feature.....	442
locked facets	skipping facets during generation	387
locking	Lock command (engineering client)	101
locking mechanisms	multiuser functionality.....	70
log files	activity log files	80
logging in	to server and clients	60
logging out	of server and clients	60
logical control projects	updating with online changes	748
loops	generating circular links	389
M		
M340	creating a control project	260
	creating controllers.....	507
	creating PRM PROFIBUS DPs.....	527
M580	creating a control project	260
	creating controllers.....	507
M580 safety systems	implementing	759
	maintenance mode	776
Manage Network Variables	saving changes in Manage Network Variables dialog box.....	461
managers	platform tools.....	35
managing	control projects	265
	supervision projects	306
managing network variables	M580 safety controllers	761
mapping	changing service mappings (control)	580
	clients (supervision)	622
	communication (control)	582
	communication (supervision)	624
	deleting mapping when updating topological templates.....	550
	editing exported hardware mappings	612
	exporting hardware mappings	610
	hardware	602
	importing hardware mappings	616
	I/O devices (supervision)	624
	mapping import rules (control).....	275
	mapping import rules (supervision).....	316
	mappings for safety control projects	774
	network variable mapping rules	598
	network variables	594, 612
	selecting variable mapping positions	596
	servers (supervision).....	622
	services (control)	580
	services (supervision).....	622
	to engines (control)	580
	topological mapping indicator.....	500
	Variable Mapping Editor window (control).....	594
	mapping stage	

completing.....	576
concept.....	51
control projects	577
exporting and importing hardware mappings	609
supervision projects	618
memory allocation	
when creating I/O scanner lines	632
menus	
common template editor menus	868
using menus	115
migrating databases	
code lock feature.....	442
Modbus explicit messaging	
communicating with devices	799
managing communication	798
Modbus serial	
managing communication	798
Modbus serial devices	
communication mapping.....	624
connectivity with supervision.....	831
Modbus TCP devices	
changing properties.....	535
communication mapping.....	624
connectivity with supervision	831
creating and configuring devices	531
exporting	552
import rules	557
importing	555
Modbus TCP explicit messaging	
communicating with devices	789
managing communication	788
Modbus TCP implicit messaging	
implementing communication.....	780
implementing communication with devices.....	781
implementing communication with PRMs	785
implementing communication with STB islands ..	784
implementing peer to peer communication	787
modules	
hardware module properties	514
monitoring	
Modbus device data from supervision.....	831
monitors	
floating Participant windows.....	116
using dual monitors	108
moving	
application folders	209
changing the position of folders and maintain hierarchy	211
facets	378
generated facets	433
instances.....	209
moving folders	166
moving sections (control).....	266
topological folders	549
multiple selection	
selecting multiple items	107
multiuser	
working with several clients.....	70

N

naming convention	
Global Templates naming convention	840
naming instances	
hierarchical naming	183
naming rules	
rules for identifiers and variable names	88
rules for Supervision data	88
navigating	

inside the Application Browser pane	163
inside the instance pane	179
navigating from facets to the instance	382
navigation services	
using after restoring systems	149
network addresses	
generating network addresses for supervision components	644
network interface cards	
actions	521
creating	520
properties	521
network redundancy	
implementing (supervision)	305
network variables	
address allocation during build	633
creating	461
description.....	452
management function rules	459
managing network variables	459
mapping	594, 612
mapping rules	598
modifying	462
properties	459
selecting variable mapping positions	596
Variable Mapping Editor window.....	594
networks	
conflicting Ethernet connections	541
connecting a station node to the Ethernet	543
connecting an STB island to the Ethernet.....	545
connecting controllers to Ethernet networks	542
connecting devices to Ethernet networks	544
connecting entities to an Ethernet network	540
connecting PRMs to Ethernet networks	546
creating an Ethernet network	538
disconnecting a station node from the Ethernet ..	544
disconnecting an STB island from the Ethernet..	546
disconnecting controllers from Ethernet networks.....	542
disconnecting devices from Ethernet networks ...	545
disconnecting PRMs from Ethernet networks	547
import rules	557
physical connection import rules.....	558
viewing Ethernet connections.....	540
NIC	
actions	521
assigning IP addresses	521
creating	520
properties	521
NOC	
configuring memory for I/O scanning	512
NOC communication modules	
configuring HART modules	536
non-FBD sections	
notification when regenerating	422
Not Built	
actions changing the build status (control).....	636
notification panel	
abort task icon	73
description.....	73
displaying	101
performance considerations.....	74

O

object models	
application templates	28
interface models	30
topological templates	29

object templates	layout of blocks after generation	420
classification	25	
description	25	
elements	25	
references	25	
types	27	
objects	object model overview	23
OFS	configuration file description	643
OFS configuration file	creating	642
deploying	693	
saving	619	
OFS services	actions for station nodes	522
creating in station node	517	
properties for station nodes	522	
OFSOPC	tag container properties	341
OFSOPC driver	communication board properties	302
OFSOPC/OPCUA configuration files	downloading	646
online changes	changing the deployed control project	727
main redundant configuration scenarios	741	
overview	723	
refining safety control projects online	776	
selecting IP addresses	728	
updating logical control projects	748	
updating projects after deploying changes	online	727
online help	HTML help for user-created templates	96
using the help for several versions of templates	94	
Online modification in RUN or STOP	505	
online refinement	importing sections	738
online refinement indicator	500	
OPC Factory Server	OFS configuration file description	643
OPC UA Server Expert	configuration file description	643
OPCUA	configuration file description	643
tag container properties	341	
variable tag addresses	341	
OPCUA driver	citect.ini parameters	302
communication board properties	302	
OPCUA embedded	mapping communication	624
operation clients	using navigations services after restoring	
systems	149	
operation servers	creating in the topology	516
operator stations	creating in the topology	516
order	changing facet order inside section	381
changing FBD section order	266	
order of sections	notification when regenerating non-FBD sections	422
Out Of Date	actions changing the build status (control)	636
Out Of Date facets		
owner	about network variables	452
managing network variables	459	
owner and consumer projects	452, 459	
owner project	description	452
P		
page templates	selecting and managing page templates	346
supervision page template import rules	315	
supervision project settings	292	
pages	actions	345
assigning genie facets	495	
creating supervision pages	345	
editing	495	
properties	345	
supervision page import rules	315	
panes	common template editor panes	874
working with panes	111	
panning	moving the workspace around	106
parameter pane	in template editors	874
parameters	accessing instance properties	176
conditional font formats	182	
configuring instance parameters	182	
hardware module properties	514	
instance configuration	184	
instance parameter filters	180	
naming rules	88	
regional options for instance parameter	182	
validating instance parameters	182	
Participant projects	creating	383
deploying changes to station nodes		
(supervision)	703	
deploying controller data to engines (control)	677	
deploying from station node (supervision)	698	
deploying to engines (control)	667	
deploying to redundant controllers	665	
deploying to station nodes from executable		
(supervision)	695	
executing with redundant controllers	711	
managing data backup files (control)	286	
redeploying to engines (control)	672	
redeploying to engines from executable		
(supervision)	696, 702	
system root folder actions	259	
Participant windows	troubleshooting floating window appearance	117
Participants		
floating Participant windows	116	
opening the Participant help	93	
Participant window description	116	
password protection		
controller properties	509	
station node properties	520	
system properties	137	
passwords	actions that require providing the password	659
changing passwords of project loaded at simulator startup	663	

changing the application password of control projects	662
changing the password for Global Templates	850
changing the password of systems	139
clearing the application password of control projects	662
clearing the password for Global Templates	851
clearing the password of systems	140
exporting password related information for topological entities	552
Global Template settings	849
importing password related information for topological entities	558
managing Control facet template passwords	849
managing passwords of control projects	658
managing system access passwords	138
password information and settings in system backups	143
password information and settings when restoring systems	147
project loaded at simulator startup	661
resetting a forgotten controller password	663
resetting a forgotten Global Templates password	851
resetting a forgotten safety password	772
resetting a forgotten system password	140
setting the application password of control projects	661
verifying password to perform operations	662
paste command	
aborting	207
pasting	
Global Templates and folders	885
pasting instances in Control projects	335
peer to peer	
communication mapping import rules	275
mapping network variables	594, 612
peer to peer communication export settings	269
peer to peer communication variable import rules	277
peer to peer communication	475
about network variables	452
address allocation during build	633
client communication channel properties	586
communication mapping (control)	582
configuring the I/O scanner of controllers	512
controller memory size configuration	509
creating network variables	461
examples	476
exporting control projects	267
exporting Control projects	475
implementing	787
importing Control projects	475
managing network variables	459
modifying network variables	462
network variable properties	459
owner and consumer projects	452
server communication channel properties	588
templates	452, 459
variable properties	456
peer to peer communication channels	
actions	585
adjusting properties	591
creating	590
moving	592
performance	
notification panel	74
OFSOPC driver configuration	302
physical connections	
physical connection import rules	558
P&ID	
instantiation stage	152
pin button	
working with panes	111
platform components	
platform tools	35
platforms	
supported hardware platforms	83
PMEPXM0100 modules	
integration workflow	836
port simulation file location	464
ports	
creating additional communication ports (supervision)	303
creating communication ports (supervision)	302
IO server communication ports	296
supervision server communication ports	296
positions	
selecting variable mapping positions	596
printing	
template layouts	870
PRM	
actions	527
assigning IP addresses	528
changing the configuration	528
changing the PRM configuration	529
configuring	528
connecting to Ethernet networks	546
creating PRM PROFIBUS DPs	527
creating PRM PROFIBUS Masters	527
deleting	530
disconnecting from Ethernet networks	547
import rules	557
managing Modbus TCP communication (I/O scanning)	785
properties	528
representation	529
PROFIBUS	
connecting to Ethernet networks	546
creating PRM PROFIBUS DPs	527
creating PRM PROFIBUS Masters	527
disconnecting from Ethernet networks	547
managing Modbus TCP communication (I/O scanning)	785
PROFIBUS remote master and DPs	
exporting	552
importing	555
PROFIBUS secondary devices	
integrating in systems	836
programming languages	
supported languages	83
progress indicators	
instance color icons	188
project color icons	321
Project Browser	
system root folder actions	259
project definition stage	
completing	254
concept	41
control projects	259
Project Browser	255
Project Explorer	255
supervision projects	288
Project Explorer	
description	256
opening	101
project files	
downloading (supervision)	646

project settings	implementing alarm server redundancy	305
analyzing M580 safety projects and configurations	770	
defining (supervision)	495	
setting during build (control)	632	
projects	redundant controllers	
color indicators	321	
generating	414	
generating after changes	416	
generating system documentation reports	929	
types of Participant projects	42	
properties	backing up controller data	733
accessing instance properties	176	
adjusting peer to peer communication channel properties	591	
application folders	166	
changing EtherNet/IP device properties	535	
changing Modbus TCP device properties	535	
changing STB islands properties	524	
configuring instance parameters	182	
hardware module properties	514	
I/O devices (supervision)	299	
IO server properties	296	
properties of sections	336	
supervision client properties	298	
supervision server properties	296	
systems	137	
Protocol	redundant I/O servers	
tag container attributes	341	
Purge command	implementing I/O server redundancy	304
aborting template purge	897	
purging	redundant networks	
Global Templates library	896	
Q	implementing network redundancy (supervision)	305
Quantum	redundant report servers	
creating a control project	260	
creating controllers	507	
creating PRM PROFIBUS DPs	527	
quick reference	redundant trend servers	
system engineering life cycle task workflow	124	
system engineering life cycle tasks	128	
quitting	implementing trend server redundancy	305
clients	68	
quitting the software	68	
R	references	
radar view	definition	25
description	877	
READ_REMOTE function	missing references in Recovery folder	844
configuration		
467	refine online	
reassigning	main redundant configuration scenarios	741
facets	723	
reconnect button	with redundant controllers	711
connecting to engines		
refreshing engine status	657	
Recovery	refinement	
Recovery folder description	844	
redeploying	consistency management (control)	449
main redundant configuration scenarios	741	
to engines (control)	672	
to engines from executable (supervision)	696, 702	
to redundant controllers	672	
redundancy	refining locked code	442
implementing (supervision)	304	
redundant alarm servers	refinement stage	
	completing	434
	concept	47
	control projects	435
	supervision project	493
	refining	
	advanced supervision project settings	495
	control projects	436
	editing supervision pages	495
	managing device DDT variables	445
	refining safety control projects	767
	refining safety control projects online	776
	supervision projects	494
	validity indicators	439
	refining safety control projects online	776
	R/W Rights of Referenced Variable	
	network variables	459
	ReGenerate command	
	aborting regeneration of projects and sections	422
	regenerating	
	configurable settings (control)	392
	control project sections	422
	control projects	421
	entire control projects	422
	FBD section size	422
	function block layout after changes	422
	functionality description	421
	importing sections after regeneration	416
	removing types	421
	regenerating non-FBD sections	
	notification when regenerating	422
	regional options	
	instance parameters	182
	release notes	
	viewing platform and library release notes	104
	relinking	
	facets	433
	removing	
	removing generated types	391
	removing types when regenerating	421
	renaming	

Global Templates	882
Replace dialog box	
description.....	916
Replace References dialog box	899
Replace Template	
command description	212
replacing	
replacing templates of elements	907
template of instances	212
templates with new type version	216
Replacing references in parent templates.....	899
replacing templates	
validating instance identifier uniqueness	217
report server redundancy	
implementing	305
reports	
application data	929
generating system documentation reports.....	927
Global Templates data.....	933
project data	929
topological data	931
reservation	
redundant configuration reservation scenarios ...	744
Resolve Uniqueness Conflicts	
dialog box description.....	252
dialog box description for device import	574
restarting	
restarting clients that closed unexpectedly	67
restore	
impact on system names in database	149
password information and settings when restoring	
systems.....	147
restore command	
aborting.....	148
restoring	
deployed Participant project from executable	
(supervision).....	696
deployed Participant project from station node	
(supervision).....	701
restoring systems.....	147
rollback	
rolling back generation	389
RTNS	
creating links to view instance information.....	201
rules	
truncation rules for equipment (supervision)	424
run	
starting controllers and simulators (control)	713
runtime navigation services	
creating links to view instance information.....	201
using after restoring systems	149
S	
S88	
assigning icons to folders.....	170
modeling the equipment hierarchy.....	154
S908	
building changes.....	638
using CCOTF with S908 architectures	505
safe area password	
managing the safety password	770
safe signature	
building safety control projects	774
deploying to safety controllers	775
safety controllers	
configuring M580 safety controllers	769
implementing M580 safety systems	759
safety password	
managing the safety password.....	770
safety passwords	
resetting a forgotten safety password	772
saving	
backing up controller data.....	733
changes in Manage Network Variables dialog	
box.....	461
Global Templates	882
saving templates	
tracking changes in the changes log	882
sbk file	
exporting templates.....	886
searching	
application instances and folders.....	156
instances.....	161
template search	158
section layout	
configurable settings (control)	392
section size	
configurable settings (control)	392
sections	
actions	335
changing execution order	266
creating and managing	329
creating FBD and non-FBD sections	264
creating FBD sections	335
creating sections in safety control projects	766
deleting FBD sections during update Control	
project	754
import rules for program sections	273
importing during online refinement.....	738
inserting FBD sections.....	335
moving sections (control).....	266
properties	336
tree view description in Containers pane	371
tree view when refining Control projects	438
tree view when refining Control projects online	728
security services	
system properties	137
selecting	
selecting multiple items	107
server	
connection to clients.....	64
starting again.....	63
stopping	68
servers	
creating servers (supervision)	296
IO server properties	296
supervision server communication ports	296
supervision server properties	296
service mapping	
changing (control)	580
services	
generating network addresses	644
session overcrowded	
troubleshooting	427
sessions	
restarting clients that closed unexpectedly	67
settings	
configuring generation settings (control)	392
Settings	
control project settings.....	263
Global Template settings	849
modifying the template folder layout	843
supervision project settings.....	292
Settings pane	
changing application folder icons.....	170
Settings window	
Systems Explorer.....	136

SFC	
using SFC language.....	83
shortcuts	
keyboard shortcuts.....	121
signature mismatch	
actions that require providing the password.....	659
controller data backup	736
controller data deployment.....	680
deployment stage	682
execution stage	709
simplified system engineering life cycle	
graphical representation	57
simplified view	
Asset Workspace Editor	196
template editors	878
simulator	
backing up controller data.....	733
changing passwords of projects loaded at startup.....	663
changing the default project loaded at startup.....	655
deploying built control projects	667
deploying changes (control).....	682
deploying controller data (control).....	677
managing data backup files (control)	286
mapping hardware	602
obtaining the IP address	656
redeploying built control projects	672
setting passwords of project loaded at simulator startup.....	661
starting.....	654
starting without loading a password-protected project	655
simulator ports	654
simulator software	
deploying to.....	654
installing.....	654
starting and stopping	713
skipping	
skipping facets during generation	387
software Participants	
floating Participant windows.....	116
window description.....	116
sorting	
sorting data	118
ST	
using ST language	83
STA	
changing the default project loaded at startup.....	655
starting the simulator without password protection	655
stages	
assignment stage concept	44
assignment stage implementation	328
build stage concept	53
build stage implementation	628
concept	32
configuration stage concept	49
configuration stage implementation	498
deployment stage concept	55
deployment stage implementation	648
execution stage concept	56
execution stage implementation	706
generation stage concept	46
generation stage implementation	383
Global Templates definition stage concept	36
instantiation stage concept	37
instantiation stage implementation	152
mapping stage concept	51
mapping stage implementation	576
project definition stage concept	41
project definition stage implementation	254
refinement stage concept	47
refinement stage implementation.....	434
task quick reference	128
starting	
Controller simulator	654
controllers and simulators (control).....	713
simulator	654
system server	63
station nodes	
actions	517
backing up controller data	733
connecting to Ethernet networks	543
creating	516
disconnecting from Ethernet networks	544
exporting	552
import rules	557
importing	555
mapping to emulated simulators (control).....	580
properties	520
selecting templates	504
status	
editing instance properties	182
instance data validity status	180
refreshing engine status	657
status of supervision executables	
status change when importing new content version	893
STB islands	
actions	523
assigning IP addresses	523
changing properties.....	524
changing the configuration.....	524
configuring	524
connecting to Ethernet networks	545
creating	523
deleting	526
disconnecting from Ethernet networks	546
exporting	552
importing	555
managing Modbus TCP communication (I/O scanning).....	784
mapping (control).....	580
mapping hardware	602
properties	523
representation	525
stop	
changes requiring to stop the controller.....	692
stopping	
controllers and simulators (control).....	713
system server	68
structure of the application	
instantiation stage	154
STU files	
changing (control)	636
creating (control).....	630
supervision	
connectivity with Modbus devices	831
supervision with M580 safety systems	777
supervision architecture	
description of tasks	289
supervision clients	
properties	298
supervision data	
naming rules.....	88
supervision pages	
actions	345
creating pages	345

properties	345
Supervision Project Browser	
description.....	258
supervision projects	
actions	290
building	642
changing built Participant projects	647
communication mapping import rules.....	316
creating.....	290
creating client	297
creating components	294
creating I/O devices	299
creating servers	296
deploying	693
equipment import rules	315
executable import rules	316
executing	757
export settings	309
exporting	307
facet import rules	315
generating after changes	417
hardware mapping import rules	316
implementing redundancy	304
import settings	312
importing	311
managing	306
mapping stage	618
page import rules	315
page template import rules.....	315
project definition.....	288
Project Explorer and Browser	255
project import window	317
properties	292
refining	493
related project data import rules	314
restoring after deploying from executable.....	696
restoring after deploying from station node.....	701
service mapping import rules	316
task quick reference	128
task workflow	124
supervision servers	
properties	296
Supervision service	
creating in station node.....	517
supervision services	
actions for station nodes	522
properties for station nodes.....	522
support	
support contact details.....	104
symbol table files	
generating symbol table files.....	652
synchronization	
timeout when refining online	712
synchronizing	
description.....	711
redundant controllers	719
timeout.....	719
when deploying changes (control)	666
syslog server	
audit trail for user activities.....	76
system	
description.....	33
System Access Password	
managing system access passwords.....	138
system access passwords	
changing the password of systems	139
clearing the password of systems.....	140
resetting a forgotten system password	140
system backup	
password information and settings	143
system engineering life cycle	
concept	32
graphical representation	33
platform components.....	35
simplified workflow	57
task quick reference	128
task workflow	124
system folders	
actions	133
creating	133
system names	
impact of system restoration	149
system restore	
password information and settings	147
system server	
logging in and out.....	60
systems	
actions	135
backing up.....	142
creating	135
description.....	23
engineering	122
managing system access passwords.....	138
organizing	133
properties	137
restoring.....	147
scheduling system backups	145
Systems Explorer	
opening	101
presentation	133

T

tabs bar	
description.....	105
tag containers	
actions	341
creating tag containers	340
properties	341
template passwords	
changing the password for Global Templates	850
clearing the password for Global Templates	851
templates	
copying and pasting Global Templates	885
creating Global Template folders	845
description.....	25
duplicating Global Templates	912
exporting	886
Global Template editors	853
Global Templates actions	848
Global Templates description	847
Global Templates Explorer	839
HTML help for user-created templates	96
import rules for control project-related objects	272
import rules for supervision project-related	
objects	314
importing	889
instances of templates	31
instantiating	153
managing	881
modification strategy	899
naming convention	840
purging the Global Templates library	896
renaming Global Templates	882
replacing templates of elements	907
saving changes in Global Templates	882
selecting	158
selecting topological templates	504

templates Import Editor	889
terminology	26
updating Global Templates	901
updating templates at folder level	905
using the HTML help for several versions of templates.....	94
Templates Browser	
browsing templates	158
templates of instances	
replacing	212
updating	212
templates of the topology	
updating	550
temporary files	
content repository	940
terminology	
for template components	26
terminology used in this manual	16
timeout	
synchronizing standby controllers.....	719
when synchronizing online changes	712
toolbars	
common template editor toolbars	868
description.....	101
engineering client UI description.....	101
Toolbox	
Global Templates Explorer.....	841
topological entities	
import rules	557
physical connection import rules.....	558
topological entity identifiers	
naming rules.....	88
topological entity nodes	
description.....	500
topological folders	
import rules	557
topological templates	
introduction	29
subtypes	29
topology	
configuration stage.....	498
creating	499
exporting	552
generating system documentation reports.....	931
import rules for control project-related objects	272
import rules for supervision project-related objects	314
importing	555
managing	548
selecting topological templates	504
supported components	499
updating templates.....	550
Topology Explorer	
opening.....	101
presentation	500
topology folders	
actions	501
creating	501
properties	503
topology objects	
exporting and importing to CSV.....	564
tracking changes	
template changes log	882
transferring	
deploying changes to station nodes (supervision).....	703
deploying from station node (supervision)	698
deploying to station nodes from executable (supervision).....	695
redeploying to engines from executable (supervision)	696, 702
synchronizing redundant controllers	719
tree view	
viewing the application structure in control projects	333
tree view in control projects	
description.....	371
when refining	438
when refining online	728
trend server redundancy	
implementing	305
troubleshooting	
floating Participant window appearance	117
generation diagnostic messages	427
truncation	
rules for equipment (supervision)	424
type versions	
changing type versions during generation (control).....	396
replacing templates with new type version	216
updating templates with new type version	216
types	
removing generated types	391
removing when regenerating.....	421

U

unassigned facets	
reassigning.....	432
unassigning	
facets	378
generated facets	432
unlinked facets	
relinking	433
unlinking	
facets	432
unmap	
unmapping hardware mappings	607
unmapping	
unmapping communication channels	591
Update Control Project command	
aborting.....	756
Update References dialog box	899
Update Template	
command description	212
updating	
control projects after deploying changes online ..	727
controller configurations	748
Global Templates	901
Global Templates at folder level.....	905
linked application templates	216
logical control projects	748
template modification strategy.....	899
template of instances	212
templates of elements	901
templates with new type version	216
topological templates	550
updating control projects	
checking consistency	754
deleting FBD sections during update Control project	754
updating safety control projects	776
with redundant controllers.....	711
updating parent templates	
editing templates.....	883
Updating references in parent templates	899
user authentication	
OPC UA configuration files	643

User Contents folder	selecting items.....	107
location in the content repository	936	
user files	working with panes.....	111
redeploying from executable after removing user files (supervision)	696, 702	
user interface	WRITE_REMOTE function configuration.....	467
engineering client.....	100	
engineering client UI description.....	101	
engineering client work areas.....	105	
user-created templates		
HTML help for user-created templates	96	
users		
activity log files	80	
logging in to server and clients	60	
working with several clients.....	70	
V		
Validity Diagnosis window	197	
validity indicators		
analyzing control projects during generation.....	388	
description.....	632	
instance color icons.....	188	
project color icons	321	
refining control Participant projects.....	439	
Variable Mapping Editor		
description.....	594	
usage example	597	
variable mappings		
peer to peer communication variable import rules	277	
variable name conflict		
troubleshooting	427	
variable tag addresses		
OPCUA embedded protocol.....	643	
variables		
about network variables.....	452	
address allocation during build	633	
managing network variables	459	
naming rules.....	88	
properties	456	
when updating Control projects	751	
version conflict		
canceling generation for version conflict (control).....	389	
troubleshooting	427	
versions		
changing type versions during generation (control).....	396	
W		
windows		
floating Participant windows.....	116	
Participant window description	116	
viewing on separate monitors.....	108	
work areas		
creating additional work areas.....	108	
creating work areas in explorers.....	109	
description.....	105	
organizing	108	
workflow		
task workflow to implement communication.....	124	
workspace		
changing the zoom level	107	
zooming in and out.....	107	
workspaces		
moving the workspace around.....	106	

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time,
please ask for confirmation of the information given in this publication.