



Process Expert - General Purpose Library

User Guide

EIO0000004045.05
03/2023



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.

© 2023 – Schneider Electric. All rights reserved.

Table of Contents

Safety Information.....	5
Qualification of Personnel	5
Proper Use.....	6
Before You Begin.....	6
Start-up and Test	7
Operation and Adjustments	7
About the Book.....	9
Overview	12
Introduction.....	12
Template.....	13
Library Structure.....	13
Template Structure	13
Optional Services	22
Alternate Services	23
Project Standards.....	24
Control	27
Owner/Setpoint	27
Modes	29
Output	30
Confirmation Failure	31
Inoperable Asset	32
Interlocks	33
Restart.....	35
Maintenance	36
Supervision	38
Situational Awareness	38
Analog Values	39
Control Modules	43
Workspace.....	44
Genie	58
Faceplates	64
Security	76
Alarms	78
Data Logging.....	80
Items	82
Configure and Run a Supervision Project	83
Customization.....	86
Supervision.....	86
Customizing Templates	98
Index	115

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.

Qualification of Personnel

A qualified person is one who has the following qualifications:

- Skills and knowledge related to the construction and operation of electrical equipment and the installation.
- Knowledge and experience in industrial control programming.
- Received safety-related training to recognize and avoid the hazards involved.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical,

electrical, or electronic equipment. The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

Proper Use

This product is a library to be used together with the automation control systems and is intended solely for the purposes described in the present documentation as applied in the industrial sector.

Always observe the applicable safety-related instructions, the specified conditions, and the technical data.

Perform a risk evaluation concerning the specific use before using the product. Take protective measures according to the result.

Since the product is used as a part of an overall system, you must ensure the safety of the personnel by means of the concept of this overall system (for example, machine concept).

Any other use is not intended and may be hazardous.

Before You Begin

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

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

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before

placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

Start-up and Test

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check are made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

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

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

Operation and Adjustments

The following precautions are from the NEMA Standards Publication ICS 7.1-1995:

(In case of divergence or contradiction between any translation and the English original, the original text in the English language will prevail.)

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.

- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

Document Scope

This document describes the concepts and customization of the template, control, and supervision.

Validity Note

This document has been updated for the release of EcoStruxure™ Process Expert 2021.

Related Documents

Title of Documentation	Reference Number
EcoStruxure™ Process Expert - General Purpose Library Process Templates Reference Manual	EIO0000004043
EcoStruxure™ Process Expert - General Purpose Library Device Templates Reference Manual	EIO0000004044

You can download these technical publications and other technical information from our website at www.se.com/ww/en/download/.

Technical Support

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

Product Related Information

⚠ WARNING
LOSS OF CONTROL <ul style="list-style-type: none">• Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, 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 them.• Review the implications of communication link interruptions and take actions to mitigate them.• Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, 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.

Examples described in this manual are provided 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.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as safety, safety function, safe state, fault, fault reset, malfunction, failure, error, error message, dangerous, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2015	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction.
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements.
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection.
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design.
IEC 62061:2015	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems.
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term zone of operation may be used in conjunction with the description of specific hazards, and is defined as it is for a hazard zone or danger zone in the Machinery Directive (2006/42/EC) and ISO 12100:2010.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Overview

Introduction

EcoStruxure™ Process Expert consists of the following two core libraries for the engineer to develop their projects:

Foundation library contains the elements required to link the process application with the physical hardware. These libraries contain the signal conditioning facets which are used in other libraries.

General Purpose Library contains the elements required to write the automation logic for the process control system.

The library contains:

- Signals (Analog, Digital, User Input)
- Calculations
- Motor
- Valves
- Process Control
- Sequence Control
- Equipment Control

The library also provides access to devices which form the control system:

- Motor Starters
- Soft Starters
- Variables Speed Drives
- Circuit Breakers
- Motor Protection Relays
- Power Meters

The library enables communication with the devices through different protocols:

- Modbus Serial
- Modbus TCP
- Ethernet IP
- Profibus

The control system is engineered by the configuration of these templates, linking of the templates together to form control loops and refinement of the generated control system configuration.

Template

Overview

This chapter describes the basic concepts of the process and device templates.

Library Structure

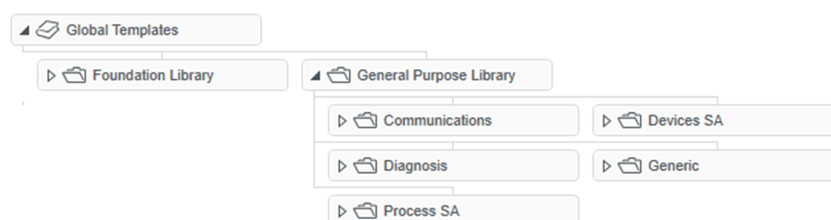
Overview

The general purpose library contains the following sections:

- Process - Modules for Process Automation
- Devices - Instrumentation, Motor and Power Management
- Communication - Fieldbus Communication
- Generic - Hyperlink services
- Diagnostics - Controller diagnostic information

The Process and Devices templates are all available with the installation and can be restored as required.

The following figure depicts the structure of the library templates:



This document is focused on the SA (Situational Awareness) version of the general-purpose library for EcoStruxure Process Expert for AVEVA System Platform. If it is installed the folders, Process SA and Devices SA are available within the general-purpose library folder.

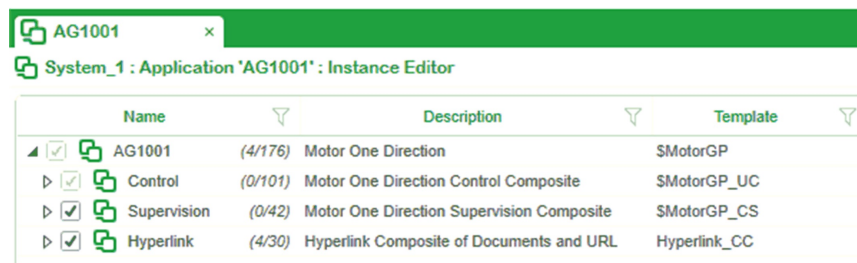
Template Structure

Overview

EcoStruxure Process Expert supports multiple participants. To enable the user to select to include individual participants, the template is divided per participant:

- **Control:** Defines the logic of the controller.
- **Supervision:** Defines the supervision configuration (optional - default enabled).
- **Hyperlinks:** Defines the data available with Runtime Navigation Services.

The following figure depicts the template structure of the \$MotorGP:



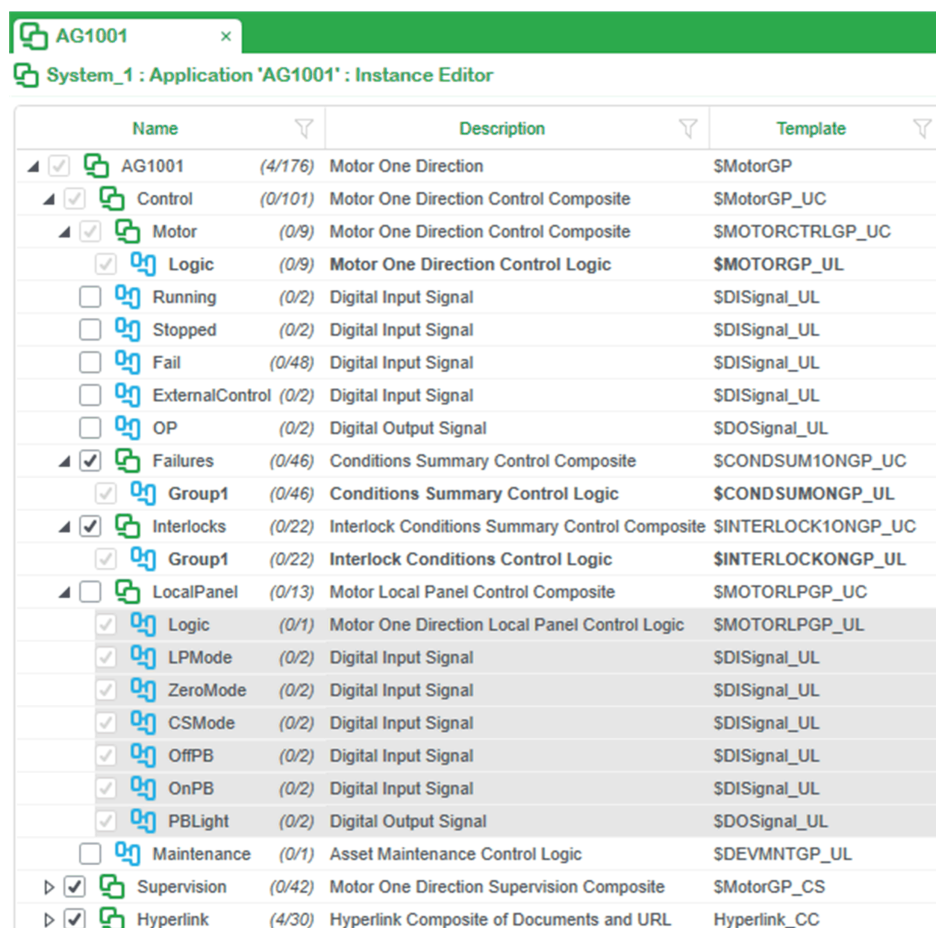
Name	Description	Template
AG1001 (4/176)	Motor One Direction	\$MotorGP
Control (0/101)	Motor One Direction Control Composite	\$MotorGP_UC
Supervision (0/42)	Motor One Direction Supervision Composite	\$MotorGP_CS
Hyperlink (4/30)	Hyperlink Composite of Documents and URL	Hyperlink_CC

Each participant contains facet templates which consist of a set of mandatory or optional services. You can configure the template to enable or disable optional services. Disabling a service removes the configuration from each participant to optimize the runtime operation of the control system:

- **Failure** (optional) - default enabled.
- Interlocks (optional) - default enabled.
- Local Panel (optional) - default disabled.
- Maintenance (optional) - default disabled.

Selecting an optional service in the control facet enables it across all enabled participants. You must have a matching facet in each participant however some facets are specific to a single participant and hence do not appear.

The following figure shows an instance of the \$MotorGP structure:



Name	Description	Template
AG1001 (4/176)	Motor One Direction	\$MotorGP
Control (0/101)	Motor One Direction Control Composite	\$MotorGP_UC
Motor (0/9)	Motor One Direction Control Composite	\$MOTORCTRLGP_UC
Logic (0/9)	Motor One Direction Control Logic	\$MOTORGP_UL
Running (0/2)	Digital Input Signal	\$DSignal_UL
Stopped (0/2)	Digital Input Signal	\$DSignal_UL
Fail (0/48)	Digital Input Signal	\$DSignal_UL
ExternalControl (0/2)	Digital Input Signal	\$DSignal_UL
OP (0/2)	Digital Output Signal	\$DSignal_UL
Failures (0/46)	Conditions Summary Control Composite	\$CONDSUM1ONGP_UC
Group1 (0/46)	Conditions Summary Control Logic	\$CONDSUMONGP_UL
Interlocks (0/22)	Interlock Conditions Summary Control Composite	\$INTERLOCK1ONGP_UC
Group1 (0/22)	Interlock Conditions Control Logic	\$INTERLOCKONGP_UL
LocalPanel (0/13)	Motor Local Panel Control Composite	\$MOTORLPGP_UC
Logic (0/1)	Motor One Direction Local Panel Control Logic	\$MOTORLPGP_UL
LPMODE (0/2)	Digital Input Signal	\$DSignal_UL
ZeroMode (0/2)	Digital Input Signal	\$DSignal_UL
CSMODE (0/2)	Digital Input Signal	\$DSignal_UL
OffPB (0/2)	Digital Input Signal	\$DSignal_UL
OnPB (0/2)	Digital Input Signal	\$DSignal_UL
PBLight (0/2)	Digital Output Signal	\$DSignal_UL
Maintenance (0/1)	Asset Maintenance Control Logic	\$DEVMTGP_UL
Supervision (0/42)	Motor One Direction Supervision Composite	\$MotorGP_CS
Hyperlink (4/30)	Hyperlink Composite of Documents and URL	Hyperlink_CC

All library templates and facets provided by Schneider Electric start with a \$ symbol. This is to avoid any overlap of the templates created by other users with library templates as the libraries and projects evolve over time.

The templates of the general purpose library with situational awareness are suffixed with GP.

Control Modules

Library templates are stored in the folder Control Modules. These control modules consist of control *composite* (*_UC*) and Supervision *composite* (*_CS*). Supervision composite is optional for the template.

Control Composites (*_UC*)

The *Control* composite defines the composition of the control logic for an asset. It contains multiple control composites or control logic facets and the interfaces which connect them. The layout of this facet is essential as it impacts the layout of the code generated within the controller.

Control Composites are named based on the core function blocks which they contain (*\$AALARMGP*) or the template in which they are located (*\$AnalogInputGP*). They are stored in the folder Control under *Control Services*.

The following figure shows the control composite:

Process SA

Control Services

Interfaces

Control

Identifier	Version	Type	Subtype	Valid	State	Description
\$AALARMGP_UC	1.0.15	Application Composite Template	True	Approved	Analog Alarms Control Composite	
\$ACalcGP_UC	1.0.8	Application Composite Template	True	Approved	Analog Calculations Control Composite	
\$AnalogGP_UC	1.0.10	Application Composite Template	True	Approved	Analog Indicator with Configurable Range with Signal Conditioning Control Composite	
\$ANPUTGP_UC	1.0.11	Application Composite Template	True	Approved	Analog Indicator with Configurable Range with Signal Conditioning Control Composite	
\$ANPUTGP_UC	1.0.4	Application Composite Template	True	Approved	Analog Indicator with Configurable Range with Signal Conditioning Control Composite	
\$AlarmSummaryGP_UC	1.0.17	Application Composite Template	True	Approved	Alarm Summary Control Composite	
\$AnalogInputGP_UC	1.0.24	Application Composite Template	True	Approved	Analog Indicator with Configurable Range with Signal Conditioning and Analog Alarms-Unity Control	
\$AnalogLinearGP_UC	1.0.7	Application Composite Template	True	Approved	Analog Signal Linearization Control Composite	
\$AnalogOutputGP_UC	1.0.37	Application Composite Template	True	Approved	Analog Output Control Composite	
\$AOUTPUTGP_UC	1.0.19	Application Composite Template	True	Approved	Analog Output Control Composite	

Control Logic (*_UL*)

Each function block (or set of blocks - to support scalability) resides within a single control logic facet. Control Logic facets can be parameterized to define the operation of the individual control block. A control composite exists for each control logic facet.

Control Logic facets are named based on the control function blocks which they contain and are stored in the folder *Control Logic* under *Control Services*.

The following figure shows the control logic facets:

Process SA

Control Services

Interfaces

Control

Control Logic

Identifier	Version	Type	Subtype	Valid	State	Description
\$ACalcGP_UL	1.0.7	Application Logic		True	Approved	Analog Calculations Control Logic
\$AnalogGP_UL	1.0.8	Application Logic		True	Approved	Configurable Analog Input and Calculation Control Logic
\$ANPUTGP_UL	1.0.5	Application Logic		True	Approved	Configurable Analog Input and Calculation Control Logic
\$ASignal_UL	0.1.0	Application Logic		True	Approved	Analog Input Signal Conditioning
\$ASignalGP_UL	1.0.6	Application Logic		True	Approved	Analog Signal Linearization Control Logic
\$AOUTPUTGP_UL	1.0.15	Application Logic		True	Approved	Analog Output Control Logic
\$AOUTPUTGP_UL	1.0.10	Application Logic		True	Approved	Analog Output Local Panel Control Logic
\$ASelectGP_UL	1.0.17	Application Logic		True	Approved	Selector for Analog Signals with Monitoring Variables Control Logic
\$CONDSUMONGP_UL	1.0.2	Application Logic		True	Approved	Conditions Summary Control Logic
\$CONDSUMONGP_UL	1.0.4	Application Logic		True	Approved	Conditions Summary Control Logic

Supervision

The supervision Participant is optional. The supervision composite contains the database and graphic elements which are required for the asset.

Supervision Composites (*_CS*)

Supervision composites contain the following core supervision elements:

- Global parameters
- Data elements
- Graphic elements

Supervision composites are stored in the *Supervision Services* folder.

Process SA

Control Services

Interfaces

Supervision Services

Supervision

Identifier	Version	Type	Subtype	Valid	State	Description
SAAlarmSummaryGP_CS	1.0.31	Application Composite Template	True	Approved	Alarm Summary Citect Data	
SAAnalogInputGP_CS	1.0.60	Application Composite Template	True	Approved	Analog Indicator with Configurable Supervision Composite	
SAAnalogOutputGP_CS	1.0.40	Application Composite Template	True	Approved	Analog Output Supervision Composite	
SAAnalogSelectGP_CS	1.0.30	Application Composite Template	True	Approved	Analog Signal Selector with Monitoring Variables Supervision Composite	
SCControlValveGP_CS	1.0.30	Application Composite Template	True	Approved	Control Valve Supervision Composite	
SDigitalInputGP_CS	1.0.33	Application Composite Template	True	Approved	Digital Input Supervision Composite	
SDigitalOutputGP_CS	1.0.32	Application Composite Template	True	Approved	Digital Output Supervision Composite	
SDualOPValveGP_CS	1.0.9	Application Composite Template	True	Approved	DValve Output Supervision Composite	
SHandValveGP_CS	1.0.26	Application Composite Template	True	Approved	Hand Valve Supervision Composite	
SMessageBoxGP_CS	1.0.16	Application Composite Template	True	Approved	Message Indicator to the Operator Supervision Composite	

Supervision parameters facets contain the supervision parameters which are common across the asset and not specific to an optional service. They also contain the equipment element.

- Asset Properties - properties used to filter data
- Display - screens to display when the asset is selected
- Security - required roles to implement functionality
- Alarm - Severities
- Trends - Historical data settings

These parameters are passed to the appropriate data and graphic elements.

The following figure shows the instance engineering of the Supervision (\$MotorGP):

AG1001
 System_1 : Application 'AG1001' : Instance Editor

Name							
AG1001 (4/176)							
Control (0/101)							
Supervision (0/42)							
Settings (0/22)							
Data (0/20)							
Motor (0/12)							
Failures (0/1)							
Group1 (0/1)							
Interlocks (0/1)							
Group1 (0/1)							
LocalPanel (0/2)							
Maintenance (0/4)							
Hyperlink (4/30)							

☒ Hierarchical Name

Parameter Filters ▼
☒ Editable ☐ Modified ☐ Child Parameters

Grouped By: Element Path Category

	Name	Description	Type	Value
Supervision Settings				
Operation				
	OwnerNormal	Normal Owner	Enum	Program & Cascade (5) ▼
Security				
	SecurityOwner	Set Owner	Enum	Operator (1) ▼
	SecuritySetpoint	Set Setpoint	Enum	Operator (1) ▼
	SecurityModeOverride	Set Mode - Override	Enum	Operator (Confirmed) (10) ▼
	SecurityModeSimulation	Set Mode - Simulation	Enum	Operator (Confirmed) (10) ▼
	SecurityModeMaintenance	Set Mode - Maintenance	Enum	Maintenance (3) ▼
	SecurityModeOutService	Set Mode - Out of Service	Enum	Operator (1) ▼
	SecurityRearm	Set Reset	Enum	Operator (1) ▼
	SecurityAcknowledge	Set Alarm Acknowledge	Enum	Operator (1) ▼
	SecurityBypass	Set Bypass Conditions	Enum	Operator (1) ▼
	SecurityBypassAll	Set Bypass All Interlocks	Enum	Supervisor (2) ▼
	SecurityMaintenance	Set Asset Maintenance	Enum	Operator (1) ▼
Asset				
	AlarmPriority	Asset Priority	Enum	Normal (4) ▼
	Location	Physical Location	String	
	Custom1	Custom 1 (Alarm Filter)	String	
	Custom2	Custom 2 (Alarm Filter)	String	
	Custom3	Custom 3 (Alarm Filter)	String	
	Custom4	Custom 4 (User Defined)	String	
	Custom5	Custom 5 (User Defined)	String	
	Custom6	Custom 6 (User Defined)	String	
Display				
	GraphicProcess	Process Page Name	String	
	GraphicOverview	Other Page Names (Optional List)	String	

Supervision parameters are stored in *Supervision Data* folder under *Supervision Services*.

Supervision data facets contain each supervision data element. The data elements can be parameterized to control the configuration of the system.

- Variables (IO, Calculated, Disk)
- Alarms (Digital, Time-Stamped, Calculated)
- Trends

- Runtime parameter

Each supervision data facet is of a data type. This type defines the structure of any packed digital data (passed as a word) from the control module. The type is passed back to the supervision parameters facet.

Supervision data is stored in *Supervision Data* Folder under *Supervision Services*.

The following figure shows the supervision data folder:

Identifier	Version	Type	Subtype	Valid	State	Description
\$AINPUTGP_CD	1.0.19	Application Data		True	Approved	Analog Indicator with Configurable Range Data
\$AlarmSummaryGP_CD	1.0.27	Application Composite Template		True	Approved	Alarm Summary Supervision Composite
\$AnalogInputGP_CD	1.0.37	Application Composite Template		True	Approved	Analog Indicator with Configurable Range and Analog Alarms Data Composite
\$AnalogOutputGP_CD	1.0.29	Application Composite Template		True	Approved	Analog Output Cited Data
\$AOUTPUTGP_CD	1.0.18	Application Data		True	Approved	Analog Output Data
\$AOUTPUTLPGP_CD	1.0.4	Application Data		True	Approved	Analog Output Local Panel Data
\$ASelectGP_CD	1.0.21	Application Data		True	Approved	Analog Signal Selector with Monitoring Variables Data
\$CONDSUM1ONGP_CD	1.0.2	Application Composite Template		True	Approved	Conditions Summary Supervision Composite
\$CONDSUM10ONGP_CD	1.0.6	Application Composite Template		True	Approved	Conditions Summary Supervision Composite
\$CONDSUMGP_CD	1.0.10	Application Data		True	Approved	Conditions Summary Data

Supervision Graphics (_CG)

Supervision Graphics facets contain the graphic elements (genies) which can be used on the graphics screen. These elements have their parameters linked to the asset.

Supervision Graphics contain a core subset of the possible graphics required for process automation. If more graphics elements are required, they can be added.

Supervision Graphics are stored in *Supervision Genies* folder under *Supervision Services*.

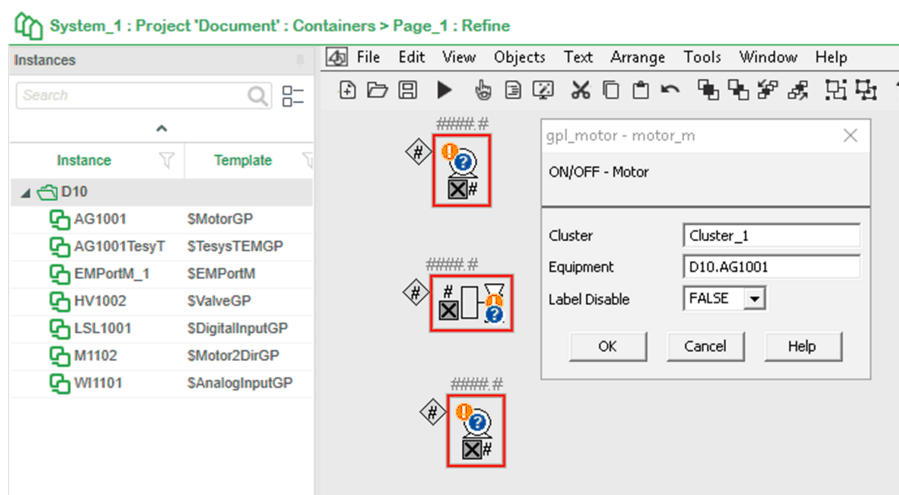
The following figure shows the supervision genies folder:

Identifier	Version	Type	Subtype	Valid	State	Description
\$AlarmSummaryGP_CG	1.0.13	Application Composite Template		True	Approved	Alarm Summary Cited Genies
\$AlarmTextGP_m_CG	1.0.3	Application Genie		True	Approved	Summary of Alarm Conditions ((DINPUT+DALARMI).
\$AnalogInputGP_CG	1.0.7	Application Composite Template		True	Approved	Analog Input Genie Composite
\$AnalogOutput_2wayHorizGP_m_CG	1.0.1	Application Genie		True	Approved	3-way control valve, shown horizontally
\$AnalogOutput_2wayVertLeftGP_m_CG	1.0.0	Application Genie		True	Approved	3-way control valve, shown horizontally
\$AnalogOutput_2wayVertRightGP_m_CG	1.0.0	Application Genie		True	Approved	3-way control valve, shown horizontally
\$AnalogOutput_3wayHorizDownLeftGP_m_CG	1.0.0	Application Genie		True	Approved	3-way control valve, shown horizontally
\$AnalogOutput_3wayHorizDownRightGP_m_CG	1.0.0	Application Genie		True	Approved	3-way control valve, shown horizontally
\$AnalogOutput_3wayHorizLeftDownGP_m_CG	1.0.0	Application Genie		True	Approved	3-way control valve, shown horizontally
\$AnalogOutput_3wayHorizLeftRightGP_m_CG	1.0.0	Application Genie		True	Approved	3-way control valve, shown horizontally

Labels

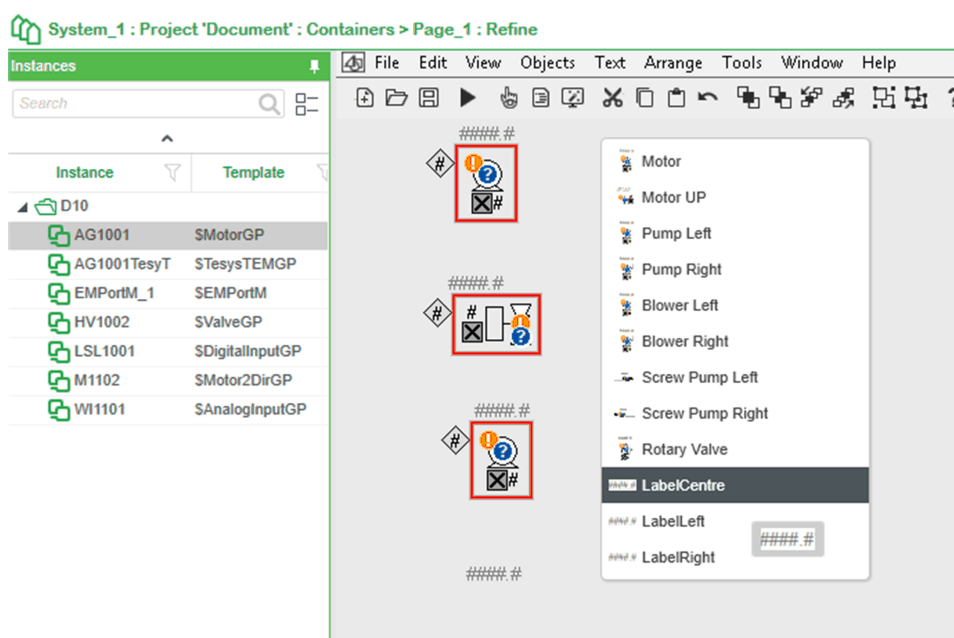
When a graphic element is placed on the screen, the label may overlap with other assets or important background information. You can select a parameter on the graphic (once placed) to remove the label.

The following figure shows the configuration in genie to disable the label:



To display a label, the asset provides a set of label genres with different justification which can be placed as required based on the design needs.

The following figure shows the additional label genres available per asset:



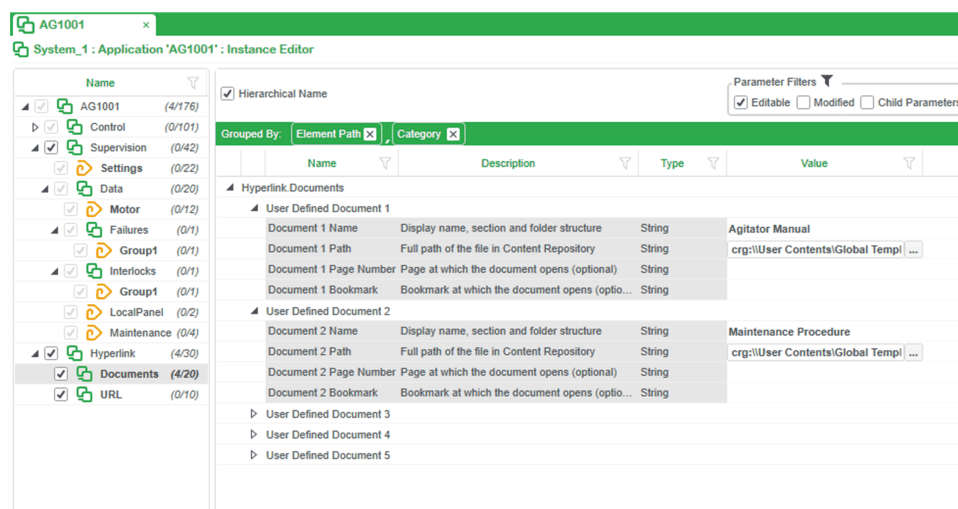
Hyperlink

EcoStruxure Process Expert provides the user links to access documents associated with an asset through the operation client (runtime navigation services). Hyperlinks are defined within the library to provide access to elements configured by the library.

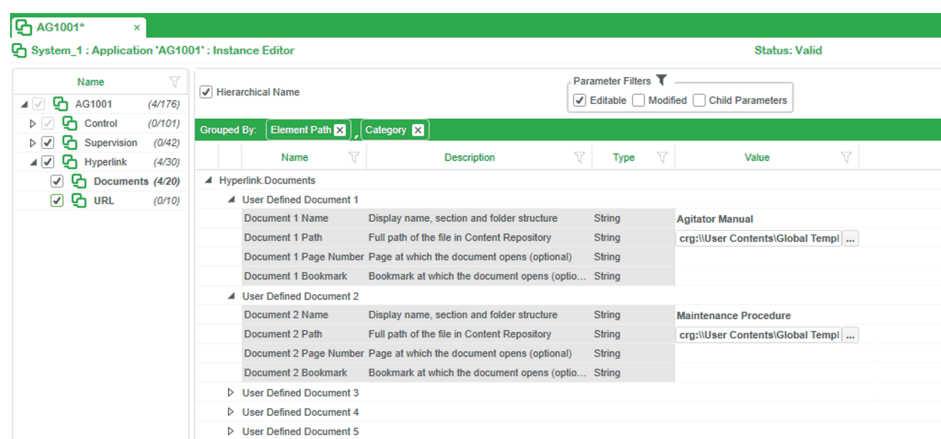
Hyperlinks are also available as configurable parameters for the user to provide their own links to the documents and web pages such as:

- Equipment Manuals
- Process and Instrumentation Diagrams
- Maintenance Procedures
- Maintenance Requests
- Reports

The following figure shows the HyperLink configuration to the documentation and web pages:

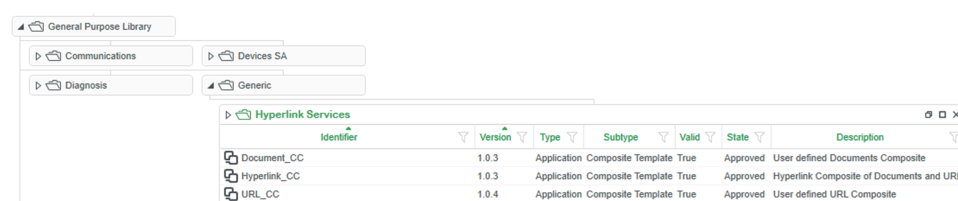


The following figure shows the Runtime Navigation to the documentation and web pages:



Hyperlink composites are stored in *Hyperlink Services* folder under *Generic* folder:

The following figure shows the Hyperlink services folder:



Interfaces

Interfaces are defined within a facet to enable the user to link the facets within a composite and to connect the templates to form the control loops. You can use interfaces to pass information from one facet to another or to generate the links between facets in the asset.

An interface can be a simple interface (comprising a single variable) and composite (comprising many variables as a structure) or be a composite of composites where different interfaces exist within a single interface. An interface can be viewed as a cable which may contain a single wire or many smaller cables with sets of wires.

Interfaces have a type so that configured links are enabled at runtime. Interfaces also contain rules that the system is configured correctly. If the interface is not

configured as per the rules, an error is detected and displayed in engineering. If the interface is configured as per the rules, the generated configuration operates at runtime.

Common Interfaces

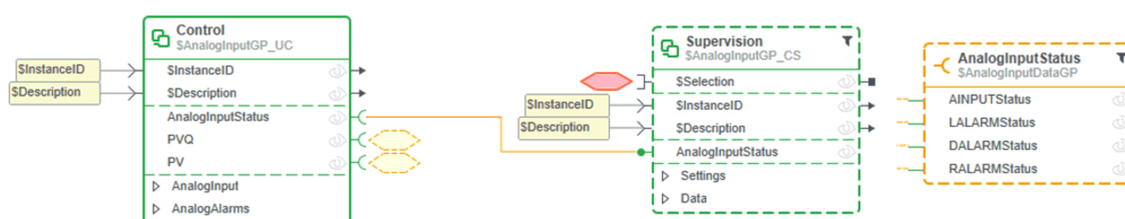
Control to Supervision

Every asset with a control and supervision facet has a single interface which links these two facets. This interface is a composite interface which contains a set of interfaces (one for each optional facet). The composite interface acts as a single link between the control and supervision facets of the asset.

The control to supervision interface contains:

- **Parameters:** Control Parameters which impact the operation of the supervision.
- **Addresses:** Addresses for tags within the controller.

The following figure shows the analog inputs linked by a single interface (it contains many optional services):



Passing addresses through an interface ensures flexibility between the control and supervision facets. A control block can be replaced or updated, and the control continues to operate correctly as the new address automatically gets updated within the supervision configuration.

The address names provided through this interface are created directly from the DDT element generated by the template in the control program hence reducing the chance of human error in linking the controller and supervision.

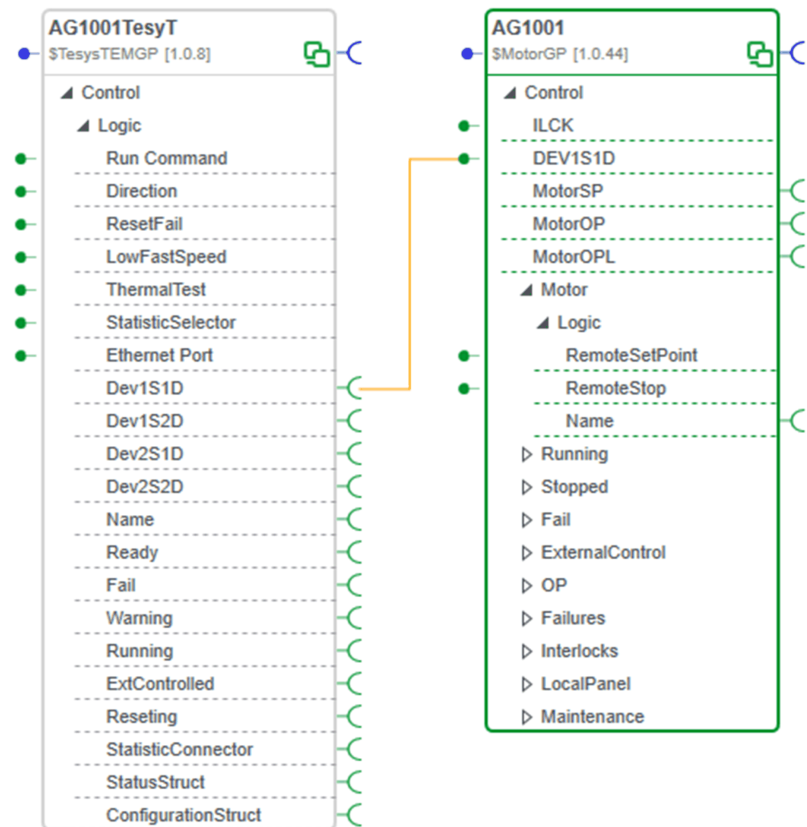
Control Modules

Many control modules are linked to the physical inputs and outputs. More advanced assets also link with devices which provide richer control and diagnostic information.

A motor control has the following interfaces:

- **DEV1S1D** - Single Direction, Single Speed Motor Management Interface (DDT).

The following figure shows the link between \$MotorGP and \$TesySTEMGP template:



Analog Values

The control loops are defined based on the analog values. A simple loop is built for linking an analog input to a PID or rate controller to an analog output.

An Analog Input has three interfaces. Each has different applications:

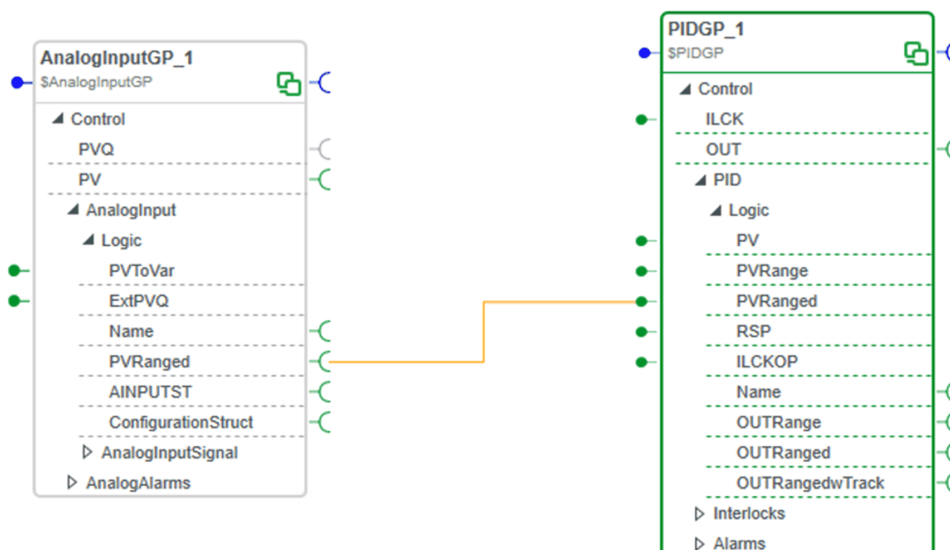
- **PV:** The value of the analog input (REAL)
- **PVQ:** The value of the analog input + channel status (DDT)
- **PVRanged:** The value of the analog input + channel status + range + engineering units and format (DDT)

When connecting the analog value with a calculation (ACALC), the PV is used to provide the value to the calculation block. There is no additional information required.

When connecting the analog value with a template where channel status is also required to be passed along with the analog input value, the PVQ interface is used.

When connecting the analog value with the loop controller (PID), the PVRanged is used to provide the value and the range values so that the values remain linked to the analog input configuration.

The following figure shows the link between \$AnalogInputGP and \$PIDGP template:







































Optional Services

Overview

EcoStruxure Process Expert templates offer optional services for the optimal use of resource. The optional services are selected by enabling these services in the control facet.

The motor template supports optional services for:

- Inputs/Output
 - Running
 - Stopped
 - Fail
 - ExternalControl
 - OP (output)
- Services
 - **Failures** conditions (up to 15)
 - **Interlocks** conditions
 - Group 1 (up to 7)
 - Group 2 (up to 14)
 - Local panel
 - Field switch inputs
 - Control push buttons inputs
 - Maintenance

Name	Description
<input checked="" type="checkbox"/>  MotorGP_3 (0/196) Motor One Direction	
<input checked="" type="checkbox"/>  Control (0/120) Motor One Direction Control Composite	
<input checked="" type="checkbox"/>  Motor (0/9) Motor One Direction Control Composite	
<input checked="" type="checkbox"/>  Logic (0/9) Motor One Direction Control Logic	
<input type="checkbox"/>  Running (0/2) Digital Input Signal	
<input type="checkbox"/>  Fail (0/48) Digital Input Signal	
<input type="checkbox"/>  ExternalControl (0/2) Digital Input Signal	
<input type="checkbox"/>  OPDOSignal (0/2) Digital Output Signal	
<input checked="" type="checkbox"/>  Failures (0/46) Conditions Summary Control Composite	
<input checked="" type="checkbox"/>  Group1 (0/46) Conditions Summary Control Logic	
<input checked="" type="checkbox"/>  Interlocks (0/44) Interlock Conditions Summary Control Composite (14 conditions)	
<input checked="" type="checkbox"/>  Group1 (0/22) Interlock Conditions Control Logic	
<input type="checkbox"/>  Group2 (0/22) Interlock Conditions Control Logic	
<input type="checkbox"/>  LocalPanel (0/13) Motor One Direction Local Panel Control Composite	
<input checked="" type="checkbox"/>  Logic (0/1) Motor One Direction Local Panel Control Logic	
<input checked="" type="checkbox"/>  LPModeDISignal (0/2) Digital Input Signal	
<input checked="" type="checkbox"/>  ZERODISignal (0/2) Digital Input Signal	
<input checked="" type="checkbox"/>  CSDISignal (0/2) Digital Input Signal	
<input checked="" type="checkbox"/>  OFFDISignal (0/2) Digital Input Signal	
<input checked="" type="checkbox"/>  ONDISignal (0/2) Digital Input Signal	
<input checked="" type="checkbox"/>  PBLightOPSignal (0/2) Digital Output Signal	
<input type="checkbox"/>  Maintenance (0/0) Generic Asset Maintenance Control Composite	
<input checked="" type="checkbox"/>  Supervision (0/43) Motor One Direction Supervision Composite	
<input checked="" type="checkbox"/>  Settings (0/22) Common Settings Data	
<input checked="" type="checkbox"/>  Data (0/21) Motor One Direction Data Composite	
<input checked="" type="checkbox"/>  Motor (0/12) Motor One Direction Data	
<input checked="" type="checkbox"/>  Failures (0/1) Failure Conditions Summary Supervision Composite	
<input checked="" type="checkbox"/>  Group1 (0/1) Conditions Summary Data	
<input checked="" type="checkbox"/>  Interlocks (0/2) Interlock Conditions Summary Supervision Composite (14 Conditions)	
<input checked="" type="checkbox"/>  Group1 (0/1) Interlock Conditions Summary Data	
<input checked="" type="checkbox"/>  Group2 (0/1) Interlock Conditions Summary Data	
<input checked="" type="checkbox"/>  LocalPanel (0/2) Motor One Direction Local Panel Data	
<input checked="" type="checkbox"/>  Maintenance (0/4) Generic Asset Maintenance Data	
<input type="checkbox"/>  Hyperlink (0/30) Hyperlink Composite of Documents and URL	
<input checked="" type="checkbox"/>  Documents (0/20) User defined Documents Composite	
<input type="checkbox"/>  URL (0/10) User defined URL Composite	

Selecting the optional service in the control enables the service across all participants. The configuration of all participants are grouped based on the core and optional services to enable optimization of the resulting projects.

Alternate Services

Overview

Within the automation world, you find different approaches to automation. Many of these approaches can be supported by providing the parameters for the standard services within the library. Others would either significantly increase the complexity of the existing services (making them slower to implement and more complex to maintain) or are decisions which should only be made once and should not be offered as a configuration option for each new asset.

To enable the library to support these styles of applications, the library has provided alternative services. An alternative service is a new facet (control and/or supervision) which works differently to the default service, but which can replace the existing service to provide more or different services.

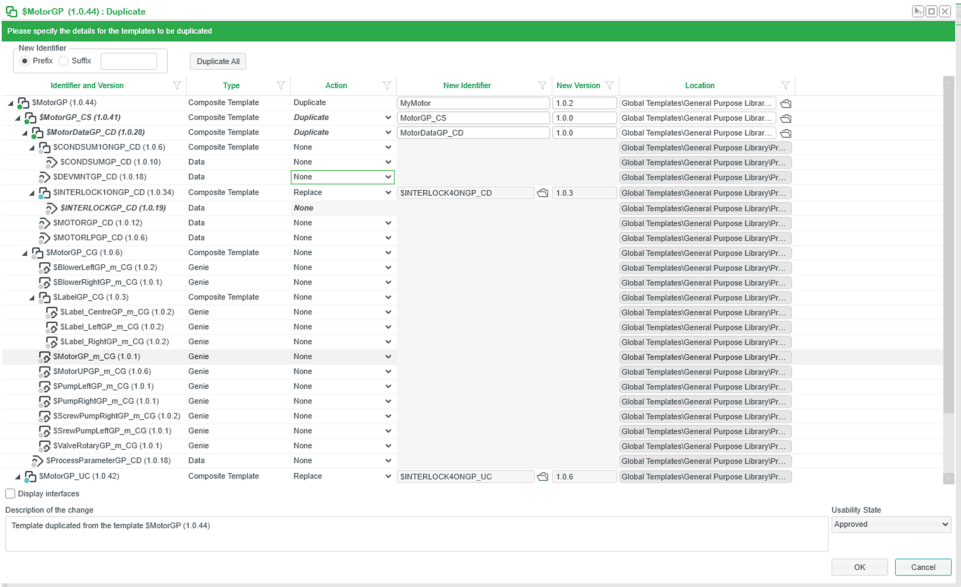
To use an alternative service, you must duplicate the existing template and replace the existing facet with that of the alternative service. Assets generated with the new template operate with the services selected for the project.

You need to duplicate the \$MotorGP template to MyMotorGP template and replace the alternate service INTERLOCK1ONGP_CD by INTERLOCK4ONGP_CD and INTERLOCK1ONGP_UC by INTERLOCK4ONGP_UC as shown in the following figure.

Open the MyMotorGP template in the control and supervision composite, defer the unbound parameters and interfaces of replaced alternate services INTERLOCK4ONGP_CD and INTERLOCK4ONGP_UC.

NOTE: If you are replacing the template with INTERLOCK8ONGP_CD and INTERLOCK8ONGP_UC alternate services, then you must replace the \$ILCK8GRPStatusGP interface in \$MotorDataGP interface as well as in the control and supervision composite.

The following figure depicts the replacement of Interlock1ON (_UC & _CD) template in motor template by Interlock4On (_UC and _CD) through duplicating:



Project Standards

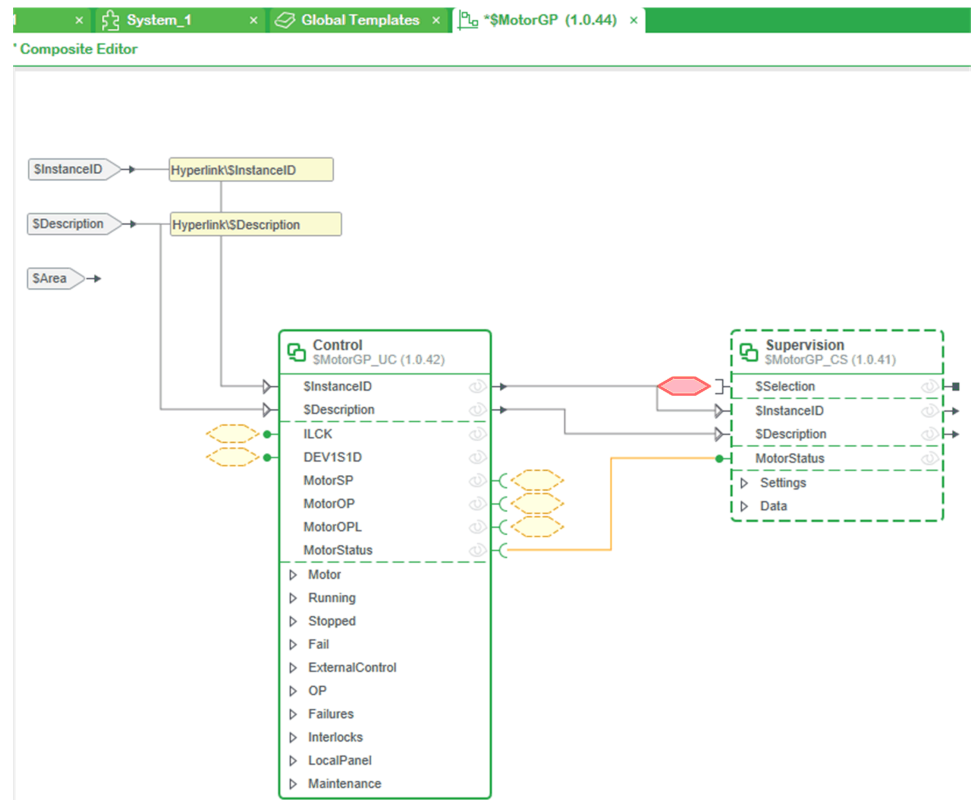
Overview

The library templates expose every parameter available to enable the user to generate (without modification) most automation solutions. Once a project has validated the parameters which have been selected and decided on its preferred operation many of these parameters can remain fixed for the project. To fix these parameters, you have to create a set of project templates.

NOTE: This step is not required but it increases the consistency of engineering and reduces effort.

An example of these parameters are the security settings. The templates provide the ability to define the roles which can be implemented for each action on each asset. This enables the site standards to be implemented and validated directly in the template. Once the asset operation has been validated, do not re-implement this configuration for every asset.

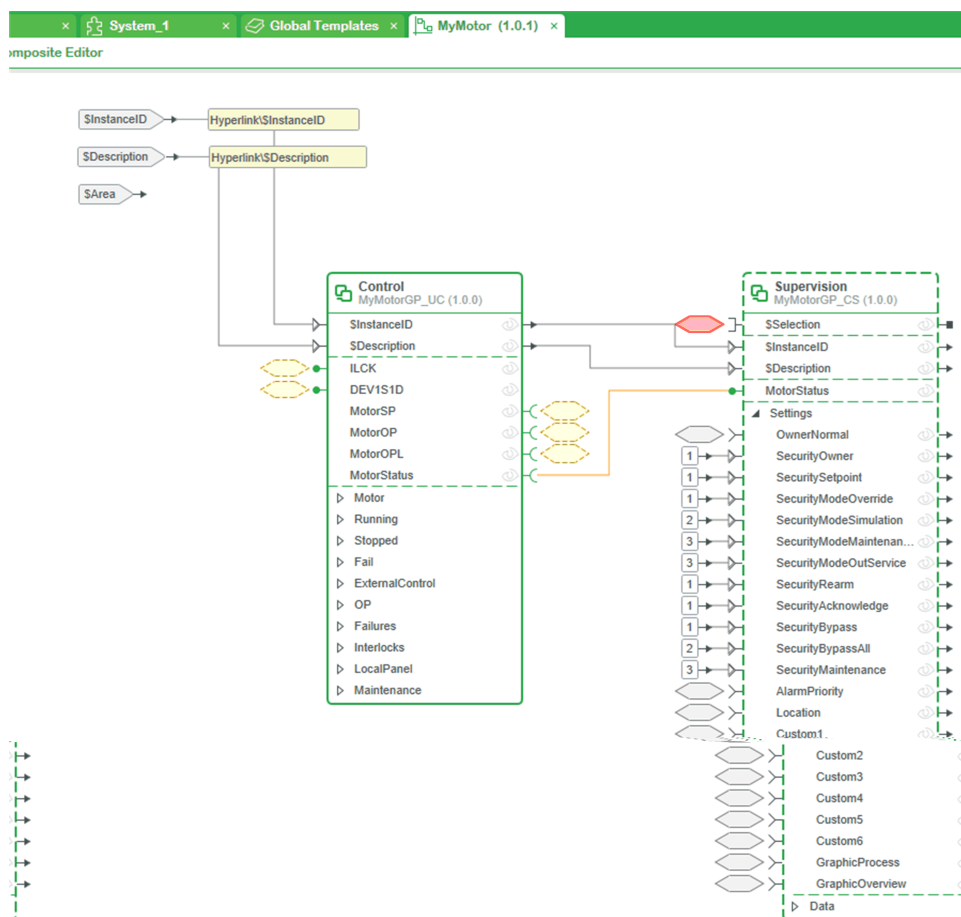
The following figure shows the security parameters in \$MotorGP:



To define a project parameter, you have to duplicate the template, control and supervision facets (see duplicate a template). There is no need to duplicate the data, genie or composites below the top level of the participant. By not duplicating the lower levels of the template, the project template continues to receive updates as new versions of the library are released.

In the project template, you need to modify the control (_UC) and supervision composite (_CS) layers. Remove the *deferred selection* and replace it with the site value for each parameter. This action removes the option for each instance of the template making it simpler. This helps in implementing the site standards for all the assets within the system.

The following figure shows an instance of the project template with fixed security parameters.



Project templates have to be created once there is an agreement on the standard operation of the site and before bulk generation of assets. If changes are required, the templates can still be updated and this impacts all assets. Alternatively, a fixed parameter can be opened for each asset by restoring the selection to *deferred* within the template.

Control

Overview

This chapter describes the basic concepts and details behind each one of the function blocks (DFBs) for implementing the common cross-process and cross-market EcoStruxure control functions.

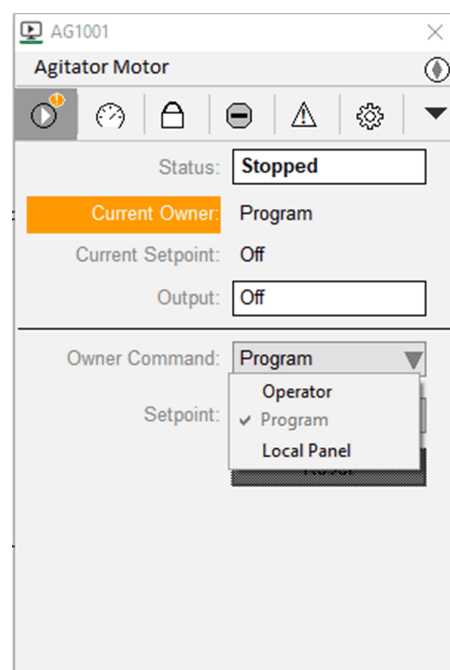
Owner/Setpoint

Overview

All assets with a setpoint have an owner. The owner defines the source of the setpoint value. Each asset will have a sub-set of the following owners:

Owner	Owner Selection	Setpoint is Controlled
Operator	Through faceplate	By the operator through the faceplate
Program	Through faceplate	By sequence logic through the SC public variable
Cascade	Through Sequential Control (SC) public variable	By controller logic through the function block input pins
Local Panel	<ul style="list-style-type: none"> Through field selector switch Through faceplate (if <code>SelectFaceplate</code> parameter is enabled from Asset template) 	From the field through push buttons
External	Through external control input pin in the function block	The setpoint can be controlled by faceplate, DFB input pin, local panel etc. But the Run Feedback overrides and confirmation alarm is not generated.

The following figure shows the motor faceplate with `LocalPanel`. `SelectFaceplate` enabled:



The setpoint displayed on the output pin of the control module is the **current setpoint** for the control module.

The control of the setpoint is exclusive. If the operator requests a motor to start or valve to open, this setpoint remains. A **Failures** condition removes the output until it is cleared. During this time, the setpoint remains and the output once again attempts to start or open the asset as requested. If the site wishes to reset the setpoint (to a passive state) when a **Failures** or **Interlocks** condition occurs, the parameters can reset the setpoint within the asset. When a setpoint is reset using this method, a message appears to the operator indicating that the actions require restarting the system.

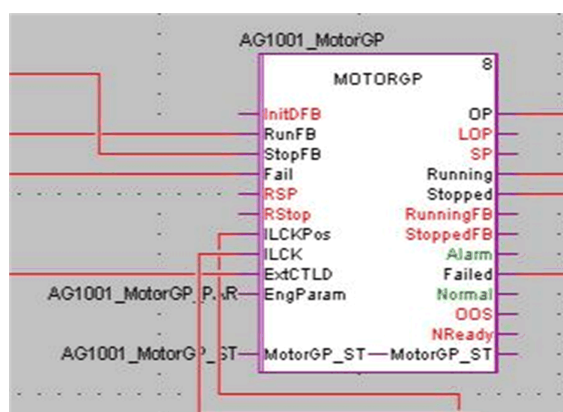
This table describes the setpoint control:

Mode	Interface	Continuous	Pulse
Operator	Supervision data (DDT)	CFG word	HMI write command
Program	Function block public variable	SC.LSP	SC.LSTART / SC.LSTOP
Cascade	Function block input pins	RSP (RSTOP not connected)	RSP / RSTOP
Local Panel	Local panel function block input pins	N/A	ON / OFF pushbutton

By default, assets work in program mode. In the program mode, the setpoint is controlled through sequential logic using the public variable SC (sequential control) within the DFB.

If a project always uses the continuous control, a `Force Remote Setpoint` parameter in the templates forces the owner to cascade. These options are available to give continuous setpoint command as well as pulse setpoint command.

The following figure shows the motor function block highlighting the cascade setpoint control pins and external owner selection:



The following figure shows the motor SC public variable highlighting cascade owner selection:

Name	Value	Type	Comment
AG1001_MotorGP		MOTORGP	
<input type="checkbox"/> <inputs>			
<input type="checkbox"/> <outputs>			
<input type="checkbox"/> <inputs/outputs>			
<input type="checkbox"/> <public>			
<input checked="" type="checkbox"/> AG1001_MotorGP.SC		MotorGP_SC_D	Sequential control structure
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Rem	0	BOOL	Setpoint mode in program owner (0 = Local, 1 = Remote) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.LSP	0	BOOL	Local setpoint (0 = Stop, 1 = Start) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.LStart	0	BOOL	Local start setpoint (Rising edge to start motor) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.LStop	0	BOOL	Local stop setpoint (Rising edge to stop motor) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.OP	0	BOOL	Field command signal (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.LOP	0	BOOL	Logical command signal (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.SP	0	BOOL	Current setpoint (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Running	0	BOOL	1 = Motor is running (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Stopped	0	BOOL	1 = Motor is stopped (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.RunningFB	0	BOOL	1 = Running feedback confirmed and stop feedback not confirmed (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.StoppedFB	0	BOOL	1 = Stop feedback confirmed and running feedback not confirmed (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Alarm	1	BOOL	1 = Confirmation failure active (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Failed	0	BOOL	1 = Motor failure active (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.FailAlarm	0	BOOL	1 = Fail alarm active (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.ILCKD	0	BOOL	1 = Motor interlocked (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Rearm	0	BOOL	Rearm command (1 = Rearm motor) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.RearmR	0	BOOL	1 = Motor requires rearm (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Owner	0	BOOL	Owner (0 = Program; 1 = Operator) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Normal	1	BOOL	Rising edge to activate normal mode (1 = Normal mode active) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Override	0	BOOL	Rising edge to activate override mode (1 = Override mode active) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Sim	0	BOOL	Rising edge to activate simulation mode (1 = Simulation mode active) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.Maint	0	BOOL	Rising edge to activate maintenance mode (1 = Maintenance mode active) (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.OOS	0	BOOL	Rising edge to activate out of service (1 = Out of Service) (Read/Write)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.NReady	0	BOOL	1 = Not ready (Read-only)
<input checked="" type="checkbox"/> AG1001_MotorGP.SC.ExtCTLD	0	BOOL	1 = Motor controlled externally (Read-only)

The following figure shows the motor SC public variable highlighting status word and configuration word:

AG1001_MotorGP_ST		MotorGP_ST_D	- Status
AG1001_MotorGP_ST.STW	16	WORD	Status word (Read-only)
AG1001_MotorGP_ST.RunningFB	0	BOOL	1 = Running feedback confirmed and stop feedback not confirmed
AG1001_MotorGP_ST.StoppedFB	0	BOOL	1 = Stop feedback confirmed and running feedback not confirmed
AG1001_MotorGP_ST.SP	0	BOOL	Current setpoint
AG1001_MotorGP_ST.LOP	0	BOOL	Command signal
AG1001_MotorGP_ST.Alarm	1	BOOL	1 = Confirmation failure active
AG1001_MotorGP_ST.Failed	0	BOOL	1 = Failure active
AG1001_MotorGP_ST.Rem	0	BOOL	Setpoint mode in program owner (0 = Local, 1 = Remote)
AG1001_MotorGP_ST.ILCK	0	BOOL	1 = Interlock active
AG1001_MotorGP_ST.RearmR	0	BOOL	1 = Motor require rearm
AG1001_MotorGP_ST.ExtCTLD	0	BOOL	1 = Motor controlled externally
AG1001_MotorGP_ST.NReady	0	BOOL	1 = Motor not ready
AG1001_MotorGP_ST.FailAlarm	0	BOOL	1 = Fail alarm active
AG1001_MotorGP_ST.SPLocked	0	BOOL	1 = Setpoint lock active
AG1001_MotorGP_ST.ILCKTrip	0	BOOL	1 = Interlock trip active
AG1001_MotorGP_ST.CFGW	1	WORD	Configuration word (Read/Write)
AG1001_MotorGP_ST.Owner	1	BOOL	Owner (0 = Program, 1 = Operator)
AG1001_MotorGP_ST.ILCKBP	0	BOOL	1 = Interlock bypass enabled from supervision
AG1001_MotorGP_ST.Override	0	BOOL	1 = Override mode active
AG1001_MotorGP_ST.Rearm	0	BOOL	1 = Rearm command from supervision
AG1001_MotorGP_ST.LSP	0	BOOL	1 = Motor Start command from supervision
AG1001_MotorGP_ST.SimFBFail	0	BOOL	1 = Simulate feedback fail active in simulation mode
AG1001_MotorGP_ST.Sim	0	BOOL	1 = Simulation mode active
AG1001_MotorGP_ST.Maint	0	BOOL	1 = Maintenance mode active
AG1001_MotorGP_ST.OOS	0	BOOL	1 = Out of service

You can select Local panel owner from a field switch (Local panel/ Control System/ Zero) or from the faceplate (if no field switch is available) when the Local Panel . Select Faceplate parameter is enabled. You can enable the push buttons from the control logic without the owner being local panel.

In some applications, restarting an asset requires appropriate checks to be made. Restarting the asset, when a **Failures** or **Interlocks** condition disappears or when asset is rearmed, is not appropriate. In these cases, you can define to reset the setpoint when the detected error occurs, by enabling the `ResetSPFail` parameter. When the detected error condition occurs, the asset setpoint and output are reset. When the **Failures** condition disappears, the asset restarts only when the setpoint is set to start/open. If the asset is using the remote setpoint (RSP) and the input remains high, then the asset restarts as if a new command was given.

In a continuous process, the product flows through the system and the interlock condition acts as a Detected error for the system. You need to restart the continuous system.

You can define to reset the setpoint after the interlock occurs. This can be achieved by enabling the `ResetSPInterlock` parameter in the asset template. Here, when the interlock condition occurs, the asset setpoint and output are reset. An interlock trip alarm is generated. When the interlock condition disappears, the asset restarts only when the setpoint is set to Start/ Open.

When the setpoint in **ON**, the asset restarts when a **Failures** or **Interlocks** condition clears. The reset `SP` does not reset the `RSP` pin. In effect, the `RSP` is an ongoing restart command.

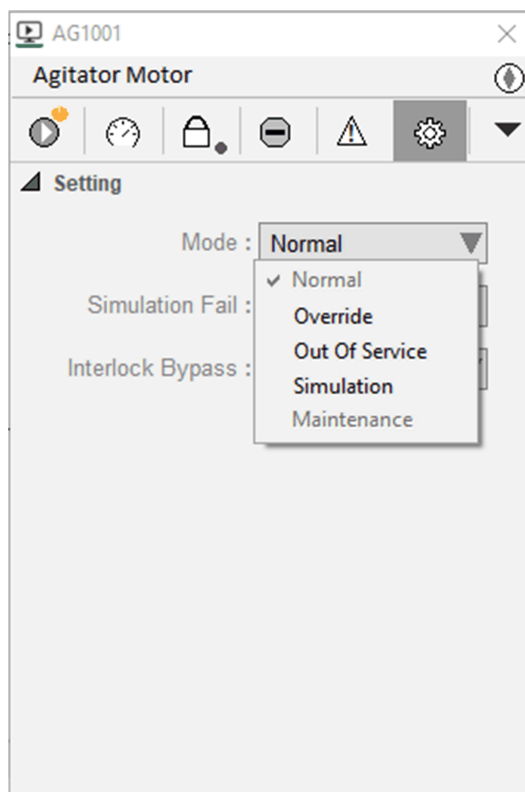
Modes

Overview

All assets can operate in different modes. Following are the modes of operation:

- **Normal:** Allows the control module to be used as part of the control system.
- **Override:** Allows the control module to continue to operate with a detected feedback error . In this mode, the input feedback follows the output or is overridden.
- **Simulation:** Allows the logic of sequences or control modules to be tested in a simulated environment. The inputs to the control system are simulated as per override and the output `OP` is suppressed. The logical output `LOP` represents the simulated output. You can not change to simulation mode if the asset is running.

- **Out of Service:** Excludes the control module operation from the control system.
- **Maintenance:** Allows the operation of the control module for maintenance and testing. Operator commands are only accepted from the HMI or Local panel. Interlock conditions are bypassed. **Failures** conditions do not impact the output. Alarms are displayed only in the faceplates **Alarm** tab and not in the active alarm page.



This table summarizes the capabilities in each operational mode:

	Available owner	SP	Feed-back	OP (Field)	OP (Logical)	Failures and Interlocks
Normal	Any	Processed	Field	Calculated	=OP	Applicable
Override	Any	Processed	Override	Calculated	=OP	Applicable
Simulation	Any	Processed	Override	Reset	Calculated	Applicable
Out of Service	Any	Disabled	N/A	Reset	Reset	Not Applicable
Maintenance	O/L	Processed	Field	Calculated	=OP	Not Applicable

NOTE: O: Operator, L: Local Panel.

Output

Overview

The output (OP) of any control module is the signal to be passed to the field IO. The output is controlled by the **Setpoint**, **Failures**, **Rearm required** and the **Interlocks** conditions:

Interlock	Failure	Rearm required	OP
X	1	X	0
X	X	1	0
1	0	0	=ILockPos
0	0	0	=Setpoint
NOTE: X represents any value			

An active **Failures** condition resets the asset output irrespective of whether `EngParam.FailRearmEn` is enabled.

If the **Failures** condition disappears and `EngParam.FailRearmEn` is enabled, the asset output will remain inactive.

You can use the interlock conditions to force an asset to run independent of the setpoint. When interlocked, the output follows the Interlock Position (`ILockPos`).

In normal condition, the asset output follows the setpoint.

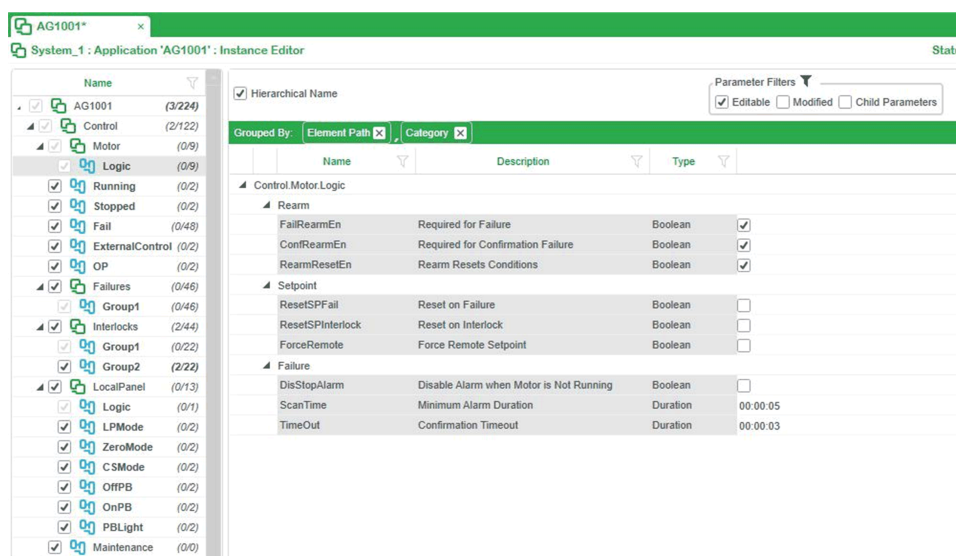
Confirmation Failure

Overview

All control modules with feedback provide non-confirmed operation alarms when the feedback signals do not match the output signal within the defined timeout.

Motor Template logic parameter highlighting `Confirmation Timeout` and `Rearm required` for non-confirmed operation.

This figure shows the motor template logic parameters `Confirmation Timeout` and rearm required for non-confirmed operation:



⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- The `ConfRearmEn` engineering parameter has to be set to true for the asset to withdraw the output under non-confirmed operations.
- Changes to the configuration of these parameters have to be performed by competent personnel only.

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

A unsuccessful operation alarm can be caused by a detected error of the asset or by a detected error of the feedback sensor. You can define the action of the output when a non-confirmed operation occurs. When the Confirmation Rearm Enable (ConfRearmEn) parameter is enabled, the output will become inactive until the asset is rearmed otherwise the output will remain active even though the alarm is generated. Non-confirmed operation becomes an alert condition rather than a detected error.

If the feedback sensor error is the cause of non-confirmed operation, you can select to operate in **override** mode. In this mode, the feedback conditions are assumed to match the output and the confirmation alarms are effectively bypassed.

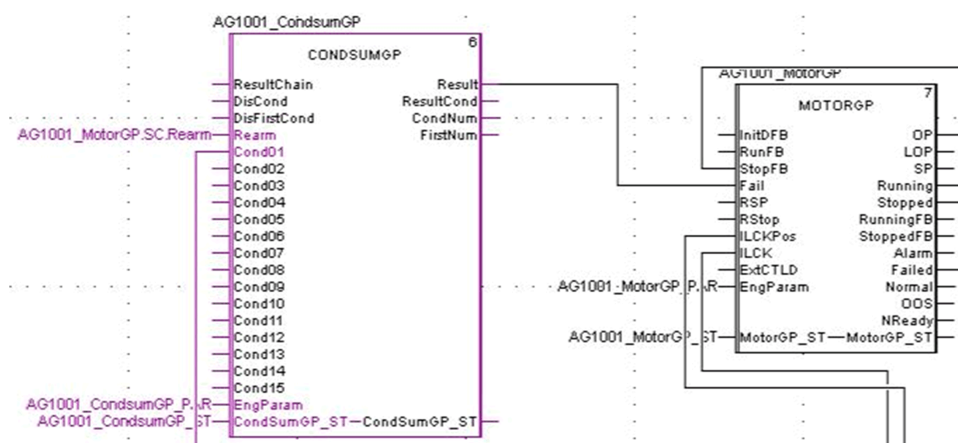
If the **Running** input pin in the asset template is not connected, the non-confirmed operation feedback will be bypassed.

Inoperable Asset

Overview

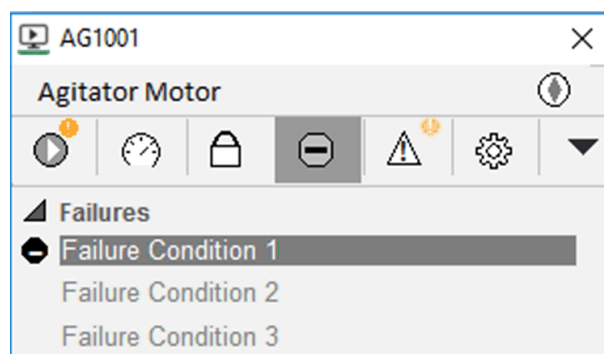
All motorized assets have a **Fail** input pin to display that an inoperable asset has been detected which will prevent its operation. The number of conditions can be extended using condition summary blocks.

This figure shows the motor function block with **Fail** input pin highlighted.

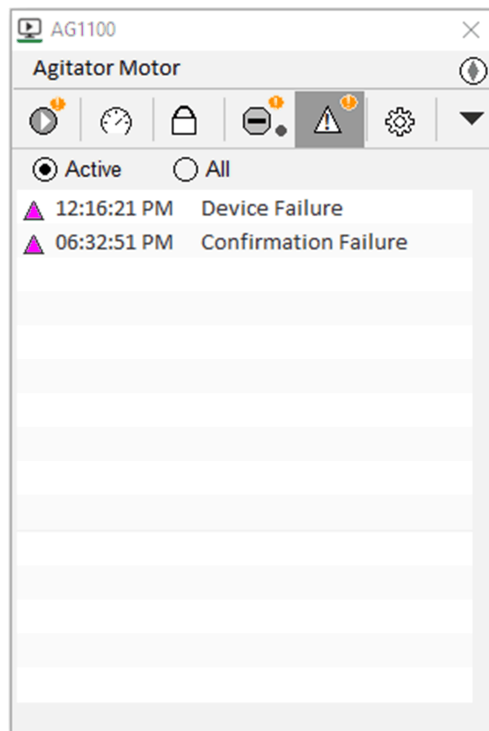


When a **Failures** condition occurs, it generates an alarm in the supervision. The alarm condition is the same for any **Failures** condition. If more than one condition exists the condition which is generated first is detected and displayed as the cause (in first position).

This figure shows the motor **Failures** tab with the active condition highlighted:

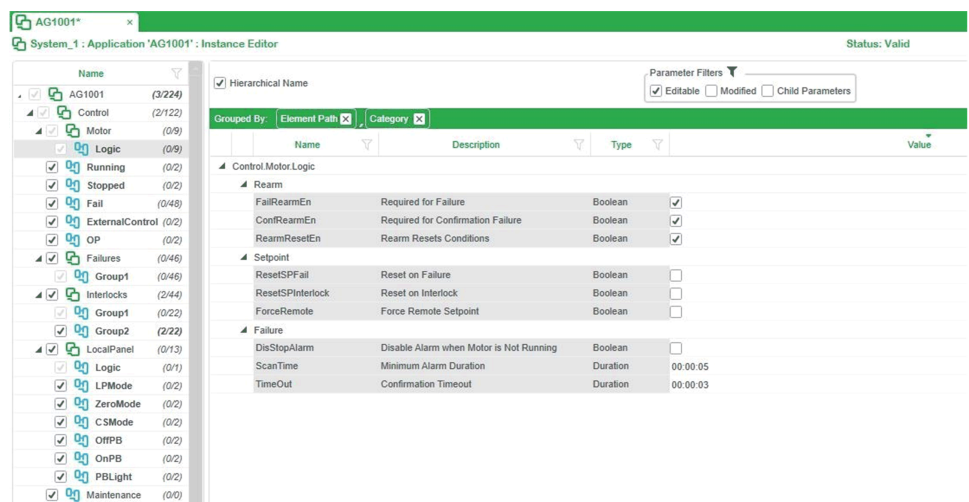


Motor alarm tab with the inoperable asset alarm highlighted:



If an asset is idle (Setpoint = 0 AND Output = 0), a **Failures** condition does not change the process. For such idle assets, you can define a new **Failures** condition that will not generate an alarm within the supervision by enabling the `DisStopAlarm` parameter in the asset template. This is done to reduce the number of alarms from non-active processes which may interrupt the operation of the process.

The following figure shows the `DisStopAlarm` parameter of the Control Motor Logic service.



Interlocks

Overview

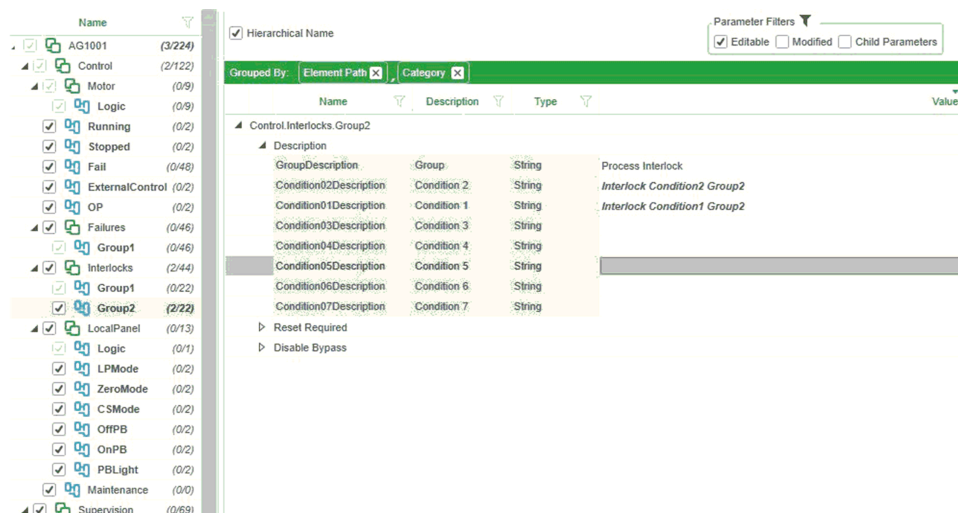
All assets with outputs have interlock conditions to prevent or force the output to a specific value.

Assets with one output have a single interlock condition (and position) while assets with two outputs have dual interlock conditions (and positions). With these

interlock conditions and positions, you can do the configuration to prevent or force the state of each output.

If selected (default), one interlock function block (with 7 interlock conditions) is generated with the asset. If more than 7 interlock conditions are required, then **Interlock Group2** can be enabled providing 14 interlock conditions.

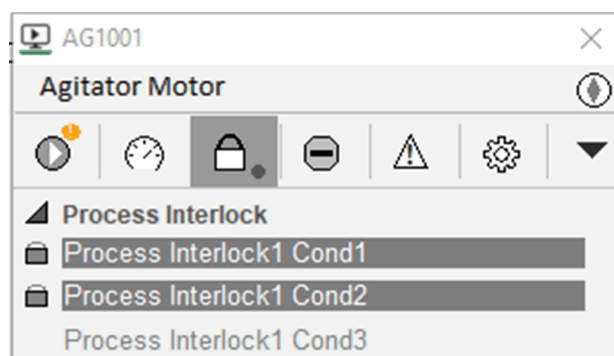
The following figure shows the motor template interlock parameters highlighting interlock group2 to enable 14 interlock conditions:



You can configure various interlock related configurations, for example the description of the interlock, Reset Required and Bypass disable from the asset template for each interlock condition.

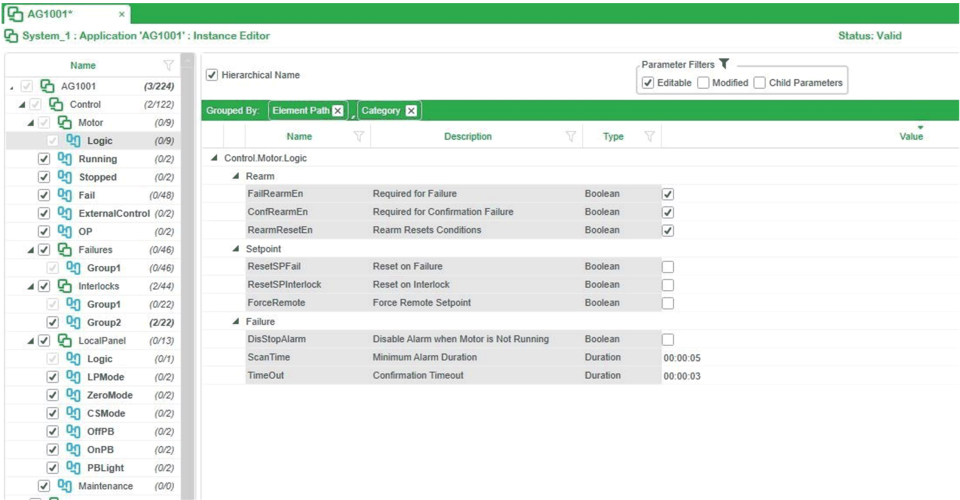
If more than one condition exists that generates an interlock, the conditions which generated the interlock are detected and displayed as the cause (in first position).

The following figure shows the motor interlock tab highlighting active interlock conditions:



If an asset is idle (Setpoint = 0 AND Output = 0), an interlock condition does not change the process. For such idle assets, you can define a new interlock condition that will not generate an alarm within the supervision by enabling the `DisStopAlarm` parameter in the Asset template. This is done to reduce the number of alarms from non-active processes which may interrupt the operation of the process.

The following figure shows the DisStopAlarm parameter of the Control Motor Logic service.



Restart

Overview

A control module has two groups of **Failures**:

- Unsuccessful operation (feedback does not match the output)
- Inoperable asset (Fail input pin is active).

If you can define that a Rearm is required (ConfRearmEn and/or FailRearmEn), the output will remain off until the operator presses the rearm button on the operator tab of the faceplate. If a rearm is not required, the output will be on when the condition is cleared (or remain active for a non-confirmed operation).

Name	Description	Type	
Control.Motor.Logic			
Rearm			
FailRearmEn	Required for Failure	Boolean	<input checked="" type="checkbox"/>
ConfRearmEn	Required for Confirmation Failure	Boolean	<input checked="" type="checkbox"/>
RearmResetEn	Rearm Resets Conditions	Boolean	<input type="checkbox"/>

As well as rearming the asset, you can define that individual **Failures** conditions can require a reset from the operator. All conditions have to be reset to clear the **Failures** condition which is required before it is possible to re-arm the asset. To speed the process to rearm an asset with reset conditions on individual **Failures** it is possible to select that a rearm operation will reset the available individual **Failures** conditions. With this option selected the rearm is available when conditions can be reset but a second rearm may be required to clear all the detected failure conditions.

Name	Description	Type	
Control.Interlocks.Group1			
Reset Required			
Condition01ResetReq	Condition 1	Boolean	<input checked="" type="checkbox"/>
Condition02ResetReq	Condition 2	Boolean	<input type="checkbox"/>
Condition03ResetReq	Condition 3	Boolean	<input checked="" type="checkbox"/>
Condition04ResetReq	Condition 4	Boolean	<input type="checkbox"/>
Condition05ResetReq	Condition 5	Boolean	<input type="checkbox"/>
Condition06ResetReq	Condition 6	Boolean	<input type="checkbox"/>
Condition07ResetReq	Condition 7	Boolean	<input type="checkbox"/>
Disable Bypass			

NOTE: When an asset is rearmed, it will attempt to match the setpoint of the asset (unless interlocked). The operator should check the setpoint and interlock status to determine which action will occur once the asset is rearmed. To confirm the rearm action, the setpoint can be reset when the detected error occurs (`ResetSPFailure`). In this case, a rearm will not make the asset active until a new command is received.

This table explains the **Operator** action for restarting the asset when the asset setpoint is active:

Condition parameter	Asset parameter			Asset will restart when condition is clear and
Reset required	Rearm required	Reset on rearm	Reset SP	
0	0	0	0	None
1				Interlocks/Failures condition is reset
0	1	0		Asset is rearmed
X		1		Asset is rearmed
1		0		Interlocks/Failures condition is reset and then the asset is rearmed.
0	1	0	1	Restart command is issued.
1				Interlocks/Failures condition is reset and restart command issued.
0	1	0		Asset is rearmed and restart command issued.
X		1		Asset is rearmed and restart command issued.
1		0		Interlocks/Failures condition is reset and asset is rearmed and restart command issued.

NOTE:

- x represents any value.
- The **Reset Required** indicates the individual **Interlock/Failures** conditions reset required parameters.
- The **Rearm Required** indicates the **FailRearmEn** and/or **ConfRearmEn** asset parameters.
- The **Reset on rearm** indicates the **RearmResetEn** asset parameters.
- The **Reset SP** indicates the **ResetSPFail** and **ResetSPInterlock** asset parameters.

Maintenance

Overview

All motors and valves can be checked to collect maintenance data. The default facet collects the essential maintenance data. If more data is required, alternative facets may be selected.

The default facet collects:

- **Motor:** Run hours, Operations
- **Valve:** Open time, Operations

Name		
	AG1001	(12/224) M
	Control	(5/122) M
	Motor	(1/9) M
	Running	(0/2) Di
	Stopped	(0/2) Di
	Fail	(0/48) Di
	ExternalControl	(0/2) Di
	OP	(0/2) Di
	Failures	(0/46) Co
	Interlocks	(4/44) Int
	LocalPanel	(0/13) M
	Maintenance	(0/0) Ac
	Supervision	(4/69) M
	Settings	(4/22) Co

AG1100
Agitator Motor

Maintenance Data
Total Run Hours : 0 h
Total Starts : 0

Supervision

Overview

This chapter describes the process and device supervision services.

Situational Awareness

Overview

The aim of situational awareness is to ensure that the user is aware of:

- The current process situation (the meaning of the value displayed)
- The future potential process situation (the direction and speed of change)
- The cause of a process event
- The impact of a process event
- The risk in maintaining the production

To deliver on all these values the library introduces new features which impact the operation of the control system.

Asset Colors

People have a strong reaction to color. Red elements on a process screen are in most cases the first elements which the operator will see when an operator screen is displayed. Power systems, for this reason, use red to display hazard. In a process system red has often been used for stopped or running/hazard. In a process system this state is a normal state based on the operation of the process and screens have become filled with high impact colors.

Situational awareness seeks to highlight the cause of an inoperable condition by making the asset which has caused the inoperable condition to appear in RED (and in alarm) while showing the process which has been stopped in more muted colors. The library has followed the traditional color scheme for situational awareness systems.

Asset	State	Color
Motor	Stopped	White
	Running	White
	Starting	Dark grey
	Unknown	Black
Valve	Closed	White
	Opening	Light green
	Open	Dark grey
	Stop	Grey

NOTE: While these colors are traditional there is no reason that they cannot be adapted to meet the needs of your site. The colors can be selected by adapting parameters in the runtime project, this is detailed in the customization section, page 86.

Color name
Black
Blue

Color name
Green
Cyan
Red
Magenta
Brown
Grey
DkGrey
LtBlue
LtGreen
LtCyan
LtRed
LtMagenta
Yellow
White
State change green

Analog Values

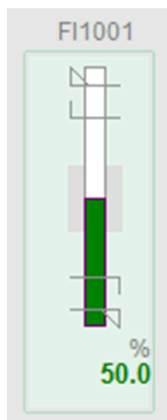
Overview

The quality of production is determined by the operator's ability to ensure the optimal operation of the process. To detect when production quality is changing and act rapidly to restore operation. To enable operators to rapidly detect these changes the library implements situational awareness.

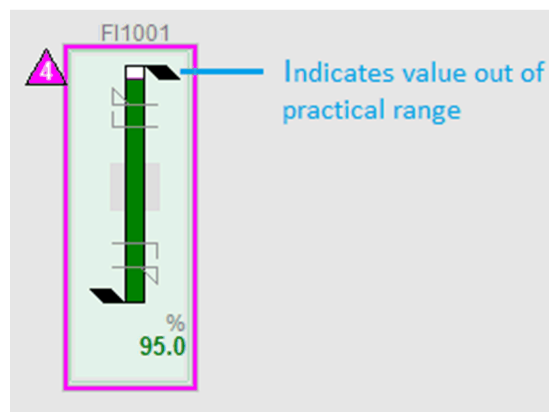
Practical Range

In a typical control system analog values are displayed as numbers and are highlighted only when the value exceeds the alarm limits defined. This encourages the control system engineer to define many alarms to help the operator understand the value.

With Situational awareness values are displayed as bar-graphs (as well as numerical values). Doing this shows both, the value and the position of that value relative to the expectations of the process being controlled. The bar-graph display is scaled to the **practical limits** defined for the process (not the range of the sensor) thus the position on the bar-graph represents both its value and its value in relationship to the needs of the process. If the value is outside the practical limits the bar-graph adds tab indicators and displays the full range of the sensor. Practical ranges can be implemented on any analog input but are mostly used when the analog value uses only a portion of the sensor range.

PV within the practical range
(InsidePR)

PV outside practical range (OutsidePR)



Examples applications for practical range:

Water pH	Furnace temperature	Tank weight
Sensor range: 0-14	sensor range: 0-2000	sensor range: 0-1000
practical range 6-8	practical range 1200-1600	practical range 500-900

In each case the value of the sensor becomes more valuable as a temperature of 1250 or a weight of 600 are both low values for the process even when they are almost average when compared to the sensor range.

Practical Deviation

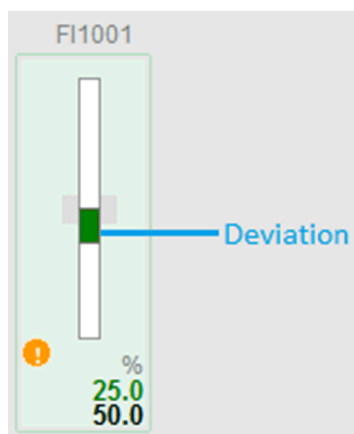
The practical deviation is requested to enable the display of a deviation bar graph. A deviation bar graph shows the setpoint as the central point and uses the practical range to display values which are above or below the setpoint. The practical deviation represents the range of deviations from which the operator could be expected to recover the process.

Deviation Bar graphs are commonly used to display information on overview screens.

Examples:

Ingredients vs setpoint (measurement error)	Practical deviation = acceptable error
Furnace Temperature (setpoint error)	Practical deviation = acceptable error
Daily Production (production performance)	practical deviation = production hour

This figure shows the deviation bar of genie:



In each case the operator can see their performance relative to the setpoint and how close they are to the acceptable performance. Variables with low levels of error will appear to increase and decrease very rapidly on the deviation bar graphs while variables with less constraint will move more slowly.

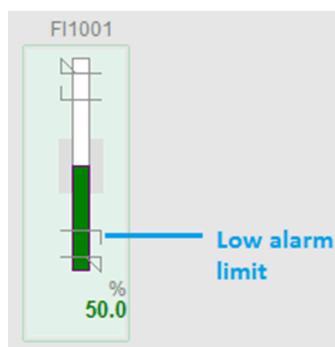
Limits

The bar-graph display will also display the value of limits which have been defined for the analog value:

- Level Alarms (Limit Indicators)
 - High-High Limit
 - High Limit
 - Low Limit
 - Low-Low Limit
- Optimal Zone (Grey Area)
 - Optimal High
 - Optimal Low

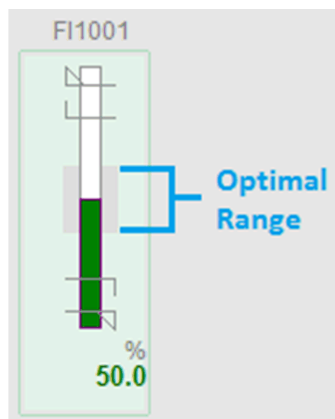
The bar-graph displays the level alarm indicators which are enabled as shown in the image below. If the alarm is shelved or disabled, then the indicator will disappear. This allows the operator to understand how close they are to the limits which will trigger a process event or alarm.

This figure shows the alarm limit:



The bar-graph displays the optional zone when enabled. The optimal zone shows the operator the "best practice" analog value enabling new operators to rapidly understand the expected operational values for quality production.

This figure shows the optimal range:



The optimal zone can be a fixed range or can adjust as a fixed offset from the setpoint. This value can be linked to the deviation alarm. The operator should see that the value is approaching the edge of the optimal zone and take action to restore it to the center of the optimal zone.

Numeric Entry

To enable the numeric touch pad in Touchscreen based system for entering the numeric values, you need to add the below parameters in `Citect.ini`:

Parameter Details

Section:

Parameter:

Value:

Comment:

After configuration, the following popup will be available for entering numeric value:

G10PMP01.StartTimeOut X

<< < > >>

7 8 9

4 5 6



1 2 3



0 <- Clr

-/+ .

OK Cancel

Fast or slow buttons to increment or decrement analog values.

Icon	If Engineering High and Engineering Low Limit is configured	If Engineering High and Engineering Low Limit is not configured
	Increase the value by 5% of the full scale..	Increase the value by 5 numbers.
	Increase the value by 1% of the full scale.	Increase the value by 1 number.

Icon	If Engineering High and Engineering Low Limit is configured	If Engineering High and Engineering Low Limit is not configured
	Decrease the value by 1% of the full scale.	Decrease the value by 1 number.
	Decrease the value by 5% of the full scale.	Decrease the value by 5 numbers.

Control Modules

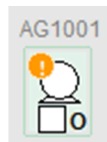
Overview

The quantity of production is determined by the operator's ability to ensure the continuous operation of the process. To detect the cause of any disturbance and to return the process to its operating state. To enable operators to rapidly detect process event and to understand risks which may lead to disturbances the library implements situational awareness.

Abnormal Owner

Every asset with a setpoint has an owner. The owner is only displayed on the process display when the owner is abnormal. Doing this removes "normal" information from the display to allow the operator to focus on any abnormal situation.

The figure shows the abnormal owner indication:



All control modules are, by default, `normal` when they are in program mode (controlled by a sequence or continuous system). All sequential control assets are, by default, `normal` when they are in operator mode (controlled by the operator). If an asset has a different `normal` owner this is configured in the template.

Always display owner.

[Section]Parameter = 1

Abnormal Operational Mode

A control module can be operated in `normal` or in an `abnormal` mode of operation. An abnormal mode of operation is defined as:

- Abnormal owner (as defined in the asset)
- Abnormal mode (any mode except normal)
- **Interlocks/Failures** condition bypassed
- Interlock condition bypassed

In any of these conditions the control module is being operated with increased risk and the operator is made aware of this by the display of the abnormal operational mode symbol.

The abnormal operation symbols are displayed on the faceplate as well as the process display and can be used to identify the source of the abnormal operational mode.

Operator Action Required

A control module may require operator action before it can operate. If this is the case the Abnormal operational mode indicator will flash. A flashing symbol represents:

- Asset requires a rearm
- **Interlocks/Failures** condition requires a reset
- Interlock condition requires a reset
- Alarm requires an acknowledgment.

Workspace

Overview

The workspace is a new supervision concept in this release of EcoStruxure Process Expert. The workspace replaces the templates used in previous releases and instead defines a set of panes on a master page which can work either on a single monitor or across multiple monitors.

The Workspace is the engine that manages the contextual content changes within each screen. It enables contextual navigation and allows the content on a page to be automatically updated at runtime based on the current Asset selected on a client.

For example, if a pump is selected on a display page, the autofill process will update any relevant panes (Faceplate and Process pages) with content that directly relates to the pump, and optionally its parent or child hierarchy.

A startup context is set for each Workspace by linking each screen to a Level 1 entry in a project menu named **Navigation**. This provides a starting point for any contextual navigation that occurs within the Workspace. Startup context is set for each screen via the Screen Setup page of the Computer Setup Wizard (refer to screen profile), page 45.

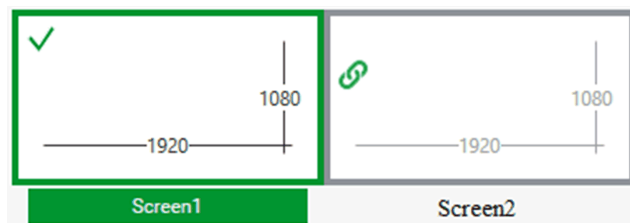
The workspace concept is designed to simplify the process of designing a user interface for:

- Multiple monitors
- Different aspect ratios

The workspace does this by making each pane within the workspace a unique page and assembling these different panes as defined by a master page. The master page used on each monitor is selected per client allowing each monitor of each workstation to display the appropriate information.

Screen Profile

A screen profile defines the arrangement of screens used by a client with multiple monitors:

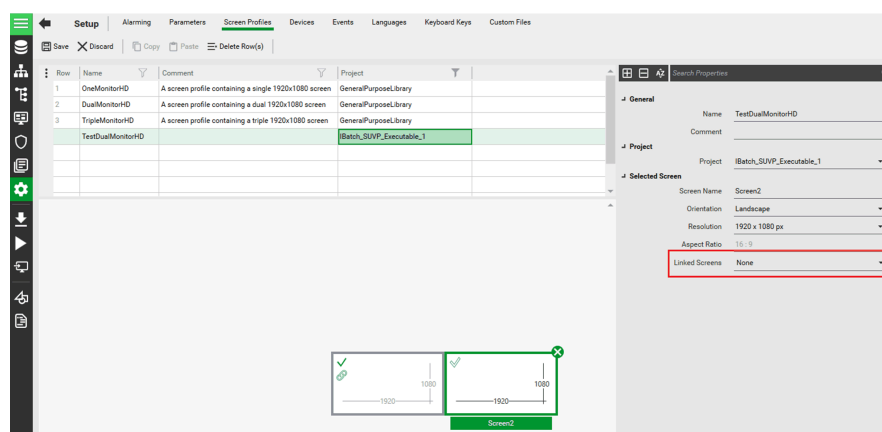


The following screen profiles are available by default with the library:

- One Monitor HD- A screen profile containing a single 1920x1080 screen
- DualMonitorHD- A screen profile containing a dual 1920x1080 screen
- TripleMonitorHD- A screen profile containing a Triple 1920x1080 screen

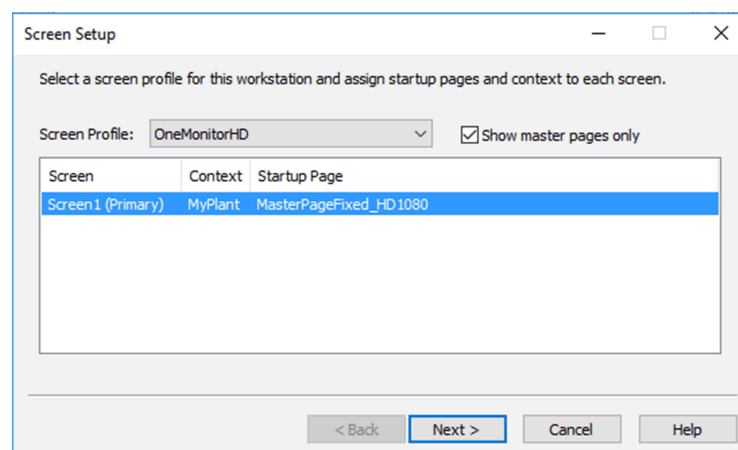
NOTE: If you want to see the two different pages on two different monitors, then you need to add the following parameter:

- TestDualMonitorD - A screen profile containing a dual 1920x1080 screen



You can then apply a screen profile to a client using the Screen Setup page of the Computer Setup Wizard.

This figure shows the selection of screen profile, context and master page for operation client:



You can create a screen profile in the Setup activity in Plant SCADA Studio as per project requirement.

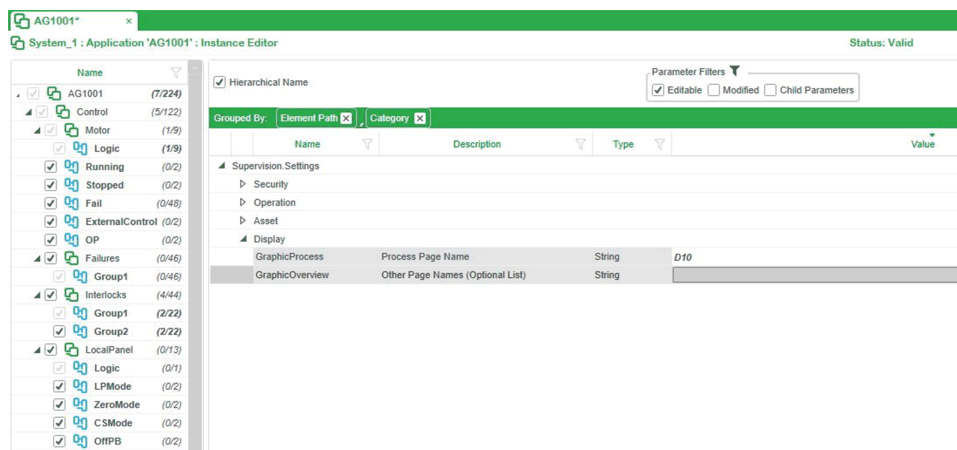
Panes

There are two ways to initiate an update within this master page:

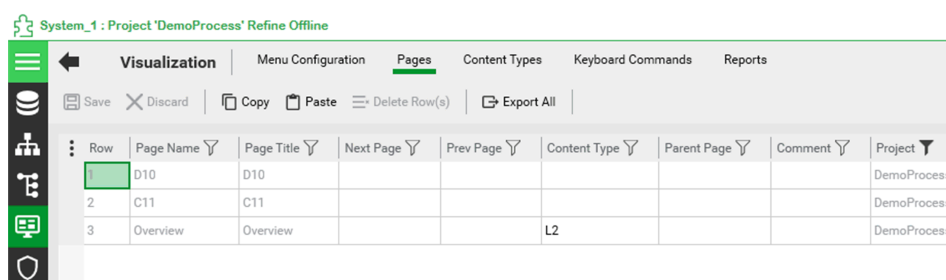
- **Commands** - typically used on the Header Bar or a navigation menu to display a page.
- **Contextual updates** - the content of a pane may update when a piece of equipment comes into context.

Contextual updates are enabled by associating respective page contents with the Asset in asset instance editor. Each page which is assigned to an asset should be assigned a content type. This allows a pane to update automatically when the asset comes into context.

This figure shows the process of assigning a page with the asset:



This figure shows the process of defining the content type for a page:



This describes the content types which are available in the Library:

Name	Description
General	General pages. Example: Alarm, Trend.
FP	Faceplate pages
L1	Level 1 pages
L2	Level 2 pages
L3	Level 3 pages
L4	Level 4 pages
Master	Workspace startup pages

It is necessary to define the content type for pages. If content type is not defined for any page which is assigned to asset, then it is displayed in the Process Display area of the Master page. This gives the flexibility that a process page of L3 type can still be displayed in the process display without configuring it.

By default, on selecting an equipment, it will not update content of the pane from where equipment is selected. However, it can be enabled by adding and configuring following parameter to "1":

Section Name	Name	Description
GPL.Display.Treeview	Cluster	0= Cluster is not displayed in Equipment tree. 1= Cluster is displayed in Equipment tree.

Master Page

A master page provides the layout for the content that is managed by a Workspace. It differs from other graphic pages, as it is made up of a set of Panes which host individual pages at runtime, refer to *Panes*, page 46 for more information.

The size of a master page needs to match the resolution of the screen on which it will display. All master pages that are provided with library have HD1080- 16:9 (1920 x1080) resolutions. To specify the master page that displays on a screen when runtime starts, use the Screen Setup page of the Computer Setup Wizard (refer to *screen profile*), page 45.

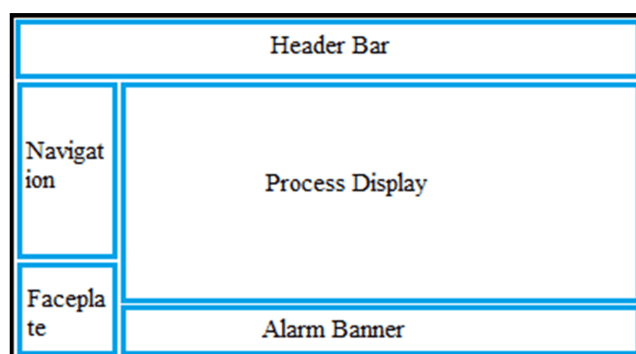
This table describes the Master pages which are available with the library.

Name	
<i>MasterPageFixed</i>	Header Bar, Navigation, Faceplate, Alarm Banner, Process Display
<i>MasterPagePopup</i>	Header Bar, Navigation, Alarm Banner, Process Display
<i>MasterPageSystem</i>	Header Bar, System page (e.g Alarm page, Process Analyst Page)
<i>MasterPageOverview</i>	Header Bar, Overview Display

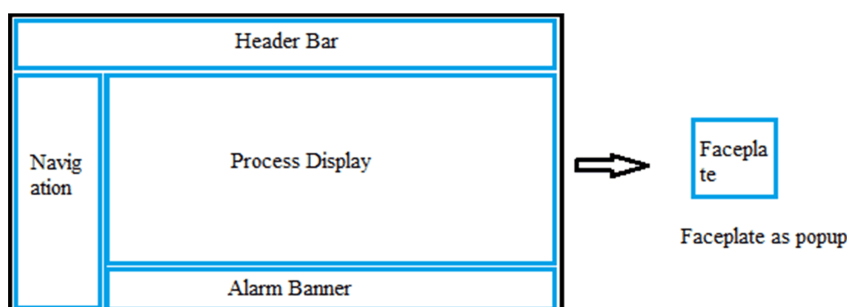
Master Page Layout

MasterPageFixed allows to open faceplate of selected asset in designated area. This layout does not allow to open multiple faceplates of an asset. While *MasterpagePopup* opens faceplate of an asset as a popup. Multiple faceplates of different assets can be open at the same time. The following figures defines the layout of different master pages available in library.

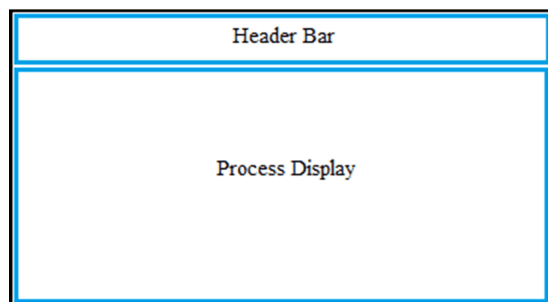
This figure shows the *MasterPageFixed* layout:



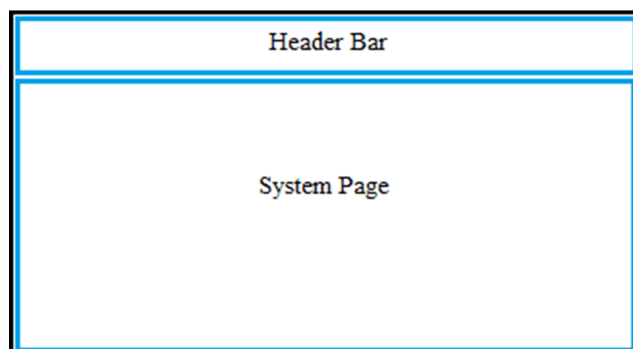
This figure shows the *MasterPagePopup* layout:



This figure shows the `MasterPageOverview` layout:



This figure shows the `MasterPageSystem` layout:



In `MasterPageFixed` and `MasterPagePopup`, Process display area is defined to autofill the L1,L2,L3,L4 or General content type of selected asset while in case of `MasterPageOverview`, it is defined to autofill the L1 or L2 content type of selected equipment.




`MasterpageSystem` is defined to autofill the General content type page in system page pane.







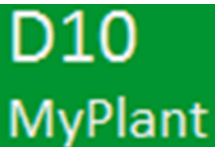
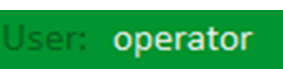

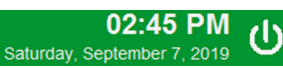
If a monitor with a different aspect ratio is used within the control system a new master page can be created. The new master page should retain the size or the aspect ratio of the existing panes as this allows the existing panes to be displayed in the new master page. Otherwise a new pane needs to be developed to support the monitor aspect ratio.

Header Bar

This section defines details of each pane available in the Master page.

All master pages come with a header bar. It includes a set of commands that an operator may require regular access to during runtime.

Icons	Common tools	Description
	Header Logo	Display EcoStruxure Process Expert logo.
	Home	Displays the home page. The home page is specified in the project Navigation menu. The highest Level 1 entry of the menu sets the context for the workspace and the target page field for the entry identifies the home page.
	Page Navigation	Back, Forward and Up page navigation commands. Back and Forward allows you to move between pages in the current browse history or stack. Up allows you to move up a level in a set of hierarchical pages.

Icons	Common tools	Description
		In multi-monitor systems, the stack is per monitor and not linked. When a log out occurs, the stack is deleted.
	Trends	Opens the trend summary page for the operational area, allowing you to access predefined trends (refer to Process Analyst Page, page 56).
	Active Alarms	Displays alarms that are unacknowledged, or acknowledged and still in an alarm state (refer to Alarm Pages, page 51).
	Historical Events	Displays a historical log of alarms and operator activity (refer to Alarm Pages, page 51).
	Shelved Alarms	Lists alarms that are temporarily disabled (refer to Alarm Pages, page 51).
	Hardware Alarm	Displays a list of hardware alarms that are unacknowledged, or acknowledged and still in alarm state (refer to Alarm Pages, page 51). At runtime a flashing circle underneath the hardware alarm icon indicates there are unacknowledged hardware alarms.
	Maintenance alarm	Displays asset maintenance alarms that are unacknowledged, or acknowledged and still in an alarm state (refer to Alarm Pages, page 51).
	Page Name and Menu breadcrumb	Title of the page displayed in process display area. If page is not in the navigation menu, the message Page does not exist in Navigation menu will be displayed. Underneath the title of the page the menu breadcrumb is displayed.
	User	Name of logged in user. For example, Operator is the current logged in user.
	Login	Menu with the options log in and log out, allows you to log in with selected language and Log out to access additional functionality according to user group or groups you belong to.
	Tools	This control allows you to toggle the label of asset genie and open Parameter set management. Select the relevant option from the menu.
	Date/Time and Shutdown	Current date and time of session plus shutdown command.

Navigation

The operator can navigate the process via a tree control displayed in the navigation pane or through adhoc navigation links based on the information in the control system. The navigation can be defined to either change to a graphics page

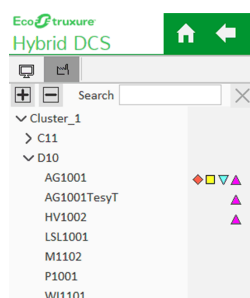
or to set the context of the workspace to a specific asset and have the panes of the workspace adjust to display the best available content.

The navigation tree provides two modes of operation.

Process Navigation

Using the process model, the user sees a view of the process as defined by the EcoStruxure Process Expert. If two assets are defined within EcoStruxure Process Expert with the same name, they may appear as a single asset in the hierarchy. Selecting an asset or a folder in this hierarchy will set the context to this asset. The control system will use the `display` configuration defined in the asset to display the process page, overview page and faceplate for the asset in the respective panes.

The following figure shows an example of the process tree navigation:



If there is any alarm active in any asset, then it is displayed on the right-hand side of the tree with the respective alarm severity symbol. If the node has multiple assets under it and it is not expanded, then this node also displays alarms of assets under it.

By default, the process navigation tree does not display cluster as a node. It can be enabled by adding and setting the below parameter to "1".

Section Name	Name	Description
GPL.Display.Treeview	Cluster	0= Cluster is not displayed in Equipment tree. 1= Cluster is displayed in Equipment tree.

Page Navigation

Using the page hierarchy model, you can refine the runtime project to define a page navigation hierarchy under menu configuration based on the graphics pages that exist within the control system. When the user selects a page with this system, it will update the process window to display the selected graphic page and reset the context, so the other pane will update with default page defined for that pane.

The following figure below shows the Page Navigation configuration in Supervision project refinement:

System_1 : Project 'DemoProcess' Refine Offline

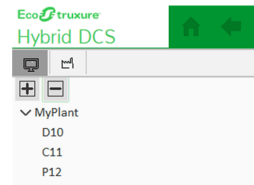
Row	Page	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Menu Command	Target Page	Ct	Order
1	Navigation	MyPlant						Navigation_Showtargetpage()	Overview		
2	Navigation	MyPlant	D10					Navigation_Showtargetpage()	D10		1
3	Navigation	MyPlant	C11					Navigation_Showtargetpage()	C11		2
4	Navigation	MyPlant	P12					Navigation_Showtargetpage()	P12		3

It is necessary to use Navigation as Page name for Page Tree to appear. If a home page is required to be configured it should be configured for Level1 entry as Target Page. See the process tree navigation, where **Overview** is configured as Home page for **MyPlant**. If a home page is configured, then home button in header bar will be available for selection.

This figure shows the home button available in header bar:



The following figure shows an example of the page navigation tree:



Navigation is also possible based on the context of the information being displayed. The alarm list allows the operator to set the context to the asset with the alarm. Doing this will update the process window and display the faceplate for the asset in alarm state. Interlocks support navigation as well. If the initial name of the interlock is the name of the asset, the user can navigate to the source of the interlock condition.

Alarm Banner

The alarm banner displays the top four active alarms. It also allows you to acknowledge, navigate and shelf selected alarm. It does not display alarm for an asset which is in maintenance mode and the asset maintenance alarm.



The following figure shows an example of the alarm banner:

OnDate	OnTime	Asset	Name	Description	Cause
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure1	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure2	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure3	
9/9/2019	10:41:19 AM	HV1002	Load Oxidant in D10	Confirmation Failure	

Alarm Pages

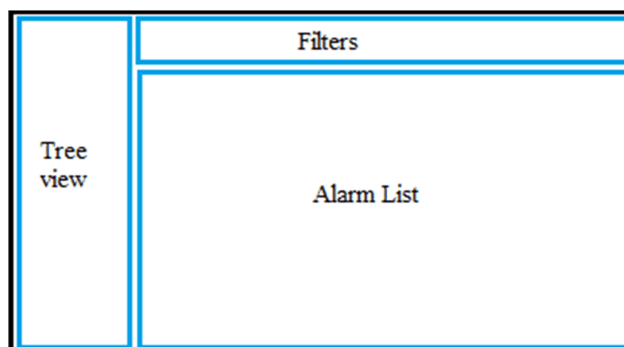
All master pages have a set of default alarm pages that are accessible from the runtime Header Bar. Select one of the buttons described in the table below to display an alarm page in the page content area (refer to [Master page layout](#), page 47).

Icons	Common tools	Description
	Active alarms	Alarms that are unacknowledged or acknowledged and still in an alarm state. This list does not display the alarm of an asset which is in maintenance mode.
	Historical events	A historical log of alarms and operator activity.
	Shelled alarms	Alarms that are temporarily shelved/disabled.

Icons	Common tools	Description
	Hardware alarms	Hardware alarms that are either unacknowledged or acknowledged and still in alarm state.
	Active maintenance alarms	Asset maintenance Alarms that are unacknowledged or acknowledged and still in an alarm state.

Apart from the Hardware Alarms page which displays a basic alarm list, these pages share a common layout that features the following sections:

The following figure shows the alarm page layout:



Alarm List

The content of the alarms list is determined by the following:

- The filters that are currently enabled (refer to the [Filters](#), page 54).
- Selections in the [Tree View](#). The list will only include alarms associated with the selected items (refer to the [Tree View](#), page 53).
- Customized filters in the Active alarm list/maintenance alarm list page from [Filter by](#) context menu. (refer to the [Customized Filters](#), page 55)

The default set of columns are determined by the type of alarm page that is displayed and the default format specified for it (see list below for default format for each list).

Section Name	Name	Available column	Applicable list
FORMAT	DefaultGPLAIActiveAlarm_HD1080	Priority and State, OnDate, OnTime, Asset, Name, Desc, Cause, Custom1, Custom2, Custom3, Group	Active alarms
	DefaultGPLAIISOE_HD1080	Priority and State, Date, Time, Asset, Message, Desc, UserName, UserLocation	Historical events
	DefaultGPLAIShelvedAlarm_HD1080	Priority and State, OnDate, OnTime, Asset, Name, Desc, DisableEndDate, DisableEndTime, DisableComment	Shelved alarms
	DefaultGPLAIMaintActiveAlarm_HD1080	Priority and State, OnDate, OnTime, Custom7, Name, Desc, Cause, Custom1, Custom2, Custom3	Active maintenance alarms

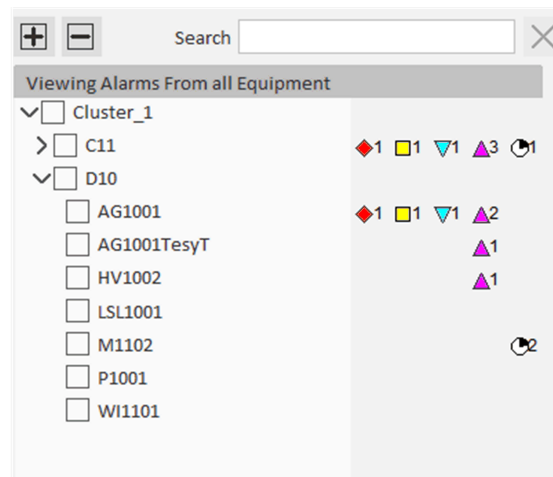
The below mentioned additional columns are available in alarm page using custom field configured by asset configuration:

Column Name	Description
Priority and State	Displays priority symbol and state of the alarm.
OnDate	On date of the alarm.
OnTime	On time of the alarm.

Column Name	Description
Asset	Displays the name configured for the asset in the <i>Instance Editor</i> . It is stored in the <i>custom7</i> field of the alarm tag.
Name	Displays the description of the asset configured in instance editor. It is stored in <i>Alarm Name</i> field of the alarm tag.
Description	Displays the description of the alarm.
Cause	Displays Interlocks/Failures condition which makes the device inoperable. It is stored in the <i>custom8</i> field of the alarm tag.
Custom1/Custom2/Custom3	Displays the custom value configured on asset configuration. This field allows filtering alarms. See customized filter in <i>Alarm</i> page for details.
Group	The collection of alarms to which the alarm belongs. Group is configured under setting section in asset configuration. It is stored in the <i>Custom4</i> field of the alarm tag.
Message	Displays the message associated with event/alarm.
UserName	Displays name of the user who has triggered the event. If it is triggered by system, it displays <i>System</i> .
UserLocation	Displays the Operation client IP address from where the event is triggered.
Shelved End Date	Displays the date when alarm is enabled.
Shelved End Time	Displays the time when alarm is enabled.
Shelved Comment	Displays the comment entered while disabling the alarm.

Tree View

The tree view provides an example of the equipment hierarchy of a project. You can use it to filter the displayed list of alarms.



Each item in the tree view includes:

- Check box allows you to select one or more items in the tree.
- Alarms count indicate the number of associated alarms in the top four alarm priorities, as well as the number of shelved alarms.
- Alarm counts that indicate the number of associated alarms in the top four alarm priorities, as well as the number of shelved alarms.
- Buttons to expand (+) and collapse (-) the tree.
- Search box that you can use to filter the displayed list of assets.

The alarms list will be filtered to display alarms associated with the selected asset. Search box allows you to search an asset in the tree.

Filters

Alarms page allows you to filter the alarms list according to priority, state, acknowledgment and customized filter. The Current Filter description indicates if any filters are currently active for the alarms list.

Priority Filters

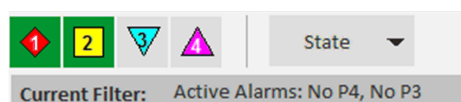
The alarm priority buttons allow you to filter out alarms included in the top three alarm priorities. By default, the three buttons are selected (indicated by a colored background).

This figure shows the alarm filter for priority:



If you toggle a button to off state, the alarms assigned to the alarm priority will be removed from the list. The current filter description will indicate that the P1, P2 or P3 alarms are no longer displayed.

This figure shows the current filter with no Priority3 and Priority4 alarms:



The Historical Events page also features a button that allows you to filter out shelved alarms.

This figure shows the shelved alarm indication:



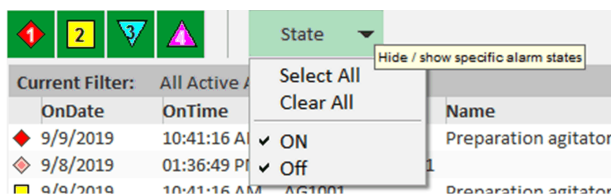
State Filters

The State filter button allows you to filter out alarms that are in a specific state.



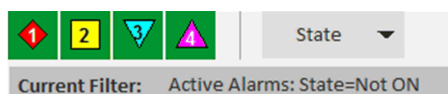
NOTE: This button is not available on the Shelved Alarms page. When you click on the button, a list of alarm states appears.

This figure shows the state filter:



Select/ Deselect the alarm states that you would like to include/ exclude from the alarm list. The Current Filter description will indicate if any of the states are not selected.

This figure shows the state filter description:



Acknowledgment Filters

The Historical Events page also features buttons that allows you to filter Acknowledged and Unacknowledged alarms.

Acknowledged

Unacknowledged

By default, both buttons are selected (indicated by a colored background). If you toggle a button to off state, the associated alarms will be removed from the list.

Customized Filters

Alarms on the active alarm page and maintenance alarm page also feature customized alarm filter. These filters can be called by selecting **Filter By** option in the context menu of an active alarm. These filters apply filter on current list based on the selected item value. Below options are available.

This figure shows an example of a customized alarm filter:

OnDate	OnTime	Asset	Name	Description	Cause	Custom1	Custom2	Custom3	Group
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure1	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure2	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure3	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	HV1002	Load Oxidant in D10	Device Failure4	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure5	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure6	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001TesyT	Preparation agitator D10	Communication Failure AL...	Agitator	Motor	Large	Failure	

This field is configured from instance editor of the asset in EcoStruxure Process Expert.

This figure shows an example of the instance configuration for custom alarm filter:

Name	Description	Type
AlarmPriority	Asset Priority	Enum
Location	Physical Location	String
Custom1	Custom 1 (Alarm Filter)	String
Custom2	Custom 2 (Alarm Filter)	String
Custom3	Custom 3 (Alarm Filter)	String
Custom4	Custom 4 (User Defined)	String
Custom5	Custom 5 (User Defined)	String
Custom6	Custom 6 (User Defined)	String
GraphicProcess	Process Page Name	String
GraphicOverview	Other Page Names (Optional List)	String

This figure shows an example of the customized alarm filter on Custom2 field:

OnDate	OnTime	Asset	Name	Description	Cause	Custom1	Custom2	Custom3	Group
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure1	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure2	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure3	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure4	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure5	Agitator	Motor	Large	Failure	
9/9/2019	10:41:16 AM	AG1001TesyT	Preparation agitator D10	Communication Failure AL...	Agitator	Motor	Large	Failure	

ClearAll option, clears all the customized filters applied on the page.

Name of the Custom1, Custom2, and Custom3 columns can be customized by adding below parameters (in Supervision project under Setup) and providing required name in value column.

Section name	Name	Value
AlarmHeading	Custom1	Example: Location
AlarmHeading	Custom1	Example: Type
AlarmHeading	Custom1	Example: Size

This figure shows the alarm page with customized column names and list of filters:

Current Filter: All Active Alarms

OnDate	OnTime	Asset	Name	Description	Cause	Location	Type	Size	Group
9/9/2019	10:41:16 AM	AG1001	Preparation agitator D10	Device Failure1		Agitator	Motor	Large	Failure
9/9/2019	10:41:16 AM	AG1001	Acknowledge	Device Failure2		Agitator	Motor	Large	Failure
9/9/2019	10:41:16 AM	AG1001	Pr	Device Failure3		Agitator	Motor	Large	Failure
9/9/2019	10:41:19 AM	HV1002	LC	Confirmation Failure		Agitator	Valve	Small	Failure
9/9/2019	10:41:16 AM	AG1001	Pr	Device Failure		Agitator	Motor	Large	Failure
9/9/2019	10:41:16 AM	AG1001	Pr	Device Failure		Agitator	Motor	Large	Failure
9/9/2019	10:41:16 AM	AG1001TesyT	Pr	Device Failure		Agitator	Motor	Large	Failure

Context Menu Options: Filter By, Description, Location, Type, Size, Group

Process Analyst Page

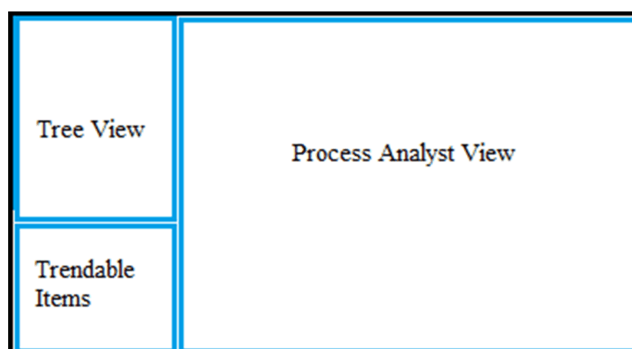
Process Analyst page is accessible from the runtime Header Bar. Click on the icon below to display a trend page in the Process display/ System page pane on the master page. (Refer to Master page layout, page 47).



Process Analyst Navigation Button

The process analyst page allows you to analyze specific trends and alarms for each asset. The layout of the process analyst features the following sections.

This figure shows the Process Analyst layout:



Behaviors that you will observe when you navigate between pages include:

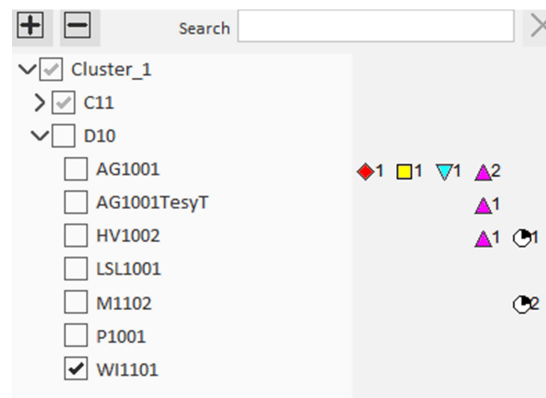
- Navigating to a different page within the Workspace will not affect the trend page. Your pens, currently selected object in the Object Tree and any text entered in the Search box will be retained.
- Navigating to a different page and selecting a different equipment and then navigating back to the trend page will clear the entry in the Search box and also select the equipment in the equipment tree.
- Navigating to a different page and selecting a different equipment and clearing that selection before navigating to the trend page will retain your configuration on the trend page.

NOTE: You can launch the process analyst page from faceplate trend tab. Clicking on the button on faceplate trend tab will launch the process analyst page with the trend and alarm of the asset.

Tree View

The tree view provides the equipment hierarchy of a project.

This figure shows the tree view layout in Process Analyst page:



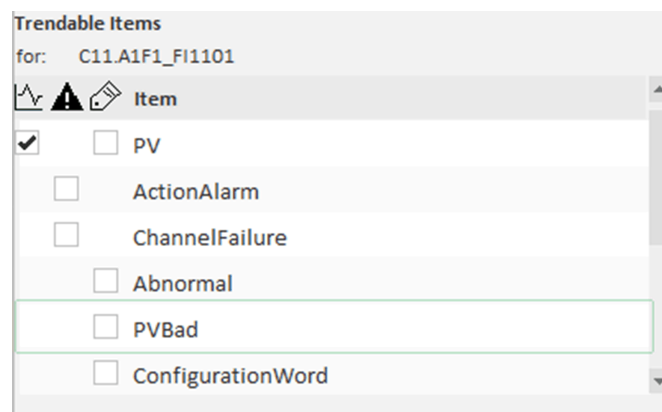
The tree view displays:

- Check boxes for assets that have an active trend.
- Gray check boxes for assets whose children have an active trend.
- Alarm counts that indicate the number of associated alarms in the top four alarm priorities, as well as the number of shelved alarms.
- Buttons to expand (+) and collapse (-) the tree.
- Search box that you can use to filter the displayed list of assets.

Trendable Items

The Trendable items section lists the items associated with each piece of equipment that can be trended.

This figure shows the Trendable Items in process analyst page:



The following icons indicate the type of an item (tag), which can be an Alarm, Trend or Instant (Variable). A check box is displayed for each applicable type.

	Trend Items
	Alarm Items
	Instant Items

Select the required check boxes to add them to the Process analyst chart. Up to 16 trends can be added. When the limit is reached, the check boxes are hidden.

Process Analyst view

Displays the process analyst view of the selected item of the selected equipment. In addition to standard tool on main toolbar, an additional tool is available to navigate in model for selected pen.

The following icon displays the `find in model` tool in process analyst page:



Genie

Overview

This section details the graphical elements provided with the library to display the elements of the control system. All elements (where possible) are drawn using vector objects and can be scaled to match the required size of the screens.

All the symbols are sized to be used with Full HD screen resolution. If your system contains a mix of monitor resolutions and sizes, it may be required to develop screens in a resolution which scales correctly to the monitors available.

Symbols Provided

The symbols represent the base graphical objects displayed on the screen (without additional animation).

Digital Inputs & Outputs (`gpl_digitalbase` genie library)

- Circle
- Arrow
- Electricswitch
- Siren

Motors (`gpl_motorbase` genie library)

- Blower
- Motor
- Motorb
- Pump
- Screwpump
- Childmotor
- Direction

Valves (`gpl_valvebase` genie library)

- 2way
- 2way_motorized
- 3way_down
- 3way_leftdown
- Butterfly
- Controlvalvehead
- Head
- Penstock
- Rotatory

Analog Inputs & Outputs (`gpl_meterbase` genie library)

- Flow_128_v
- Level_128_v
- Pressure_128_v
- Deviation_128_v

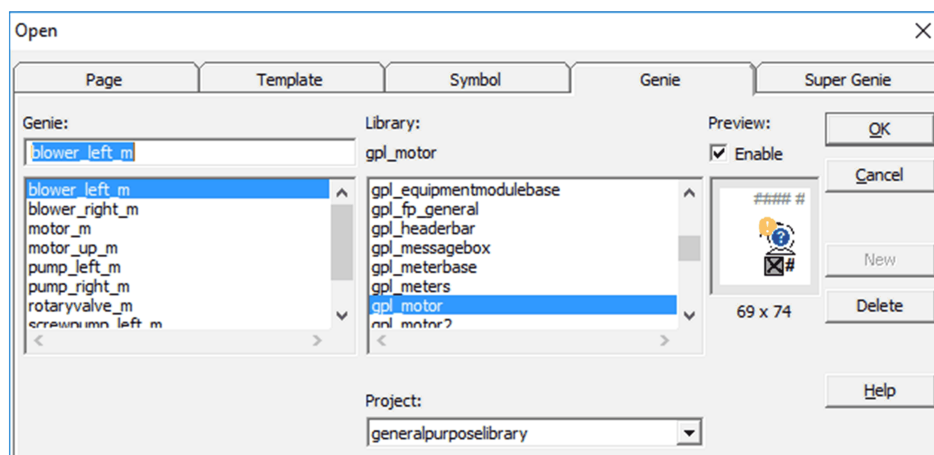
- Analyzer_128_v
- Meter_trend
- Misc_128_v
- Misc_128_h
- Range_optimal_dev_vert
- Range_optimal_horiz
- Range_optimal_vert
- Setpoint_horiz_ao
- Setpoint_horiz
- Setpoint_vert
- Descriptor
- Alarm_limits_dev_vert
- Alarm_limits_horiz
- Alarm_limits_vert

Sequence (gpl_sequencebase genie library)

- Seqbuttons
- Seqmanstepitem
- Sequencecontrol
- Transstep

Genie Location

All the genies are located within the `generalpurposelibrary` project.





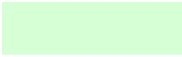


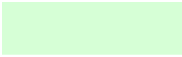

Genie Elements

A genie is composed of the graphic symbol as well as a set of additional genies which provide consistent functionality.

Naming Conventions









Genies are named with respect to the output of the asset. For example, `Pump_Right` pumps water to the right of the page (input on the left). `Blower_Left` blows air to the left. For valves, the first direction is the valve inlet and the second the outlet. So `3way_vert_UpRight` valve genie takes water from above and pumps it to the right.

Equipment Status

Asset	Equipment status	Color (default)
Motors	Stopped	White
	Starting	White
	Running	 Dark Grey
	Unknown	 Black
Valves	Closed	White
	Opening	 Light Green
	Open	 Dark Grey
	Stop	 Grey
	Closing	 Light Green
	Unknown position	 Black

Equipment State

The equipment state represents the state of the control module output. The equipment state genie is a square box which displays the state of the output.

Equipment state	Description
	Off
	On
	Interlocked
	Interlocked On
	Failed state
	Out of service / Maintenance mode
	Active transition
	Passive transition

The symbol displayed is determined by a function for each type of control module. This function is defined within the template of the asset.

Abnormal Owner

The owner of the asset is only displayed in the genie when it is defined as abnormal by the engineer. The value displayed is a text string. The letter displayed is defined by the enumeration "":


Item	Value	Enum	Symbol
Owner	1	Operator	O
	2	Program	P
	3	Cascade	C
	4	Local Panel	L
	5	Zero	Z
	11	External (O)	E
	12	External (P)	E
	13	External (C)	E
	14	External (LP)	E
	15	External (Z)	E

The string displayed is determined by a function for each type of control module. The function is defined within the template of the asset.

Abnormal Mode/Operator Action

The symbol for an abnormal mode or operator action is an orange circle with an exclamation mark (!).

The Abnormal mode or operator action genies are available in two sizes (medium and small) in `gpl_sabase` genie library in `generalpurposelibrary`:

Genie symbol	Genie name	Description
	Abnormal	Abnormal genie

The symbol displayed is determined by a function for each control module. The function is defined within the template of the asset.

Label

The label for a control module displays the asset name. The asset name is stored as the display name of the asset. The asset name can be displayed justified to the left, right or center. The genie used to display the label is `Label_centre`.

The operator can select to display (or hide) the labels for a screen from the `Show/Hide Labels` button available in the options section of the workspace.

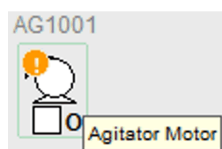


NOTE: If a label associated with a genie appears in the wrong position (overlaps other graphics), it can be permanently disabled via a parameter on the genie.

A separate label genie is available, so a new label can be positioned in the best location. The individual label will hide/show and be updated like the integrated label.

Selected Asset

When an asset is selected in the runtime, it is highlighted by a green box. The box defines the size of the symbol on the screen. The genie used to display the green box is *Selection* and it is available in *gpl_sabase* genie library in *generalpurposelibrary*.



Alarm Border

In the group properties of the selected genie, Alarm indicator has been enabled so that based on the Alarm priority and category, the Alarms can be displayed based on their criticality:

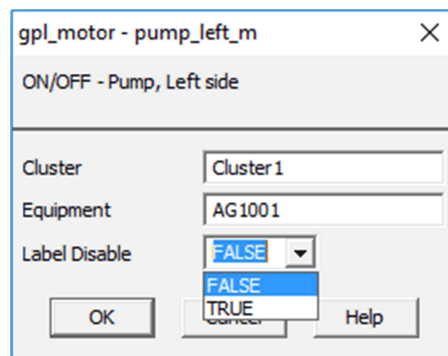
Representation	Description
	Critical alarm On and Unacknowledged
	Critical alarm Off and Unacknowledged
	High alarm On and Unacknowledged
	High alarm Off and Unacknowledged
	Medium alarm On and Unacknowledged
	Medium alarm Off and Unacknowledged
	Low alarm On and Unacknowledged
	Low alarm Off and Unacknowledged

Genie Parameters

Equipment

The process model defined by the engineer is used to create an equipment name in the supervision which combines the asset location and name (*MyArea.MySystem.MyProcess.AG1001*). The equipment name is generated by EcoStruxure Process Expert and may represent a short version of some names used within EcoStruxure Process Expert to ensure system operation.

When graphics pages are built within EcoStruxure Process Expert, their equipment name is automatically assigned when the graphics page is saved. The equipment name is also updated if a change is required in the control system over time.



Cluster

This defines the cluster name of the equipment.

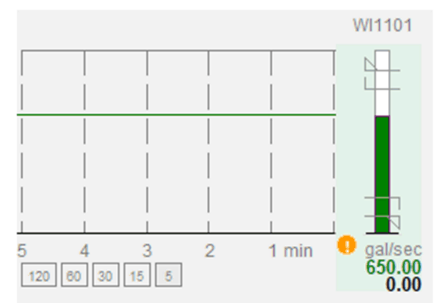
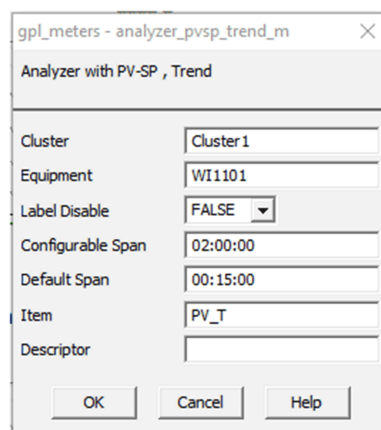
Label Disable

This selection box is used to remove a label which is obstructing other elements on the graphics screen. The alternative label genie should be used to display a label for the asset in a better location.

Item (Analog with Trend)

Item name of the equipment which has to be plotted on the trend.

NOTE: If item is not specified in the genie parameters, the pens will be added for eight items of the equipment. For example, in case of analog input, the trend pens of PV, SP, LowLowLimit, LowLimit, HighLimit, HighHighLimit, ORHigh, and ORlow will be displayed in the trend panel of the genie.



Configurable Span (Analog with Trend)

In the **Trend** genies of control modules, the trend span can be configured from the genie parameters. The default value of the Configured span is given as 02:00:00 representing 120 minutes.

Default Span (Analog with Trend)

In the Trend genies of control modules, the default trend span can be configured from the genie parameters. The default value of the span is given as 00:15:00 representing 15 minutes.

Bar Graph Description (Bargraphs)

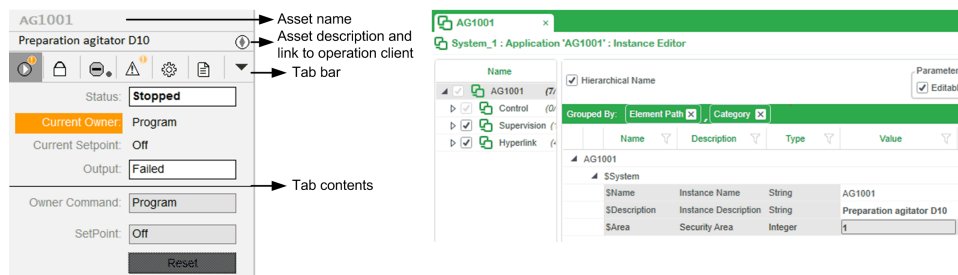
The Miscellaneous, Deviation, and Analyzer meters can all be used to show many different types of readings. Therefore, The short text (up to 3 characters) is provided to help identify in the reading (example, O2 for Oxygen content, SG for density, Vis for viscosity, Trq for torque, etc.)

Faceplates

Overview

A faceplate within the library is a small workspace. The button bar on the top is to select the content associated with the asset. Each page is created independently, and engineers can control the selection of data and the order in which the tabs are displayed. If projects have additional information they wish to display, adding a tab to the faceplate is a simple method to extend the capabilities of the system.

This figure shows an example of the Faceplate layout:



To open multiple faceplates of same asset, create the following parameter:

Section name	Name	Value
GPL. Display	MultipleFaceplate	1

To resize faceplate based on the page from which faceplate is opened, create the following parameter:

Section name	Name	Value
GPL. Display	FaceplateResizable	1

The faceplate tabs displayed for each asset are defined as pages in menu configuration under visualization. The asset type with suffix (`_Tabs`) is the page name in the menu configuration (for example, `Motor_Tabs` for Motor template) and the available tabs are defined as Level 1 entry. The symbol, name and visibility conditions defined in the page navigation configuration are evaluated so tabs can be made to appear and disappear based on properties of the asset.

The following configuration must be done for each tab in menu configuration. All tabs available in library are pre-configured.

Column name	Description	Example
Page	Name of page for each asset type. This name needs to be same for all the tabs of a same asset. Same name to be configured in the Custom7 field of the equipment.	Motor_Tabs
Level1	Name of the tab	Operator
Target Page	Name of the page which will open on clicking tabs in runtime.	IFP_OnOffDevice_Operator
Order	Tab order sequence in faceplate. In library this order is defined using a Label to make it consistent across all the object	TB_OPER
Symbol	Icon which appears on the tab. A genie is displayed on the tab. Genie name to be specified with the library name. Example, Library Name.Genie Name.	gpl_tabicons.operator_m











Column name	Description	Example
HiddenWhen	This field allows a tab to be optional in the runtime. When the expression in this field is True then tab will not appear.	Not (GPL_OptionCheck (Interlock)) In this example, if the asset is not having.
Custom1	1= Tab cannot be close in runtime	Default in library all tabs cannot be close
Custom2	1= Hide icon on the tab	Default in library all tabs shows icon
Custom3	1= Hide Tab label	Default in library all tab do not show label

The library provides an **Operator** tab for each asset type. This is the only faceplate page which is explicitly defined in the library per asset. The remainder of the faceplate pages are generic pages which react to the asset which is the context of the faceplate.

NOTE: An user trying to open a faceplate of an asset from another area that is not accessible, will be able to open the asset but the user will not be able to operate or see alarm/ trends of that asset in the faceplate.

Available Tabs


The table describes the functions that are available in each tab:

Tab name	Icon	Description
Operator		Module control
Failure		inoperable asset conditions (grouped if required)
Alarm		Active or all alarms
Interlocks		Interlock conditions (grouped if required)
Events		State change and operator Action Log
Engineering		Engineering parameters
Measure		Asset performance measures
State machine		Sequence state parameters
Output parameter		Sequence output parameters
Autotune		PID autotune parameters

The operator of most tabs is explained in association with their equivalent configuration. Some blocks are more generic and can receive data from multiple sources.

NOTE: The order of the tabs in a faceplate can be adapted to suit the application without impact on functionality. The tabs order shown in documentation images may differ from that displayed in runtime to the operator.

Faceplate Icon

Name	Icon	Description
Link		To navigate to linked asset.

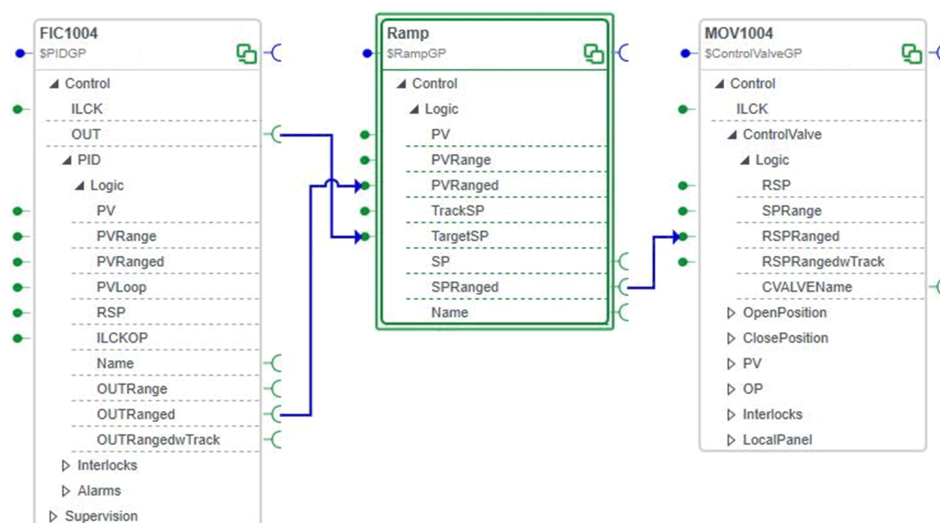
Operator Tab

The **Operator** tab is a graphics page animated based on the asset selected as context. On this page there is no need to define the name of the equipment. The library provides a set of genies to display the values on the **Operator** tab.

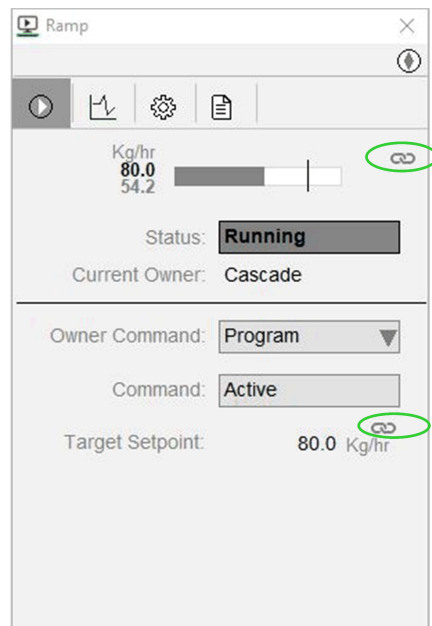
The genies adapt their display to the properties of the asset which are being displayed.

Navigation Link

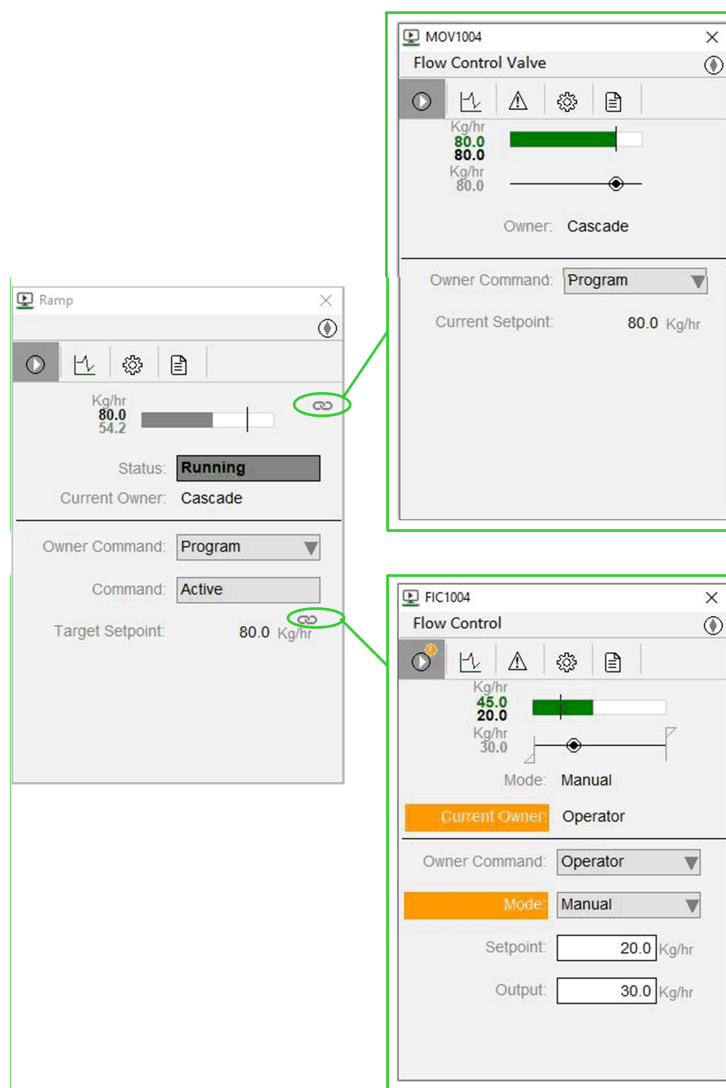
The GPL offers the feature to navigate between faceplates of linked equipment in EcoStruxure Process Expert. The configuration needed to achieve this is detailed below and \$RampGP template is used as an example.



The link icon will appear on the **Ramp** instance faceplate if linked equipment has a defined faceplate.



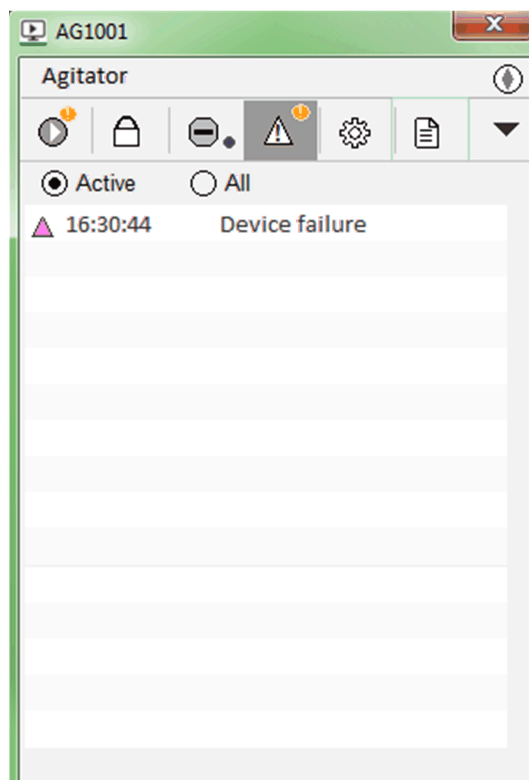
Clicking the link icon navigates to the faceplate of the linked equipment.



Alarm Tab

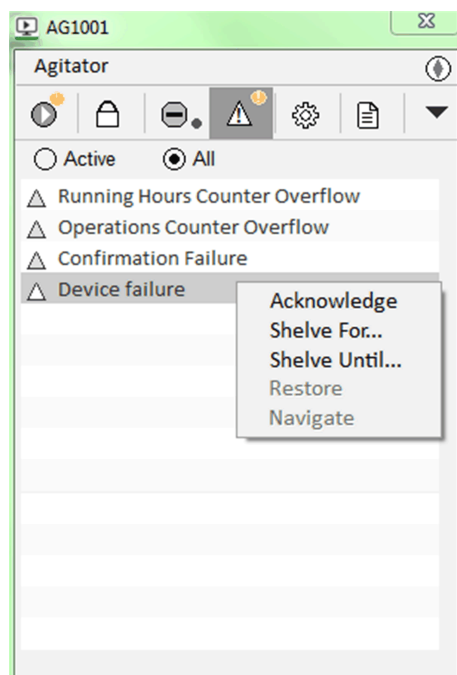
The alarm tab lets the user to see the active alarms or all configured alarms. When the user navigates to the **Alarm** tab, it shows the active alarms.

This figure shows an example of the **Alarm** tab:



Clicking the **All** button displays all the configured alarms in the asset.

This figure shows an example of all the configured alarms in **Alarm** tab:



From the **Alarm** tab users can acknowledge and shelve/ restore the alarms.

Create the following parameter in the Plant SCADA parameter section, so that the alarm page can be configured to display the alarms from the children of this equipment.

Section: GPL.Alarm

Name: EnableChildSelection

Value: TRUE

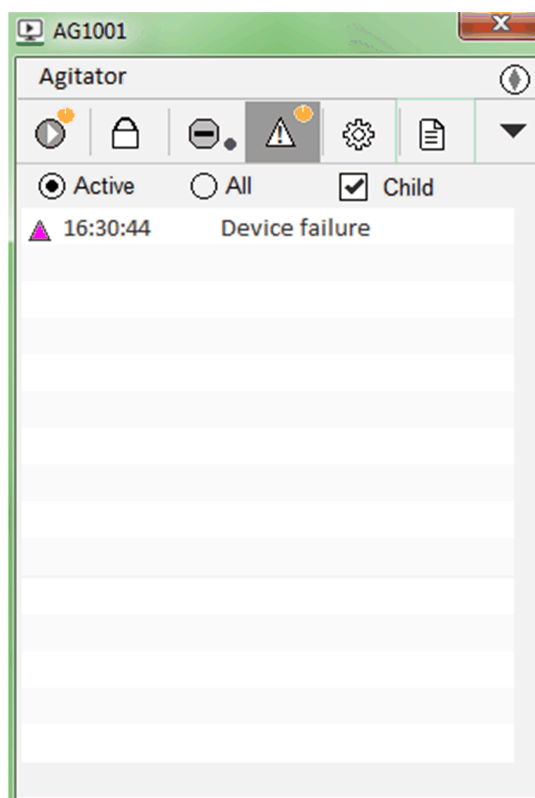
Create the following parameter in the Plant SCADA parameter section, so that the alarm page includes the referenced equipment.

Section: GPL.Alarm

Name: IncludeReferences

Value:1

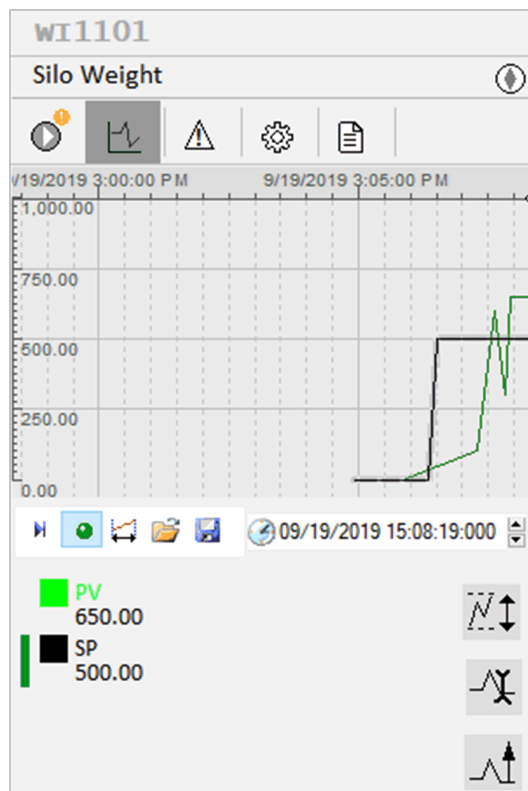
This figure shows an example of the **Alarm** tab with child alarm option enabled:



Trend Tab

The trend objects provide the ability to embed trend information in display pages. The trend object provides a basic trending capability for any instrument.

By default, the PV, SP, OP will be displayed for the process and devices depending on the trend item configured.



To delete or add the trend items to be displayed on the trend, the following parameters must be configured.

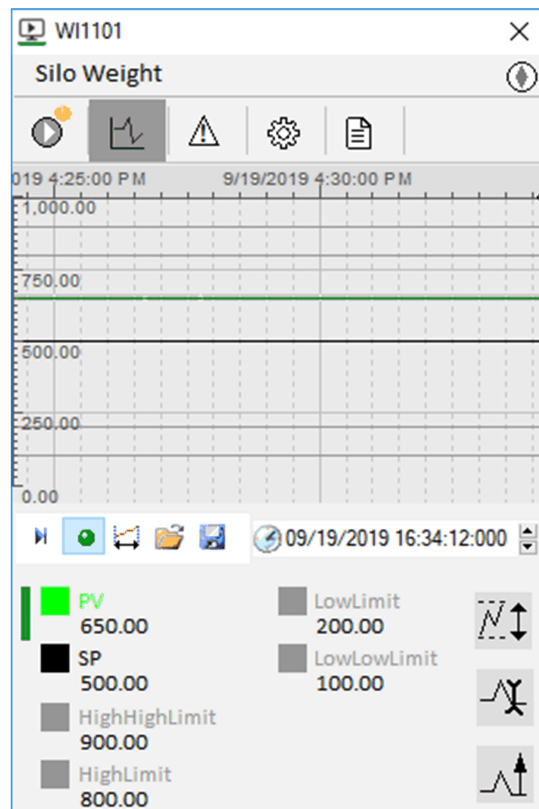
Section name - FpTrendItems

Parameter - Equipment Name (Example: WI1101)

Value - Item name with comma separated

(Example: PV,SP,OP,HighLimit,HighHighLimit,LowLimit,LowLowLimit)

Row	Section Name	Name	Value
13	FpTrendItems	WI1101	PV,SP,HighHighLimit,HighLimit,LowLimit,LowLowLimit



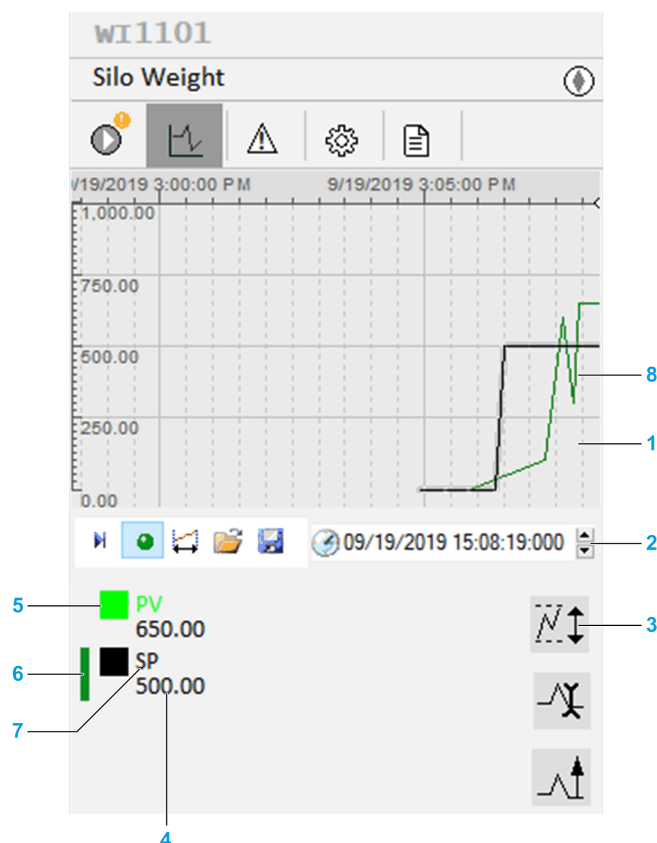
By default, the color of PV, SP, OP will be configured as Green, Black, Grey. If user wants to customize the color of Trend then the value of required item in the FpTrendItems section must be changed and mentioned in the required name which is available from the RGB_Color parameters (as mentioned in the Picture 2).

Picture 1

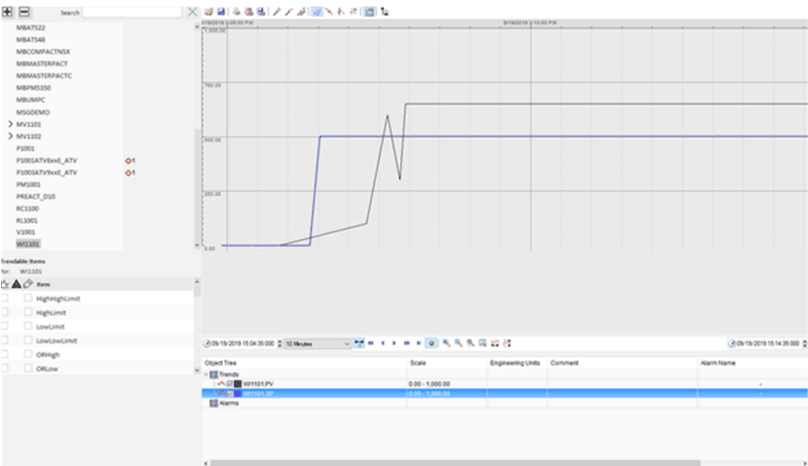

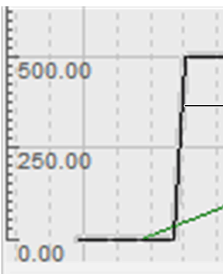
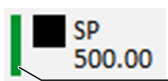
Row	Section Name	Name	Value
1	FpTrendColor	ORHigh	Alto
2	FpTrendColor	PV	Green
3	FpTrendColor	SP	Black
4	FpTrendColor	OP	Grey

Picture 2

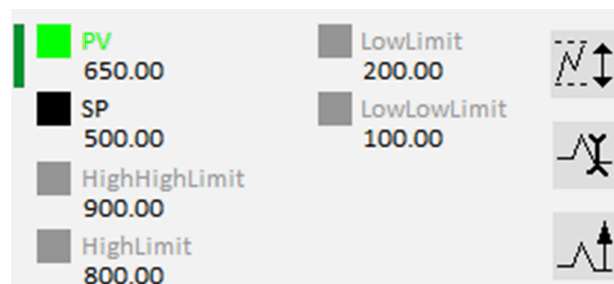
Row	Section	Name	Value
11	RGB_Color	White	255,255,255
12	RGB_Color	Black	0,0,0
13	RGB_Color	Grey	128,128,128
14	RGB_Color	Melrose	179,164,255
15	RGB_Color	Yourpink	255,204,204
16	RGB_Color	BrightTurquoise	12,198,231
17	RGB_Color	JapaneseLaurel	0,125,0
18	RGB_Color	Pinegreen	0,125,109



Sl. No.	Description
1	<p>Edit Vertical Scale - Allows operators to edit the vertical scale of a selected analog pen to display more appropriate values</p>
2	<p>Toggle Cursor - A cursor enables an operator to determine the value of a pen at a given point in time by dragging the cursor to the specific point on the pen line. A cursor label is used to display the value.</p>

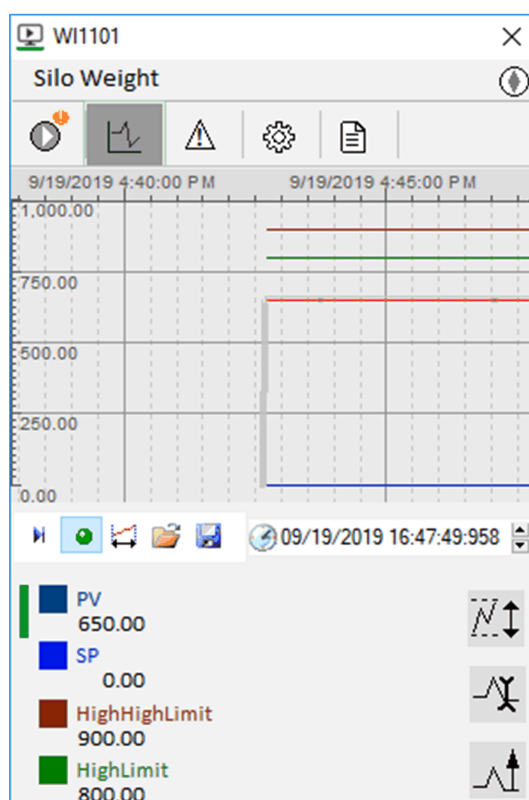
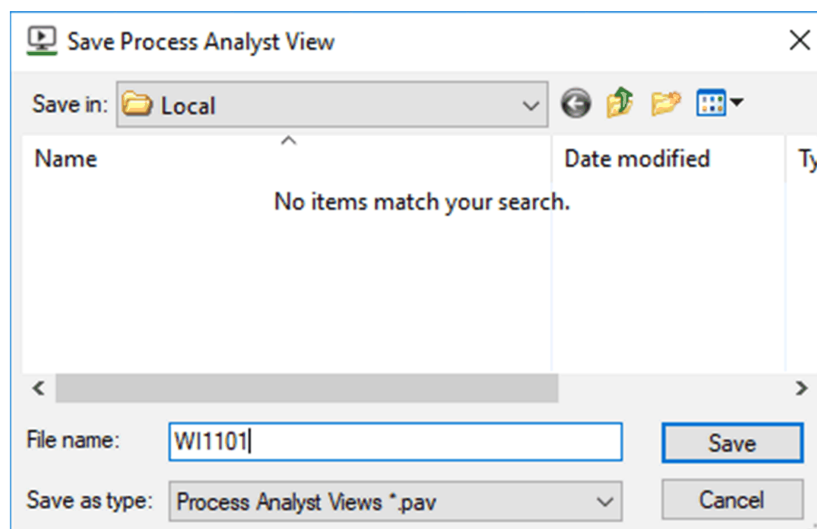
Sl. No.	Description
3	<p>Display the Trend Page - Clicking this option will open the Trend page where operator can configure or visualize the Trends in the bigger screen.</p> 
4	<p>Item Value - Displays the current item value, if cursor is shown then displays the value of item to the cursor position.</p>
5	<p>Item Color - Represents the color of the Item displayed.</p> 
6	<p>Item Highlighted - By Clicking on the object, a pane pen will be highlighted in the object view and will represent the object in rectangle shape.</p>   <p>Indicates Selection</p>
7	<p>Item Name - Displays the trend Item name of the equipment.</p>
8	<p>Trend Pen - Trend items plotted in the object view.</p>

User can add or delete pens as per their requirement. It displays all the trends on the object view but only eight items will be visible in the bottom section.



Operator can add, delete, or change the required color of pen item and save the file in the name of equipment (Instance name) in the **Local** folder of the project. In this case the Trend in the Faceplate will load the same file.

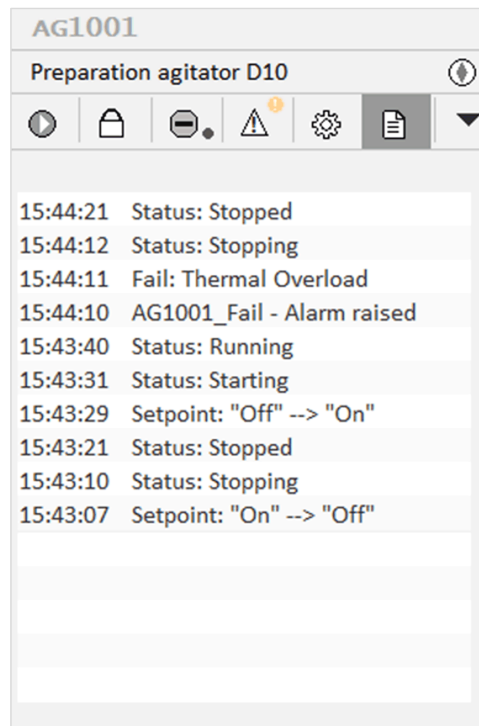
Example: In the following example, the color of the trend is changed and the file saved.



Event Tab

The **Event** tab shows the historical event log of the equipment. For details on event logging refer to the data logging section, page 80.

The following figure shows an example of the **Event** tab.



Engineering Tab

The **Engineering** tab contains groups of parameters the engineer can adjust to control the operation of the asset. Each **Engineering** tab can contain 16 groups of parameters each containing 10 parameters.

To display a variable in the engineering tab the variable tag must have the following properties. Engineering value can be read-only (no security defined) but it is intended for use with `read/write` variables:

Custom 5	Engineering	Required to identify the variable for this tab.
Custom 6	<Group Name>, <Position>	Defines the group and position within the group.
Custom 8	Security	Defines the users who can change the value.
Custom 2	Item Label	Defines the label to display which describes the group.

Measures Tab

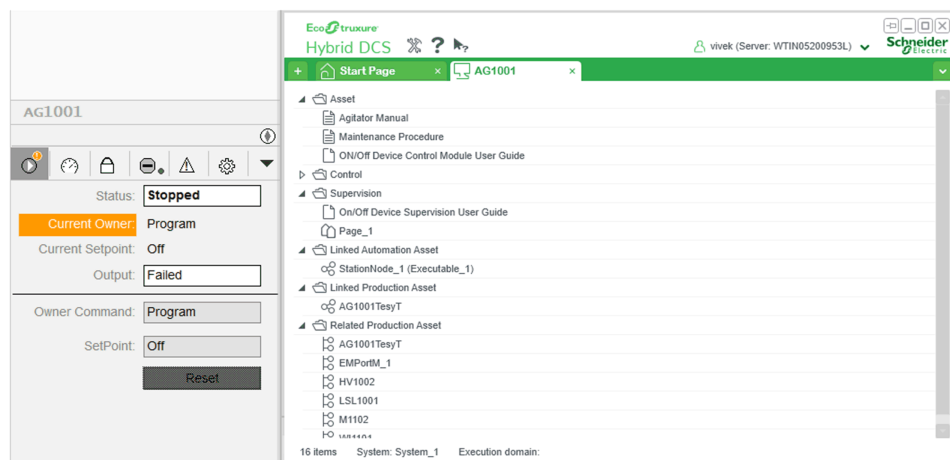
The **Measures** tab measures the performance of assets. The values are displayed with their value and engineering units. Digital values are displayed either as enumerations or are highlighted when **ON**. The **Measures** tab can contain 16 groups of measures each containing 30 variables. The parameters appear in a scroll list if more than 10 parameters are defined.

To display a variable in the **Measures** tab the variable must have the following properties defined in the template:

Custom 5	Measures	Required to identify the variable for this tab.
Custom 6	<Group Name>, <Position>	Defines the group and position within the group.
Custom 2	Item Label	Defines the label to display which describes the group.

Runtime Navigation Services

To display the operation client of EcoStruxure Process Expert select the compass icon (navigation) in the top right-hand corner of every faceplate as shown in the following figure.



Security

Users

EcoStruxure Process Expert recommend the use of Windows Integrated Security to manage user for the control system. Windows users are assigned to groups which are associated to specific roles within the process. Roles such as `MineOperator` or `ProcessOperator` define the same operator role in different areas of the process. Each user can have more than one role and the rights of the user are based on the combination of groups to which they are a member.

NOTE: For system diagnostics it is important that a kernel user be declared for the system. As this user can execute high level functions the password must be defined for each system.

Roles

EcoStruxure Process Expert provides default roles in each area of the process:

Role	Description
Operator	Responsible to operate the process
Maintenance	Responsible to maintain the process equipment
Supervisor	Responsible to accept risk in operations (bypass conditions)
Engineer	Responsible to tune the performance
Administrator	Responsible to manage the control system
Disabled	Not available

All actions within the control system are defined as being available only to users of a specific role.

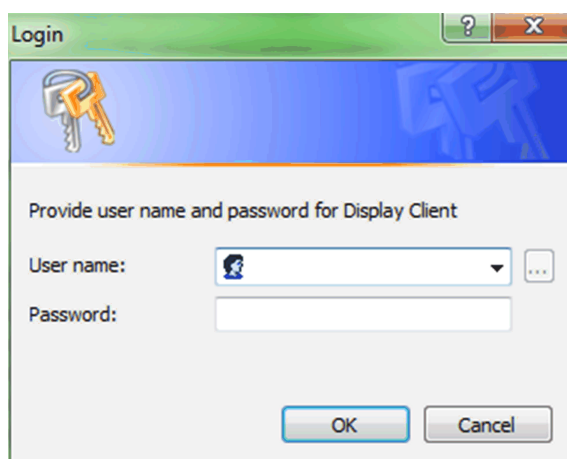
The roles are not hierarchical. It is possible for a user to be an administrator but not be able operate the process. If an operator is also a supervisor they should be added to both roles.

The library provides a pre-defined role for each type of role available. These do not support multiple areas. If the project has multiple areas or wishes to link to different windows groups the roles must be created within a user project.

Areas

Every asset can be assigned to an area (optional). Defining an area allows each user to have different roles within different areas of the process. Areas deny access to resources within that area to user without access rights.

Any system element that is not assigned to an area between 1 and 255 is automatically placed in a default area known as **Area 0**. Every user can view the system elements in **Area 0**, but without the matching privilege will be unable to control them. The default mode set in library asks the user to login before the runtime starts, the login window is as shown in the following figure.



If user needs to modify this mode, set the following parameter in the setup editor. The details of this parameter are available in Plant SCADA help.

Section: `Client`

Name: `Autologinmode`

An area is a number between 1-255. It is not required to start with 1 so it is typical to use the first two digits of the plant asset number to represent the group to simplify entry.

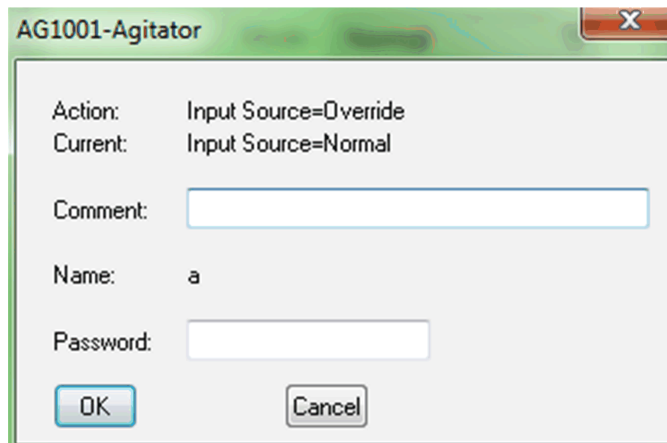
Example. 4361PU1001 is in area 43.

Confirmation

Every action can require a specific role but can also require a confirmation by the same user or by a user with different privilege level. The library allows the user level and the confirmation type to be defined.

Confirmation: Requires the operator to re-enter their password and a comment.

This figure shows the **Confirmation** window:



The image shows a software window titled "AG1001-Agitator" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains the following fields and controls:

- Action:** Input Source=Override
- Current:** Input Source=Normal
- Comment:** A text input field.
- Name:** a
- Password:** A text input field.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

Dual Confirmation: Requires the operator and another user to enter password and comment.

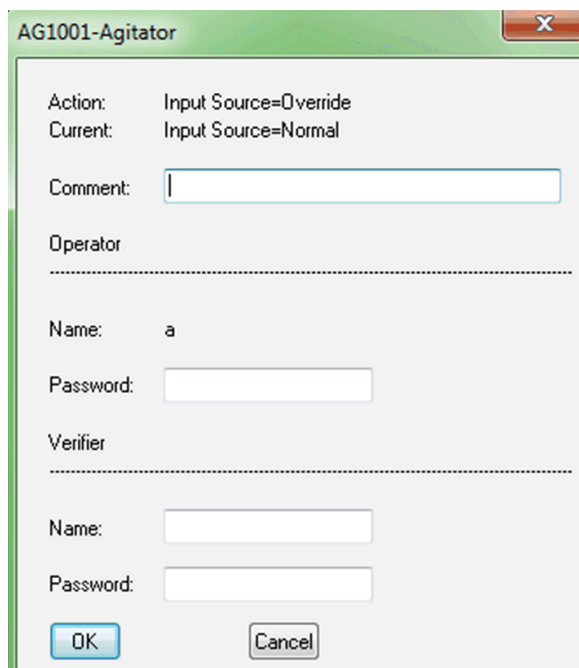
NOTE: While not the default operation it is also possible to enforce entering a comment for each action by creating the following parameter.

Section: `GPL.Command.Confirmation`

Name: Enforce Comment

Valve: 1

This figure shows an example of the dual signature window:



The image shows a software window titled "AG1001-Agitator" with a standard Windows-style title bar. The window contains the following fields and controls:

- Action:** Input Source=Override
- Current:** Input Source=Normal
- Comment:** A text input field.
- Operator:** A section header followed by a dashed line separator.
- Name:** a
- Password:** A text input field.
- Verifier:** A section header followed by a dashed line separator.
- Name:** A text input field.
- Password:** A text input field.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

Alarms

Alarm Category Numbers

Alarm category defines the importance of an alarm.

By default, the following general criteria is applied to assign category numbers to alarms:

Alarm Category	Comment
1001	Critical alarms
2001	High alarms
3001	Medium alarms
4001	Low alarms
5200	Action events
5300	Mode events
5400	State events

NOTE: Alarm category 5300 and 5400 are not applicable for the device templates.





Alarm Priority

Four Situational Awareness (SA) alarm priorities are configured in the GeneralPurposeLibrary project. Each alarm signal is assigned an alarm category, which is related to its alarm priority. For example, if an alarm belongs to an alarm category of 1001, it is assigned an alarm priority of 1 by default. You can change the priority that is associated with each alarm category to adapt the behavior to the specific requirements of your project.

NOTE: The alarm priority value associates an alarm severity, page 79 to the alarm.

Default Alarm Priority Configuration

The table describes the alarm priority and their default properties:

Alarm Priority	Alarm Category	Representation
1	1001	
2	2001	
3	3001	
4	4001	

Alarm Severity

The severity is used to classify the alarm conditions. The library supports 4 levels of severity:

1	Critical	Process is about to stop (act now to prevent downtime) - 1000
2	High	Process quality is below expectations (resolve quality) - 2000
3	Medium	Process has stopped (resolve and restart) - 3000
4	Low	Process may stop (alert to understand) - 4000

Each severity is represented by a specific color and symbol.

Each severity contains a set of 10 alarm categories. The different alarm categories are represented by the priority of the asset:

1	Very High	Long repair time - 1
2	Normal	Average - 5
3	Auxiliary	Non-critical system - 9

The combination of the alarm severity and asset priority results in the alarm category for the alarms from the asset:

4001	Low Asset Severity (4000), High Asset Priority (1) Alarm
2005	High Asset Severity (2000), Average Asset Priority (5) Alarm

Alarm Shelving

Alarm Shelving Configuration

Alarm shelving is enabled by default for all categories of alarms.

Shelving Alarms

During operation, shelving of an alarm is done by clicking **Shelve For** or **Shelve Until** in the Alarm tab (see EcoStruxure™ Process Expert - General Purpose Library, Process Templates Reference Manual). A user can restore the shelved alarm by clicking **Restore**.

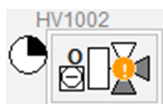
An alarm does not need to be active to be shelved. The shelving period starts when you shelve the alarm.

To shelve an alarm during runtime, your role needs to have the required operational permission.

Representation of Shelved Alarms During Operation

At the symbol level, the timer icon displayed on the genie indicates that an alarm is shelved. However, if for an object, an alarm is shelved and at the same time, another alarm is active, the timer icon does not appear. This is because active alarms are displayed over shelved alarms based on their respective priority.

The figure shows an alarm that is shelved during operation.

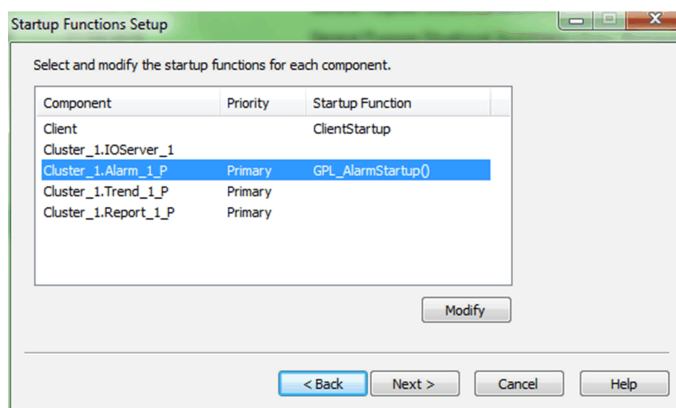


Data Logging

Overview

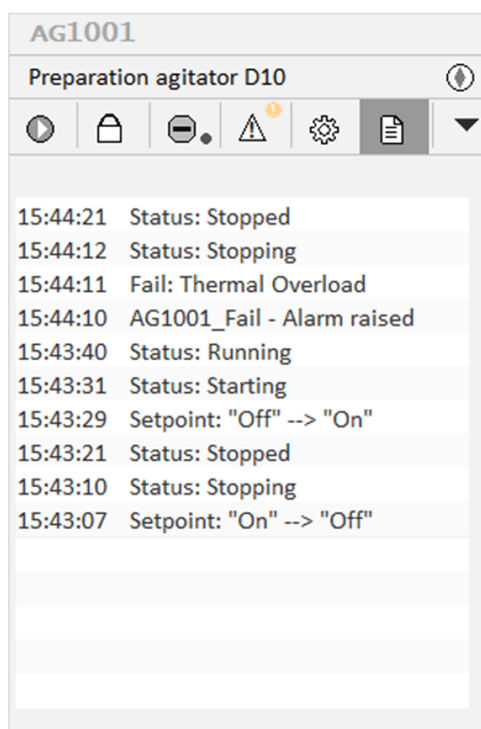
By default, the library logs operator actions and alarms. It optionally allows users to log asset status change, owner change and **Interlocks/Failures** first conditions. Logging operator actions is not an optional configuration. Logging asset changes can be optionally enabled from template as shown in the following figure.

To enable the datalogging the `GPL_AlarmStartup()` has to be called as the startup code in the alarm servers:



Operator action logging	The operator actions are logged in the format
	Label: Current Value → New Value.
Asset status change	The changes in the equipment status such as Running, Stopped etc. logged in the format
	Label: New Value
Owner Change	The change of Owner to local panel, external and cascade.
	Label: New Value
First Conditions	Failures and Interlocks first conditions are logged in the format.
	Interlocks/Failures: Condition description

This figure shows an example of controller change logging:



Items

Overview

Each asset/equipment contains items of information. The item may be an individual element (variable, alarm or trend) or an item may contain a combination of these elements.

Example: `PU3412.Running`

Item `Running` can refer to a digital variable only or also an event (used to log the status of the pump).

The syntax `<process model>.<asset name>.item` is available to be used to access information about the item. The item name is also displayed in user interfaces. The trended items associated with an asset are displayed to enable the user to select which trends to display.

The core element of most items is a variable. The library provides access to four different types of variable items.

Controller Variables

Controller Variables represent a tag within the controller.

All numeric variables (integer and real) are controller tags (Example: Gain and Speed).

All string variables are controller tags (Example: `StateDesc`).

All word registers containing 16 digital values are controller tags (Example: `StatusWord`, `ConditionWord`).

Calculated Variables

Calculated Items represent a summary variable based on a combination of other variables - Common Calculated Variables (Status, State, Owner, Mode, Abnormal, ActionRequired etc.).

Calculated variables are executed on the IO server. Their centralized execution may restrict system scalability.

Disk Variables

Disk items represent variables available only at the supervision level. All variables which only impact supervision operation are disk items (Example: Practical Range).

Virtual Items

Virtual items are used to extract digital variables from controller variables which are bit packed words and to setup enumerations to integer variables.

Maps of the contents of a digital array in a controller variable are part of the library.

Virtual Items are calculated on the client.

Properties

The supervision system provides properties for each item. The properties include its range, engineering units and format. These properties control the way the item is displayed. The library utilizes the additional custom fields associated with an item to add new properties which are used in their display.

The supervision elements display data as arrays. This allows the supervision templates to adapt as the user selects options within their template. The additional properties move the configuration from being fixed on the page to be part of the item.

Label (Custom 2)

The label is the text displayed before the variable. For a digital input the variable can represent any data. The engineer can use this property to enter a text that describes the variable.

Security (Custom 8)

The roles which can write to a variable are defined as a property of the item. Regardless of the location on the screen where these values are displayed the security settings will be maintained.

Custom Write Function (Custom 7)

Every item displayed can be changed (based on security settings) without a custom write function. The custom write function is used when variables (calculated variables) combine multiple variables in a single display and hence require complex menus to be written correctly. Mode is a typical example in which there can be different modes of operation selectable based on the current equipment status. Refer to the `GPL_ActionMode` function in the library.

The Custom command functions mainly have three section:

- Getting the enumeration
- Creating the Menu based on the current value of the item
- Executing the write function based on the selection made from the menu

Item API

Based on the value and additional variable properties the user can access all items through a common set of functions.

GPL_GetItemLabel (*ItemName*, *Cluster.Equipment*) - Returns a string Get the label property of the variable.

GPL_GetItemValue (*ItemName*, *Cluster.Equipment*) - Returns an Integer which is the numeric value of the variable.

GPL_GetItemReal (*ItemName*, *Cluster.Equipment*) - Returns a Real which is the numeric value of the variable.

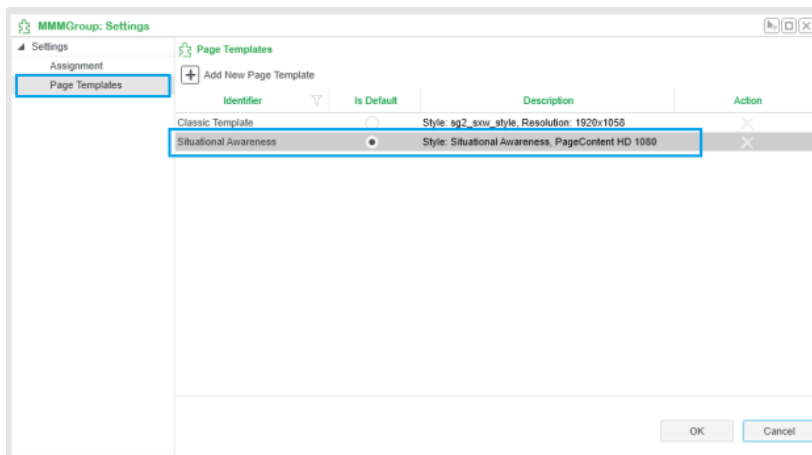
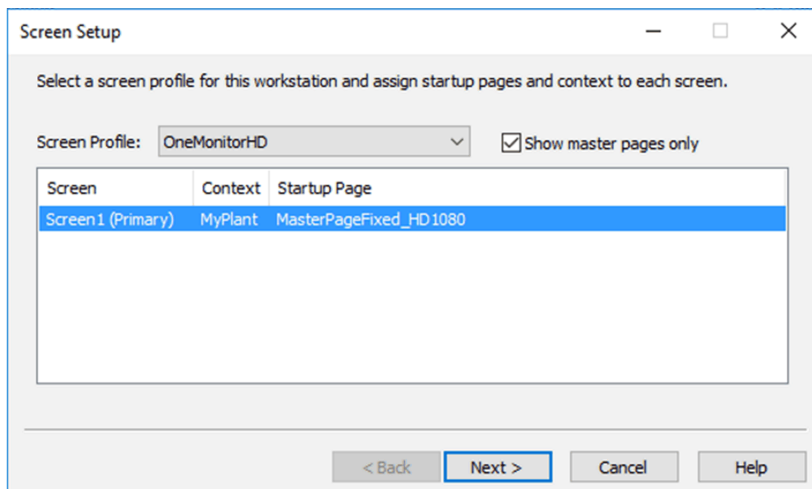
GPL_GetItemDisplay (*ItemName*, *Cluster.Equipment*) - Returns a string which is the formatted value or the text from the enumeration.

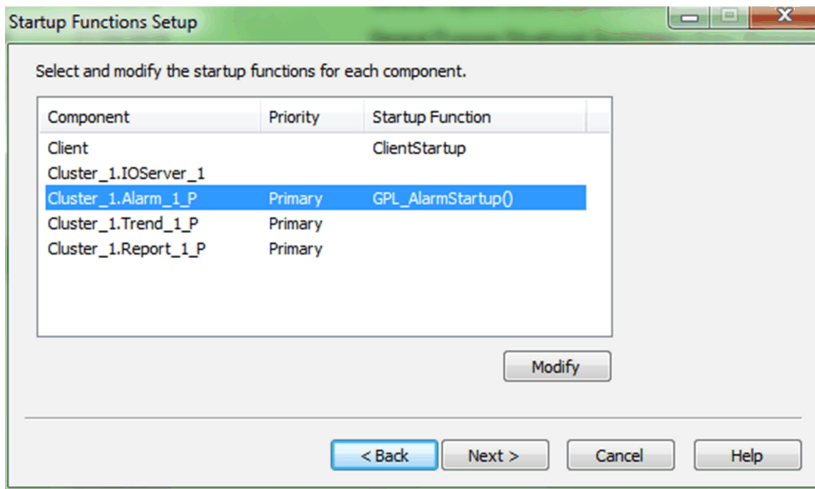
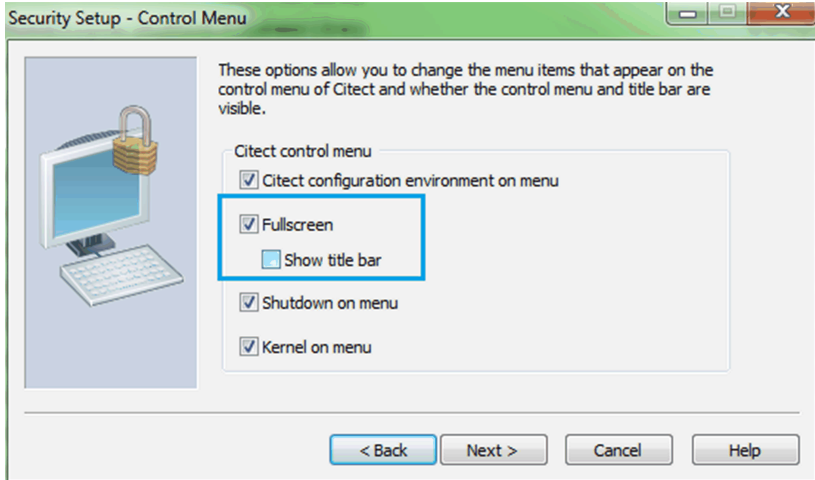
GPL_GetItemPriv (*ItemName*, *Cluster.Equipment*) - Returns an Integer which is the privilege level required to write to an Item.

When used on a faceplate *Cluster.Equipment* is left blank (it will read based on the equipment defined on the faceplate). On a process page the *Cluster.Equipment* must be specified.

Configure and Run a Supervision Project

The following table explains the steps to be followed to configure and run a supervision project using the situational awareness page template.

Step	Action												
1	<p>Create a new supervision project and right-click on Supervision Project and navigate to Settings → Page Templates → Select Situation Awareness as shown in the following figure.</p>  <p>The screenshot shows the 'MMMGroup: Settings' dialog box with the 'Page Templates' tab selected. A table lists available templates:</p> <table><tr><th>Identifier</th><th>Is Default</th><th>Description</th><th>Action</th></tr><tr><td>Classic Template</td><td><input type="radio"/></td><td>Style: sg2_srw_style, Resolution: 1920x1058</td><td></td></tr><tr><td>Situational Awareness</td><td><input checked="" type="radio"/></td><td>Style: Situational Awareness, PageContent HD 1080</td><td></td></tr></table> <p>Buttons at the bottom: OK, Cancel.</p>	Identifier	Is Default	Description	Action	Classic Template	<input type="radio"/>	Style: sg2_srw_style, Resolution: 1920x1058		Situational Awareness	<input checked="" type="radio"/>	Style: Situational Awareness, PageContent HD 1080	
Identifier	Is Default	Description	Action										
Classic Template	<input type="radio"/>	Style: sg2_srw_style, Resolution: 1920x1058											
Situational Awareness	<input checked="" type="radio"/>	Style: Situational Awareness, PageContent HD 1080											
2	Create new page in the supervision project and instantiate new genie on the page.												
3	Generate the project and refine it to create the navigation menu to navigate from the page tree on workspace (Refer page navigation section, page 50).												
4	Save the refinement and build and deploy the project.												
5	<p>Run the Computer Setup Wizard and select the profile and master page. For details, refer to the topics describing the screen setup and master pages in the help of AVEVA Plant SCADA.</p>  <p>The screenshot shows the 'Screen Setup' dialog box. It includes a dropdown for 'Screen Profile' set to 'OneMonitorHD' and a checked checkbox for 'Show master pages only'. Below is a table with screen configuration:</p> <table><tr><th>Screen</th><th>Context</th><th>Startup Page</th></tr><tr><td>Screen1 (Primary)</td><td>MyPlant</td><td>MasterPageFixed_HD 1080</td></tr></table> <p>Buttons at the bottom: < Back, Next >, Cancel, Help.</p>	Screen	Context	Startup Page	Screen1 (Primary)	MyPlant	MasterPageFixed_HD 1080						
Screen	Context	Startup Page											
Screen1 (Primary)	MyPlant	MasterPageFixed_HD 1080											

Step	Action
6	<p>To enable the datalogging the <code>ICodeGPL_AlarmStartup()</code> / <code>ICode</code> has to be called as the startup code in the alarm servers.</p> 
7	<p>Proceed with the setup and select fullscreen and uncheck show title bar in the security setup.</p>  <p>NOTE: The resolution of the system on which the runtime is hosted should be set to 1920 x 1080.</p>

Customization

Overview

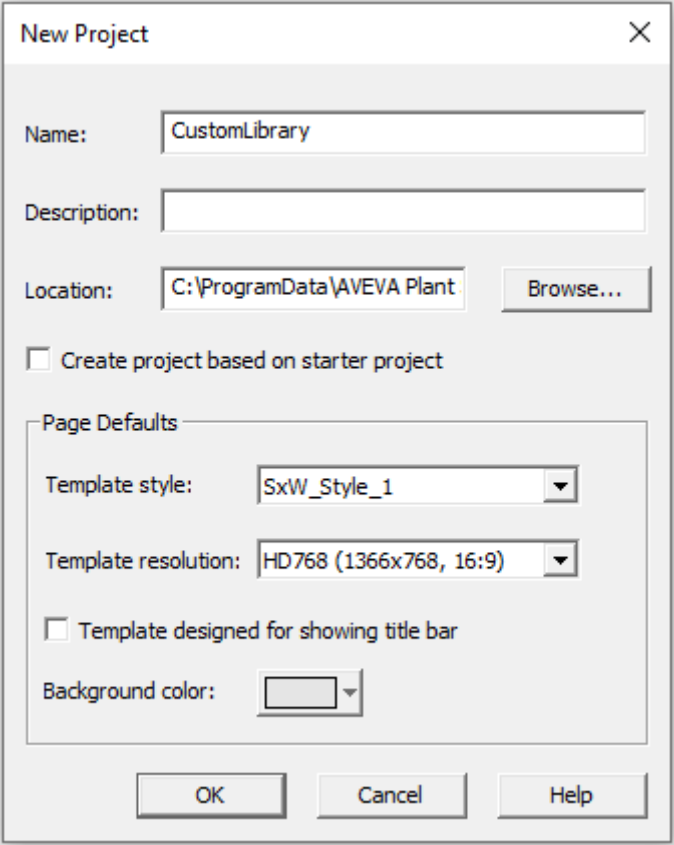
This chapter describes the custom library for process and devices.

Supervision

Custom Library

`CustomLibrary` is an include in which the customization required for a project is stored and it has to be included in the runtime project.

The following table describes the creation of custom library.

Step	Action
1	<div>Create a new project in <code>Plant SCADA</code> as shown below.</div> <div></div> <div>NOTE: The new project name is just an example.</div>
2	Select <code>GeneralPurposeLibrary</code> as an include to this project.

Color Customization

The colors can be customized for equipment status and Present Value (PV).

The following table shows the list of available colors:

Color Name
Black
Blue
Green
Cyan
Red
DkGrey
LtBlue
LtGreen
LtCyan
LtRed
LtMagenta
Yellow
White
State Change Green

Any one of the above mentioned colors can be chosen using the color name.

There is a set of default colors defined in the Parameters of GeneralPurposeLibrary project, as shown in the following table.

Section name	Name	Value	Comment
GPL.DefaultColor.Valve	Open	DkGrey	Default color for valve status - open
GPL.DefaultColor.Valve	Closed	White	Default color for valve status - closed
GPL.DefaultColor.Motor	Stopping	White	Default color for motor status - stopping
GPL.DefaultColor.Motor	Starting	White	Default color for motor status - starting
GPL.DefaultColor.Motor	Stopped	White	Default color for motor status - stopped
GPL.DefaultColor.Analog	PV	Green	Default color for analog present value
GPL.DefaultColor.Motor	Running	DkGrey	Default color for motor status - running
GPL.DefaultColor.Valve	Traveling	State Change Green	Default color for valve status - traveling
GPL.DefaultColor.Valve	Stopped	Grey	Default color for valve status - stopped
GPL.DefaultColor.Valve	Unknown	Black	Default color for valve status - unknown
GPL.DefaultColor.Valve	Closing	State Change Green	Default color for valve status - closing
GPL.DefaultColor.Valve	Opening	State Change Green	Default color for valve status - opening
GPL.DefaultColor.Motor2	Stopping	DkGrey	Default color for motor2 status - stopping
GPL.DefaultColor.Motor2	Starting Forward	White	Default color for motor2 status-starting forward
GPL.DefaultColor.Motor2	Starting Reverse	White	Default color for motor2 status-starting reverse
GPL.DefaultColor.Motor2	Forward	DkGrey	Default color for motor2 status - forward
GPL.DefaultColor.Motor2	Reverse	DkGrey	Default color for motor2 status - reverse
GPL.DefaultColor.Motor2	Stopped	White	Default color for motor2 status - stopped
GPL.DefaultColor.Motor2	Unknown	Black	Default color for motor2 status - unknown
GPL.DefaultColor.Valve	Inlet	DkGrey	Default color for valve inlet of 3 way valve
GPL.DefaultColor.Valve	Stopping	DkGrey	Default color for valve status - stopping
GPL.DefaultColor.Valve	Transit Stop	Grey	Default color for valve status - transit stop
GPL.DefaultColor.Valve	Fully Open	Black	Default color for valve status - fully open

You can change the color of a status by creating a new parameter in Custom library in the same format as done for default colors and replacing "Default" by "Custom" in the section name.

For example, if the running color of motor has to be made green, follow these steps:

Step	Action
1	Go to the <code>customlibrary</code> and copy the default color row of Motor running.
2	Paste it in the Custom library, change the Comment and rename the Section Name to <code>GPL.CustomColor.Motor</code>
3	Change the Value to your desired color from the list of available colors which is green in this example. Make sure the name is an exact match.
4	The customized color will be reflected for all motor types when you start the runtime.

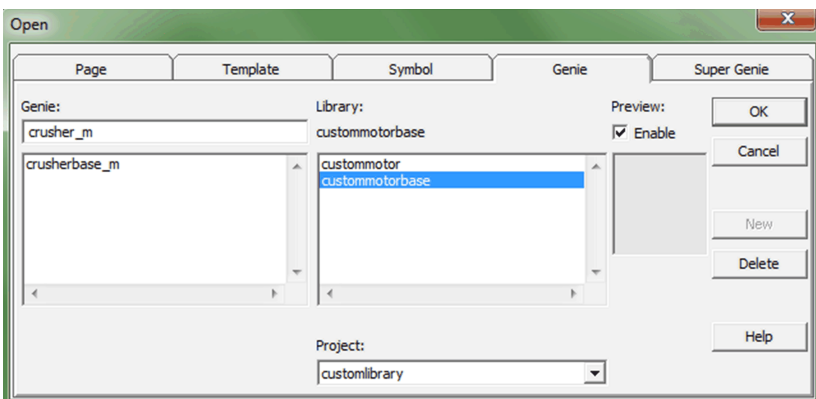
The following table explains which section name is applicable for genres of which equipment type.

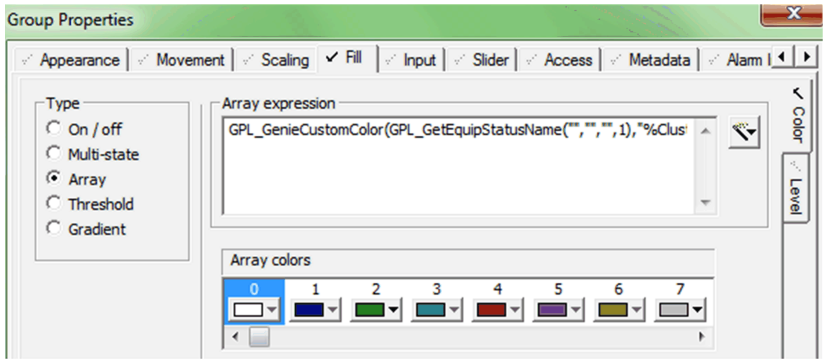
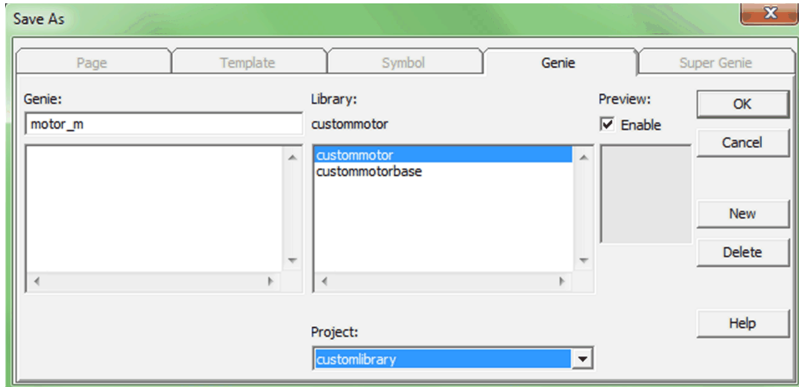
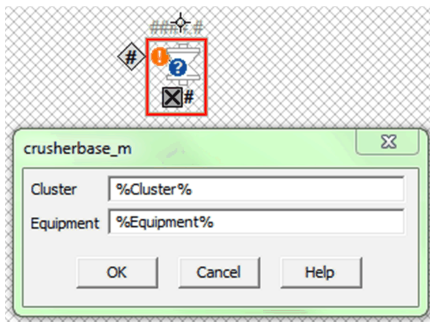
Section name	Applicable equipment types
<code>GPL.DefaultColor.Motor</code>	Motor
<code>GPL.DefaultColor.Motor2</code>	Motor2, MotorVS, MValve- motor, MValveWithPositioner- motor
<code>GPL.DefaultColor.Valve</code>	Valve, HandValve, MValve, MValveWithPositioner, DualOutputValve, ControlValve, AnalogOutput
<code>GPL.DefaultColor.Analog</code>	All equipment types showing Present Value (PV)

Creating a Custom Genie

This section explains the steps required to create a custom genie.

Example: The mining industry requires a crusher symbol instead of the regular representation.

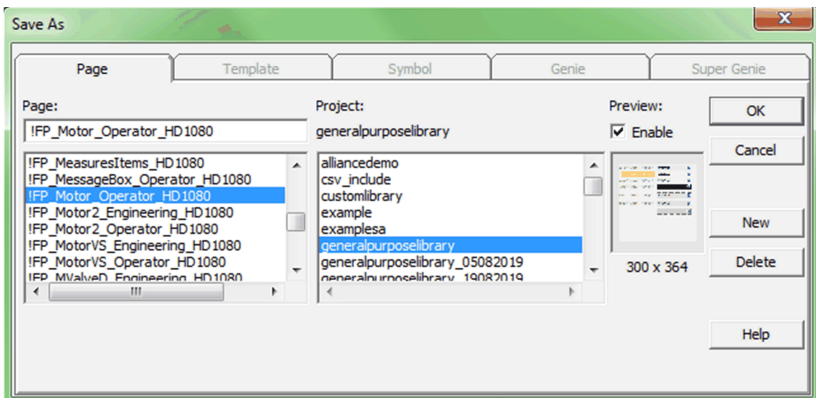
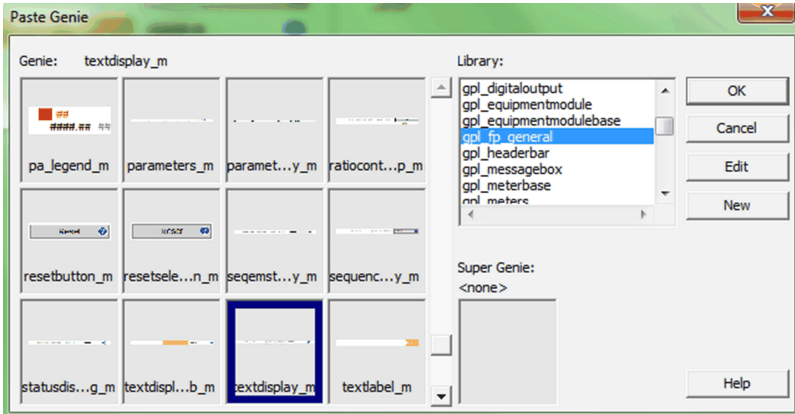
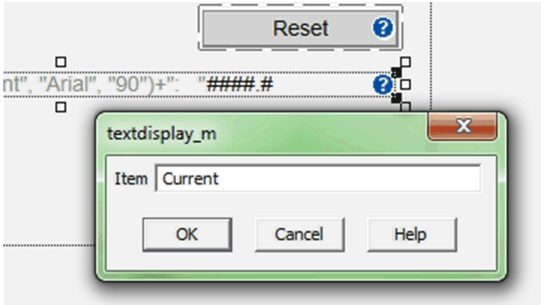
Step	Action
1	<p>Create the crusher base symbol in the <code>customlibrary</code>.</p>  <p>NOTE: The user should not group multiple layers, for more details refer to BaseSymbol in GeneralPurposeLibrary project.</p>
2	Copy the script in the fill array animation from any base motor symbol from the GeneralPurposeLibrary project.

Step	Action
	 <p>NOTE: In case of a read/write variable, user has to use NumericDisplayWrite genie.</p>
3	<p>Save the motor genie from general purpose library to the custom library and give a name.</p> 
4	<p>Replace the base symbol with the new base symbol created in the custom library and pass the cluster and equipment parameters and save the genie.</p> 

Adding a Numeric Data to the Operator Tab

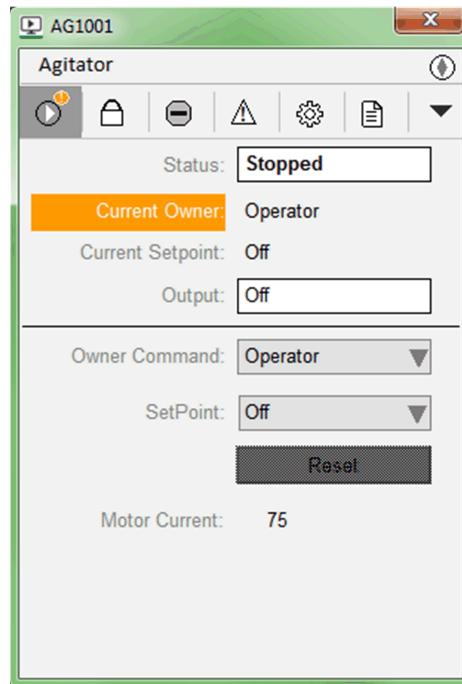
To insert a new data on faceplate, the variable has to be added in the function block and the corresponding tag has to be generated.

The steps to add the new data to faceplate which is already generated from EcoStruxure Process Expert are explained in the following table.

Step	Action
1	<p>Open the faceplate "IFP_Motor_Operator_HD1080" and save as "IFP_Motor_Operator_Custom_HD1080" in the <code>customlibrary</code>. All the operator tabs can be customized by adding the suffix "Custom" as per the above example for motor.</p> 
2	<p>Instantiate the <code>textdisplay_m</code> genie from the <code>gpl_fp_general</code> library on the saved page.</p> 
3	<p>Enter the item name of the tag which is generated from EcoStruxure Process Expert (For example: Current).</p> 

NOTE: If the user needs the engineering unit to be shown, then an additional display symbol must be added to show the unit.

The method to import the custom Library to EcoStruxure Process Expert is explained in the template customization section, page 98.



Adding a Digital Value to the Measures Tab

When the digital signal value is propagated by using the existing status word, the item has to be added to the type definition. The item has to be added to type if there is a necessity to set up enumeration for integer data type.

Type Definition

All the templates in the library are categorized as a specific type, the items belonging to the types and its configuration are predefined and loaded into the cache during the startup using cicode functions.

The function which is used to load an item is `GPL_LoadItem`. The parameters of this function are as follows:

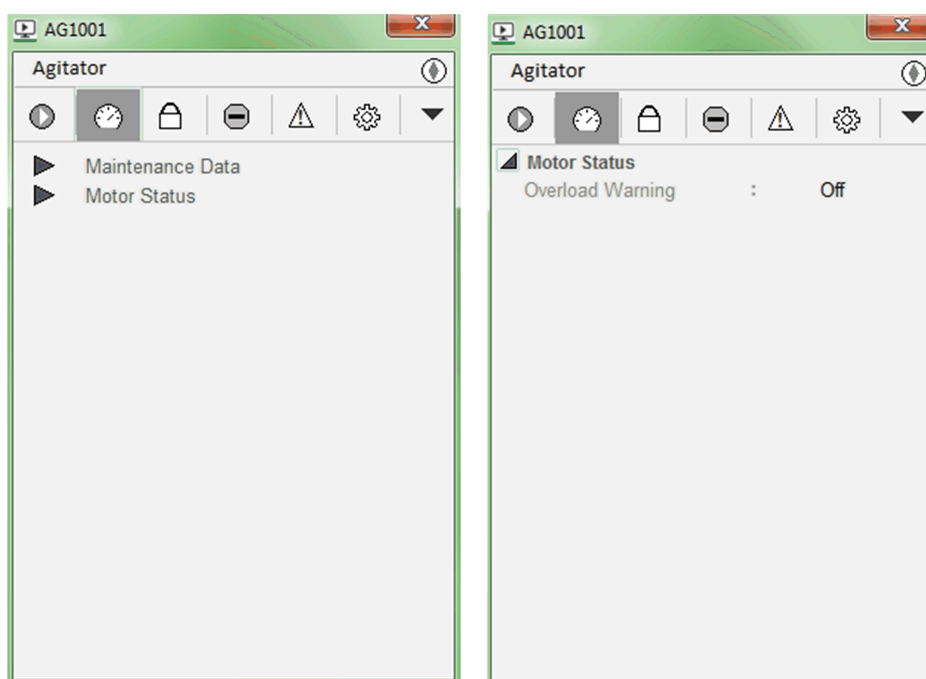
Parameter	Description
sEquipment	The type to which the equipment belongs (example: <code>Type.Motor</code>), the phrase <code>Type.</code> has to be prefixed with the type.
sItem	Name of the item (example: <code>warning</code>)
sAddr	The bit number of the source tag in which the item is available (example: 12) (If the item is not a part of a word, then any dummy address)
sComment	Comment of the item (example: <code>@(warning Status)</code>)
sType	The data type (example: <code>DIGITAL</code>)
sSourceTag	Item name of the source tag. (example: <code>StatusWord</code>) Source tag: Source tag has to be defined if a digital value is passed in word or an enumeration has to be set up for an actual integer variable. <ul style="list-style-type: none"> Case 1: Digital value is passed in word. Source tag is the item name of the variable tag which has the word value. Case 2: Setup enumeration for integers. Source tag is the name of the actual item for which enumeration has to be setup.
sSetupFunc	The name of the setup function (Example: <code>GPLSetupBit</code>) Setup function: It is the name of the function which creates the subscription to the source tag and defines a callback function which updates the cache.
sEnum	The name of the enumeration if applicable (Example: <code>OnOff</code>)

Parameter	Description
sLabel	The display label of the item (Example: @ (Overload Warning))
sParamTab	The tab on which the item should be displayed (Example: Measures)
sParamGroup	The name of the group in which the item has to be displayed in the tab. The position of the item in the group (example: @ (Motor Status) , 1)
sCustomCommand	The custom function name to be executed when an action is performed by using the command genie.

The following is an example of the script to be written in the custom library, if the user has to add an alert information to the 12th bit of status word and display the same in the measures tab of motor.

```
FUNCTION GPL_CustomVariableParams ()
GPL_LoadItem("Type.Motor", "Warning", "12", "@ (Warning
status)", "DIGITAL", "StatusWord", "GPL_
SetupBit", "OnOff", "Overload Warning", "Measures", "@
(MotorStatus), 1", "") END
```

The measures tab will display the data as displayed in the following figures.



Order of the group in measures tab

If user wants `Motor Status` group to appear as the first group, check the order number of `Maintenance Data` group in the Parameters section, `GPL.Group.MeasuresPos`.

NOTE: Parameters are created for all the Measures tab groups. Refer to parameter section for details.

itect Studio - CustomLibrary [Active Project]

Row	Section Name	Name	Value
1	GPL.Group.MeasuresPos	Maintenance Data	100

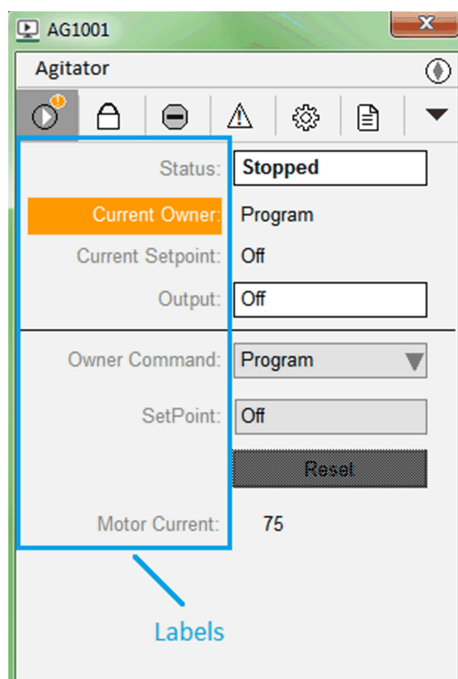
Create a new parameter for group for Motor Status with a lower value than the Maintenance data group as shown in the following figure:

Setup Alarming Parameters Screen Profiles Devices Events Languages Keyboard Keys Custom Files			
Save Discard Copy Paste Delete Row(s) Export All Import All			
Row	Section Name	Name	Value
1	GPL.Group.MeasuresPos	Motor Status	99

You need to restart the runtime to see the change.

Overriding label for a type.

The following figure shows the default labels



Each row in the faceplate is mapped to a certain item, the details are available in the supervision section. The function which helps the user to override is `GPL_LoadOverrideItem()`. The parameters of this function are as follows:

- **sEquipment:** The type of equipment for which the override has to be made.
- **sName:** itemname_Desc
NOTE: _Desc is a key word for label. The other supported keyword for this parameter is _Enum
- **sValue:** The new label.

The first row in the motor faceplate is mapped to item `Equipstatus` and the third row is mapped to `Setpoint`. To override the labels of these items, the following script has to be written in the `customlibrary`.

```
FUNCTION GPL_LoadOverrideItems();

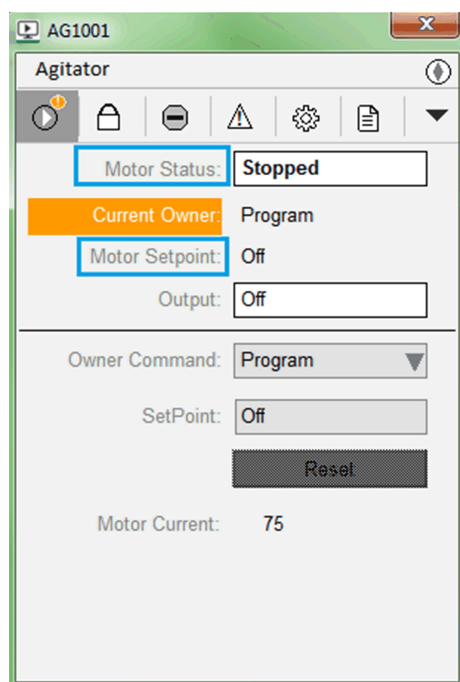
GPL_LoadOverrideItem("Type.Motor","EquipStatus_Desc","@(
Motor Status)");

GPL_LoadOverrideItem("Type.Motor","EquipStatus_Desc","@(
Motor Setpoint)");

END
```

NOTE: Only one instance of `GPL_OverrideItems` has to be called in the custom library.

The faceplate will show the new text once the new script is written:



Customizing Enum for a type.

The enumerations are shown on the right side of the faceplate. If an enumeration has to be changed then the new enumeration has to be loaded first. The new enumeration can be loaded using the function `GPL_LoadEnum()`. The parameters of this function are as follows:

- **sEnum:** EnumName_IntegerValue (example: `MotorSetpoint_0`).
- **sValue:** The corresponding Enum text. (Example: `MotorOff`).

The following is an example of adding a new enumeration to be added in the custom library.

If the user needs to change the enumeration in the third row of motor to the new enumeration, then the `GPL_OverrideItems` should contain the following override.

```
FUNCTION GPL_LoadOverrideItems();

GPL_LoadOverrideItem("Type.Motor", "Setpoint_Enum", "MotorWarning");

END

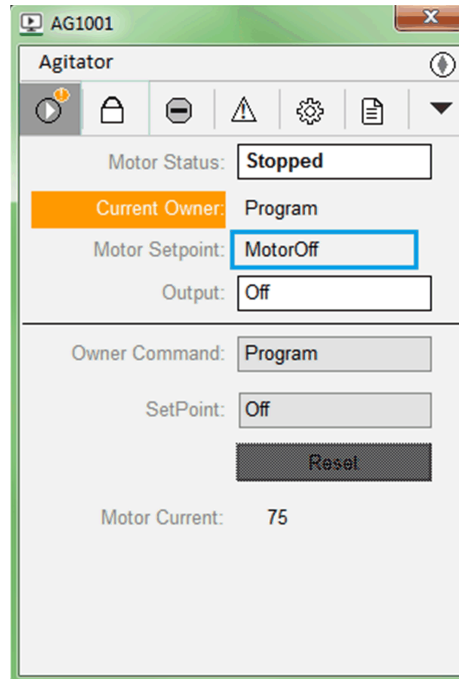
FUNCTION GPL_CustomEnum()

GPL_LoadEnum("MotorWarning_0", "@(MotorOff)");

GPL_LoadEnum("MotorWarning_1", "@(MotorOn)");

END
```

The faceplate will reflect the change as shown in the following figure:



Customizing an Enum across all types.

The Enum can be customized across all the strings using the `GPL_LoadOverrideItem()` function and the parameters must be configured as below:

- **sEquipment:** Enum (This is a keyword).
- **sName:** EnumName_Value (EnumName is the name of the Enum which has to be overridden).
- **sValue:** The new string.

```
FUNCTION GPL_LoadOverrideItems();
```

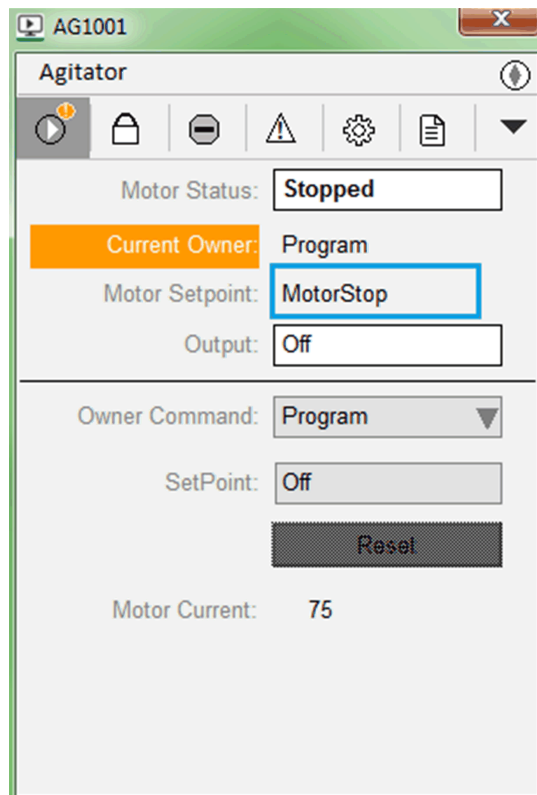
```
GPL_LoadOverrideItem("Type.Motor", "EquipStatus_Desc", "@(MotorStatus)");
```

```
GPL_LoadOverrideItem("Type.Motor", "Setpoint_Desc", "@(MotorSetpoint)");
```

```
GPL_LoadOverrideItem("Type.Motor", "Setpoint_Enum", "MotorSetPoint");
```

```
GPL_LoadOverrideItem("Enum", "MotorWarning_0", "MotorStop");
```

```
END
```



Adding a New Tab in the Faceplate

This table explains the how a new tab can be added to the faceplate:

Step	Action
1	Open the menu configuration (Visualization > Menu Configuration) of the custom library in Plant SCADA.
2	<p>Add the new tab in the corresponding menu which is applicable to the asset.</p> <p>Page: Menu name (Example: <code>Motor_Tabs</code>)</p> <p>Level 1: Tab name (Example: <code>@ (Custom)</code>)</p> <p>Target Page: Name of the page which has to be opened while navigating to this tab</p> <p>Order: The position of the tab in the menu</p> <p>Symbol: The symbol which has to be displayed in the tabs bar</p> <p>Custom1: 1</p> <p>Custom3: 1</p>

Replacing the Operator tab in the Faceplate

The following table explains the steps to replace the operator tab with a custom operator tab in the faceplate.

Step	Action
1	Create a tab with name as FP_<Operator Tab Name>_Custom (Example: FP_Motor_Operator_Custom) in the custom library.
2	Open the Menu Configuration (Visualization > Menu Configuration) in the custom library.
3	<p>Add the new entry in the corresponding menu which is applicable to the asset.</p> <p>Page: Menu name (Example: Motor_Tabs) Level 1 : @ (Operator_Custom)</p> <p>Target Page: Name of the page which has to be opened while navigating to this tab FP_<Operator Tab Name>_Custom (Example : FP_Motor_Operator_Custom)</p> <p>Order: The same order of the operator tab (Example :TB_OPER)</p> <p>Symbol: The symbol which has to be displayed in the tab bar same as that of the tab to be replaced</p> <p>Custom1: 1</p> <p>Custom3: 1</p>

Alarm Format Customization

This figure shows the default format of the alarm pages.

Row	Section Name	Name	Value	Comment
1	FORMAT	DefaultGPLAllActiveAlarm_HD1080	{PriorityandState,24}{OnDate,90}{OnTime,90}{Custom7,120}{Name,180}{Desc,180}{Custom8,100}{Custom1,100}{Custom2,100}{Custom3,100}{Custom4,100}	Alarm list format for Active alarm page in 1080
2	FORMAT	DefaultGPLAllSOE_HD1080	{PriorityAndState,24}{Date,90}{Time,90}{Custom7,120}{Message,450}{Desc,180}{UserName,100}{UserLocation,100}	Alarm list format for SOE page in 1080
3	FORMAT	DefaultGPLAllShelvedAlarm_HD1080	{PriorityandState,24}{OnDate,90}{OnTime,90}{Custom7,120}{Name,180}{Desc,180}{DisableEndDate,130}{DisableEndTime,130}{DisableComment,240}	Alarm list format for shelved alarm page in 1080
4	FORMAT	DefaultGPLAllMaintActiveAlarm_HD1080	{PriorityandState,24}{OnDate,90}{OnTime,90}{Custom7,120}{Name,180}{Desc,280}{Custom8,100}{Custom1,100}{Custom2,100}{Custom3,100}	Alarm list format for MaintenaActive alarm page
5	FORMAT	DefaultGPLTop4ActiveAlarms_HD1080	{PriorityandState,24}{OnDate,90}{OnTime,90}{Custom7,120}{Name,200}{Desc,350}{Custom8,150}	Alarm list format for the top 4 alarms

User can manage the display format of an alarm list details by adding the following parameters in the **Parameter** tab (**Setup > Parameters**) or in `Citect.ini`. By adding this the default configuration will be overwritten and will be displayed as per the configuration format in the Alarm pages. Refer to the following figure:

Row	Section Name	Name	Value	Comment
1	FORMAT	GPLAllActiveAlarm_HD1080	{PriorityandState,24}{OnDate,90}{OnTime,90}{Custom7,120}{Name,180}{Desc,180}	Alarm list format for Active alarm page in 1080

Example:

Section Name - FORMAT

Parameter Name - GPLAllActiveAlarm_HD1080 (Changes as per the type of alarm page)

Value -- {PriorityandState,24}{OnDate,90}{OnTime,90}{Custom7,120}{Name,180}{Desc,180}

Verified User Name Customization

User can customize the name to be displayed in the Verified user confirmation pop-up by configuring the parameters in **Parameters** tab or `Citect.ini`.

Example:

Section Name - GPL.Verifier.Label

Parameter Name - Privilage6

Value - Plant Head

Row	Section Name	Name	Value
1	GPL.Verifier.Label	Privilage6	Plant Head

AG1001-Preparation agitator D10

Action: Owner Command=Operator

Current: Owner Command=Program

Comment:

Operator

Name: ABC

Password:

Plant Head

Name:

Password:

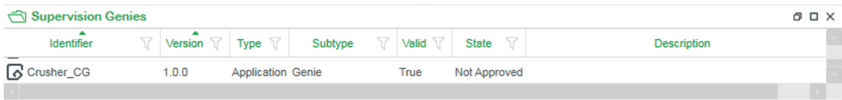
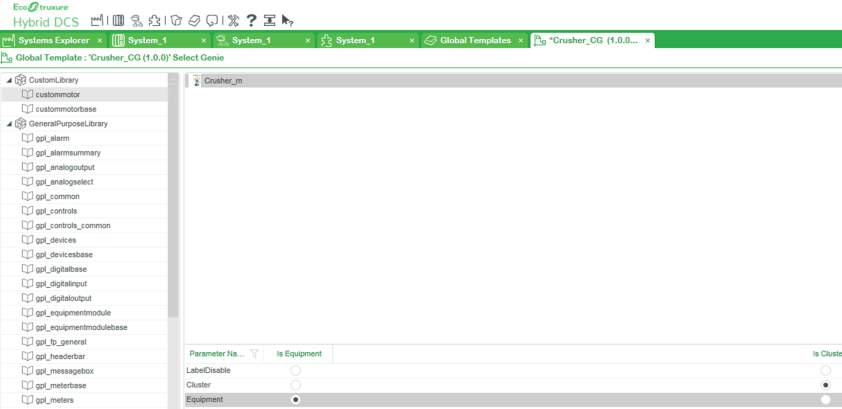
OK

Cancel

Customizing Templates

Templatizing a Custom Genie

In this section the steps needed to templatize a custom genie are explained, a crusher genie which was created in the supervision customization is considered as an example.

Step	Action
1	Create a new genie template. 
2	Open the template, select Templatizer and click + symbol present at the bottom left corner of the screen to import the CustomLibrary.ctz backed up from Plant SCADA.
3	Select the genie which has to be templatized and configure the Equipment and cluster parameter. Save the template after this configuration. 
4	Duplicate the \$MotorGP control module and select to duplicate the genie composite as shown in the following figure.

\$MotorGP (1.0.44) : Duplicate

Please specify the details for the templates to be duplicated

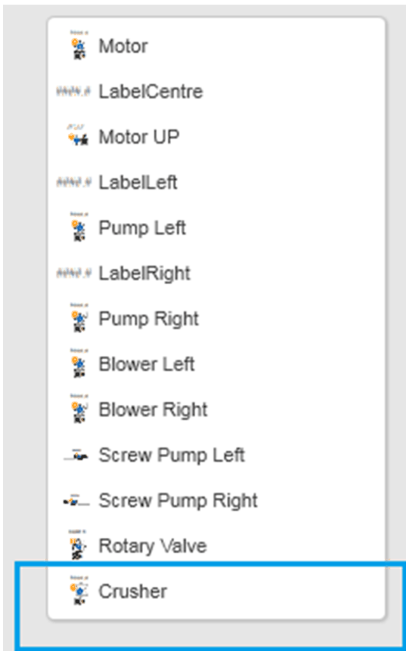

New Identifier
☒ Prefix ☐ Suffix

Identifier and Version	Type	Action	New Identifier	New Version
\$MotorGP (1.0.44)	Composite Template	Duplicate	MotorGP	1.0.0
\$MotorGP_CS (1.0.41)	Composite Template	Duplicate	MotorGP_CS	1.0.0
\$MotorDataGP_CD (1.0.28)	Composite Template	None		
\$CONDSUM1ONGP_CD (1.0.6)	Composite Template	None		
\$CONDSUMGP_CD (1.0.10)	Data	None		
\$DEVMMTGP_CD (1.0.18)	Data	None		
\$INTERLOCK1ONGP_CD (1.0.34)	Composite Template	None		
\$INTERLOCKGP_CD (1.0.19)	Data	None		
\$MOTORGP_CD (1.0.12)	Data	None		
\$MOTORLPGP_CD (1.0.6)	Data	None		
\$MotorGP_CG (1.0.6)	Composite Template	Duplicate	MotorGP_CG	1.0.0
\$BlowerLeftGP_m_CG (1.0.2)	Genie	None		
\$BlowerRightGP_m_CG (1.0.1)	Genie	None		
\$LabelGP_CG (1.0.3)	Composite Template	None		
\$Label_CentreGP_m_CG (1.0.3)	Genie	None		

☐ Display interfaces

Description of the change
Template duplicated from the template \$MotorGP (1.0.44)

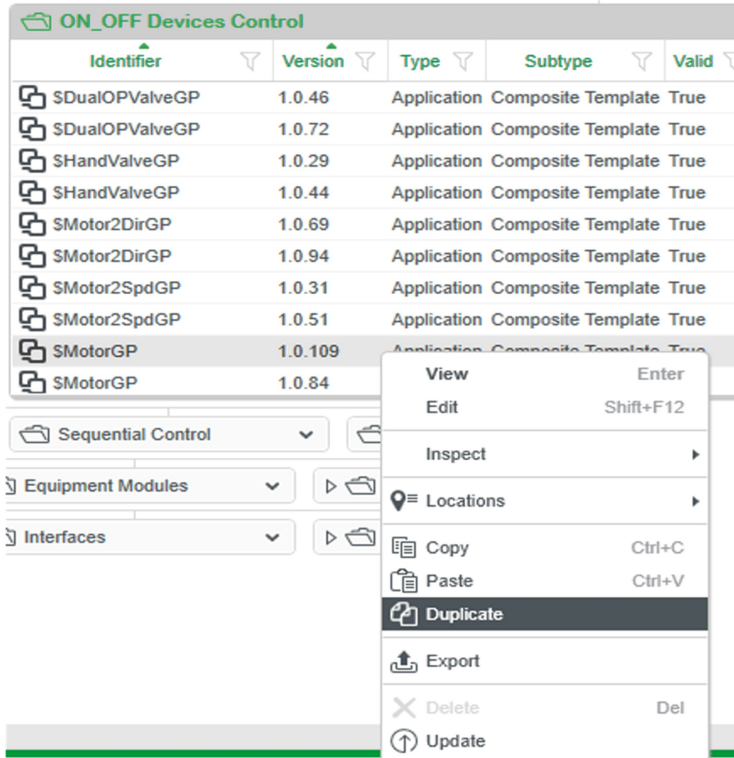
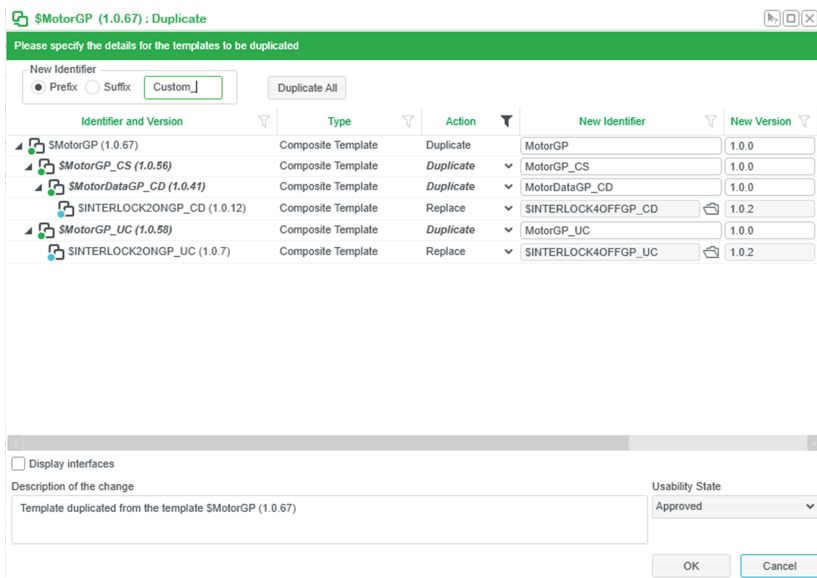
Usability State
Approved

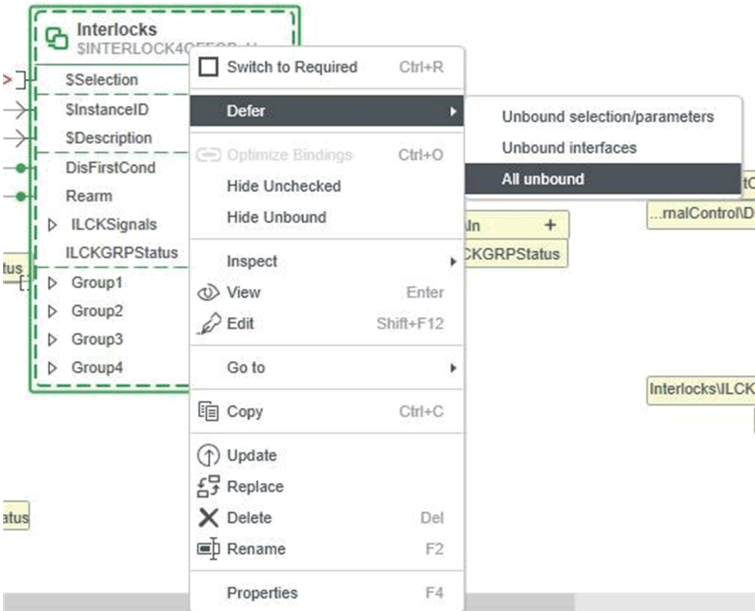
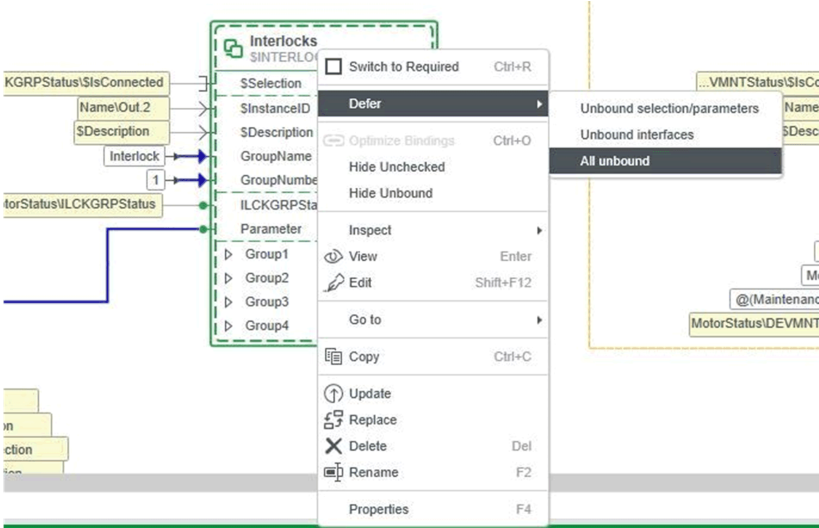
Step	Action
6	<p>When you create an instance from this template, the newly added genie is available for instantiation.</p> 
7	<p>Generate, build and deploy the supervision project to see the genie in the runtime.</p> 

Replacing the Default Interlock Service by a Different One

Replacing an existing interlock service by four interlock services

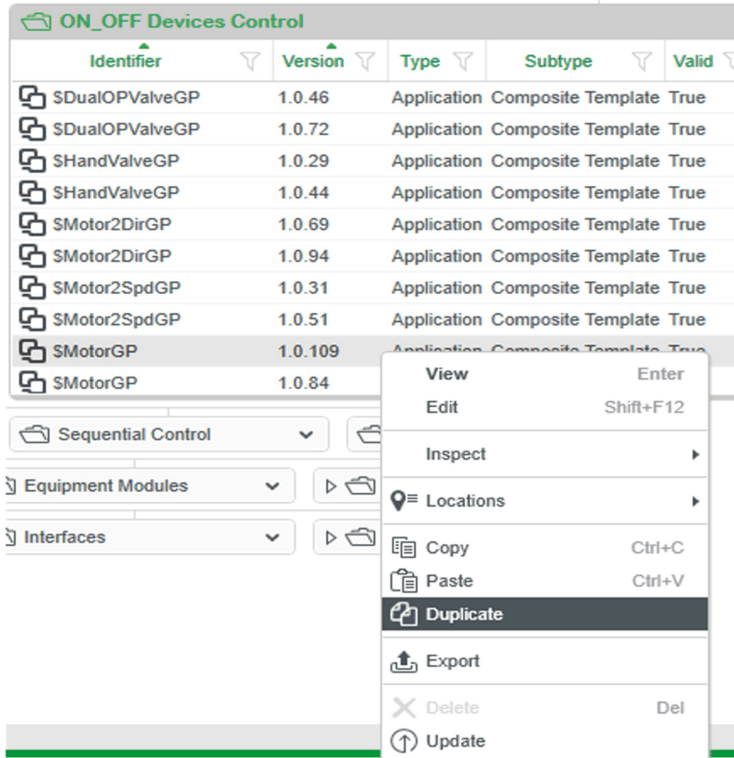
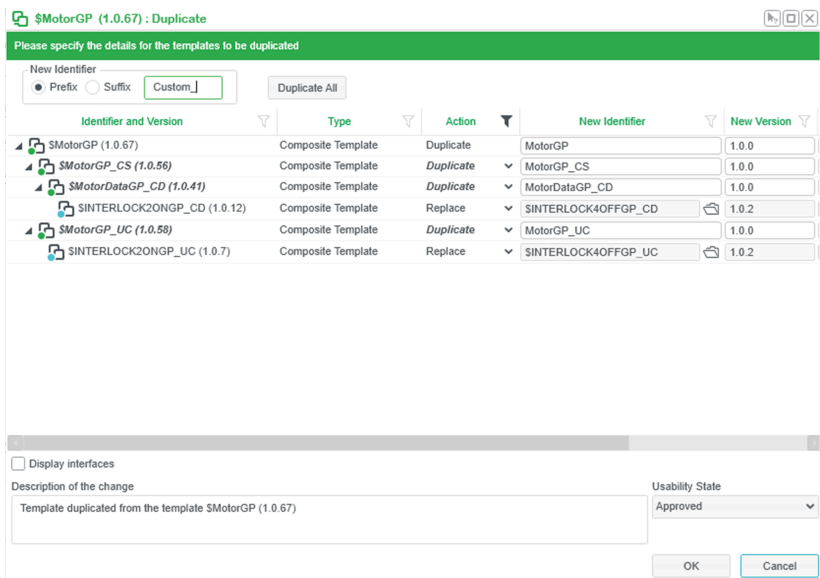
Perform the following steps to replace the existing interlock service by four interlock service:

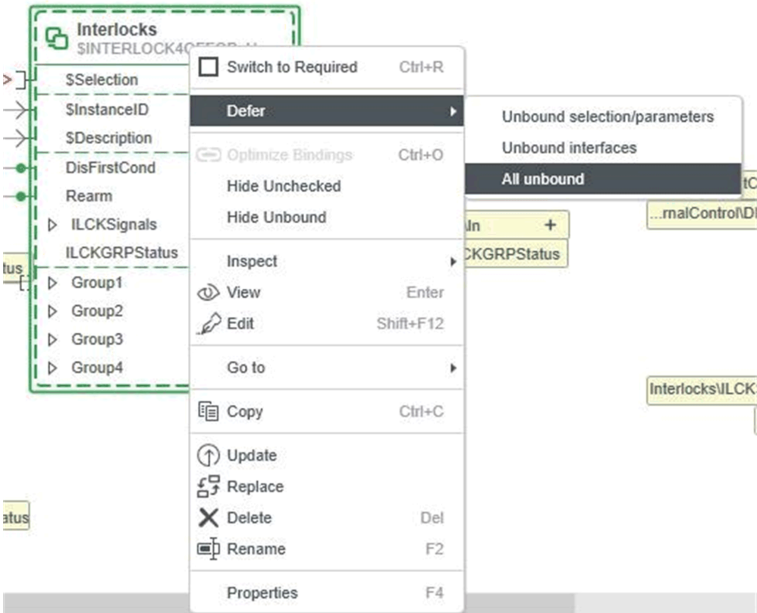
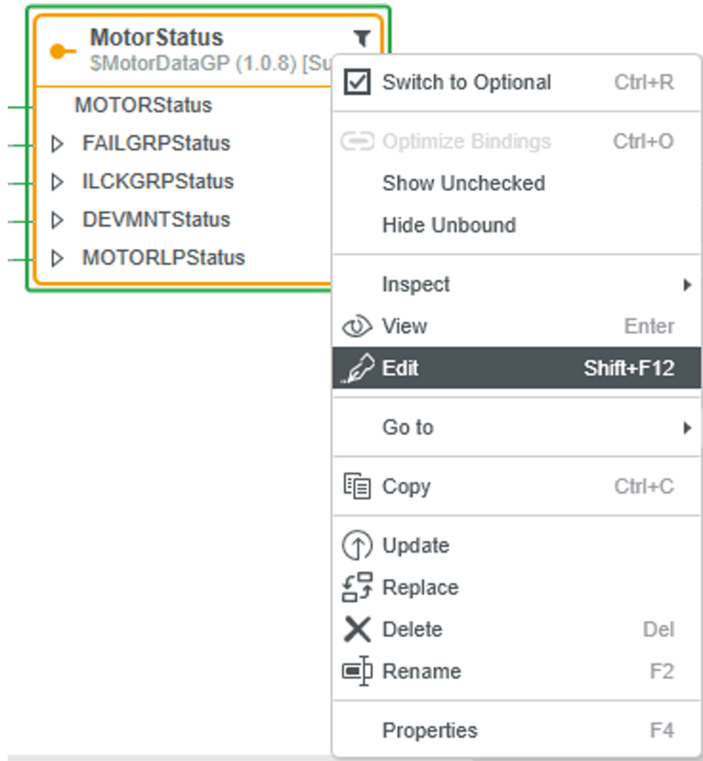
Step	Action
1	<p>Select Duplicate on the \$MotorGP control module and rename as Custom_MotorGP.</p>  <p>The screenshot shows the 'ON_OFF Devices Control' window with a table of modules. The \$MotorGP module (version 1.0.84) is selected. A right-click context menu is open, showing options like View, Edit, Inspect, Locations, Copy, Paste, Duplicate, Export, Delete, and Update. The 'Duplicate' option is highlighted.</p>
2	<p>Select Replace to replace InterlockON2GP by Iterlock4ONGP for both UC and CD the upper layers will get automatically duplicated.</p>  <p>The screenshot shows the '\$MotorGP (1.0.67) : Duplicate' dialog box. The 'New Identifier' is set to 'Custom_'. The 'Action' column shows 'Replace' for the selected templates. The 'Usability State' is set to 'Approved'.</p>
3	<p>Open Custom_MotorGP template and edit the Custom_MotorGP_UC control composite.</p>
4	<p>Right-click the Interlock4ON_UC composite > Defer, and select the All unbound to defer the parameters. Then, update all the parent templates within the Custom_MotorGP_UC template.</p>

Step	Action
	
5	Open Custom_MotorGP_CS composite and edit the Custom_MotorDataGP_CD supervision composite.
6	<p>Right click the Interlock4ON_CD composite > Defer, and select All unbound to defer the parameters. Then, update all the parent templates within the Custom_MotorGP_UC template.</p> 
7	In control module, Defer the control and supervision composite. Then, delete the parameters EPEFORASP, MotorSC in control composite and GroupSC in supervision composite.
8	Save the template with new build version.
9	Instantiate the template in application level, generate the control and supervision projects, build and deploy the control projects.

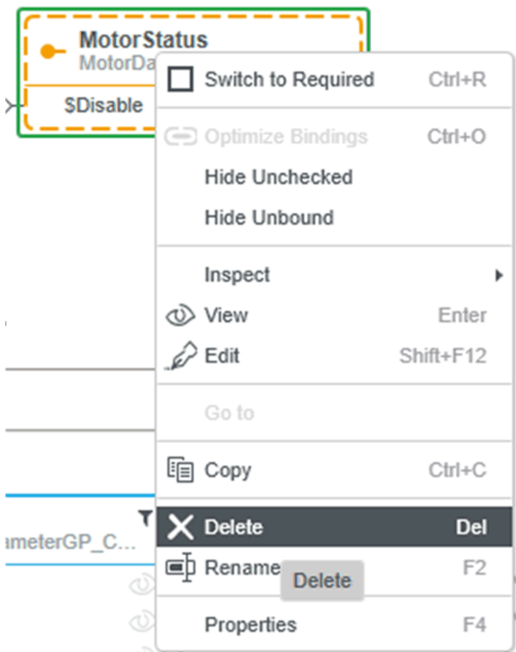
Replacing an existing interlock service by eight interlock services

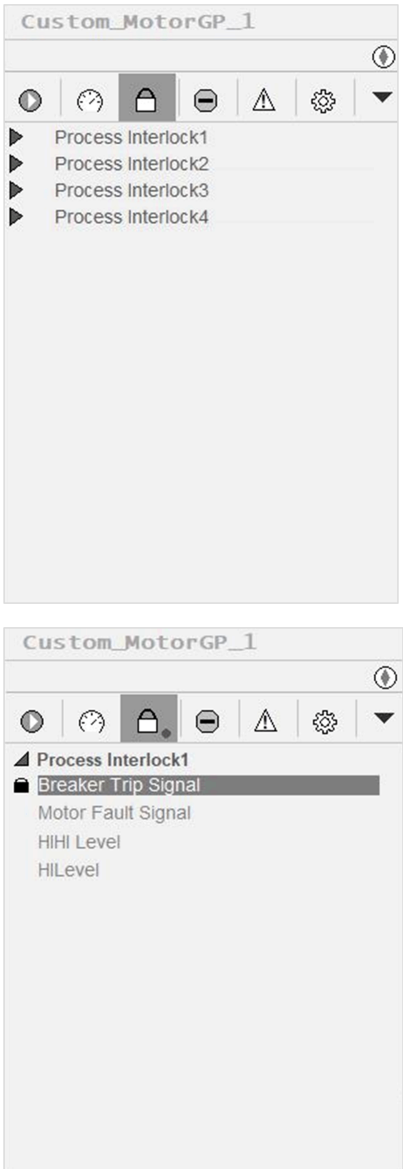
Perform the following steps to replace the existing interlock service by eight interlock services:

Step	Action
1	<p>Select Duplicate on the \$MotorGP control module and rename as Custom_MotorGP.</p>  <p>The screenshot shows the 'ON_OFF Devices Control' window with a table of modules. The \$MotorGP module (version 1.0.84) is selected. A right-click context menu is open, showing options like View, Edit, Inspect, Locations, Copy, Paste, Duplicate, Export, Delete, and Update. The 'Duplicate' option is highlighted.</p>
2	<p>Select Replace to replace InterlockON2GP by Iterlock8ONGP for both UC and CD, the upper layers will get automatically duplicated.</p>  <p>The screenshot shows the '\$MotorGP (1.0.67) : Duplicate' dialog box. The 'New Identifier' is set to 'Custom_'. The 'Action' column shows 'Replace' for the UC and CD templates. The 'Usability State' is set to 'Approved'.</p>
3	<p>Open Custom_MotorGP template and edit the Custom_MotorGP_UC control composite.</p>
4	<p>Right-click the Interlock8ON_UC composite > Defer, and select the All unbound to defer the parameters. Then, update all the parent templates within the Custom_MotorGP_UC template.</p>

Step	Action
	 <p>A screenshot of a software interface showing a context menu for the 'Interlocks' component. The menu is open, displaying various options such as 'Switch to Required', 'Defer', 'Optimize Bindings', 'Hide Unchecked', 'Hide Unbound', 'Inspect', 'View', 'Edit', 'Go to', 'Copy', 'Update', 'Replace', 'Delete', 'Rename', and 'Properties'. The 'Interlocks' component is highlighted in the background, and the menu is overlaid on it.</p>
5	<p>Right-click the \$MotorDataGP and Edit the interface.</p>  <p>A screenshot of a software interface showing a context menu for the 'MotorStatus' component. The menu is open, displaying various options such as 'Switch to Optional', 'Optimize Bindings', 'Show Unchecked', 'Hide Unbound', 'Inspect', 'View', 'Edit', 'Go to', 'Copy', 'Update', 'Replace', 'Delete', 'Rename', and 'Properties'. The 'MotorStatus' component is highlighted in the background, and the menu is overlaid on it.</p>
6	<p>Right-click the ILCKGRPStatus interface and replace with ILCK8GRPStatusGP interface and click OK in the confirmation dialog box.</p>

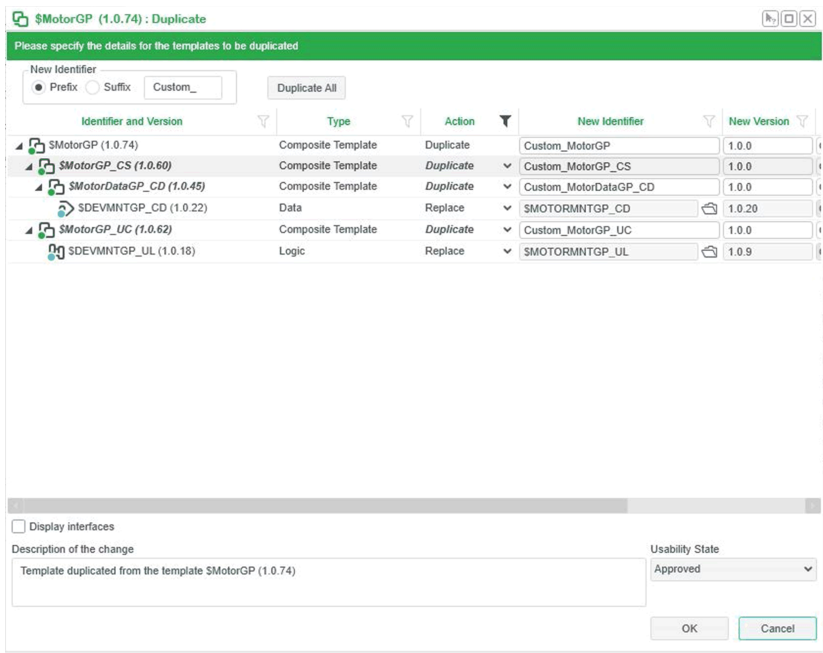
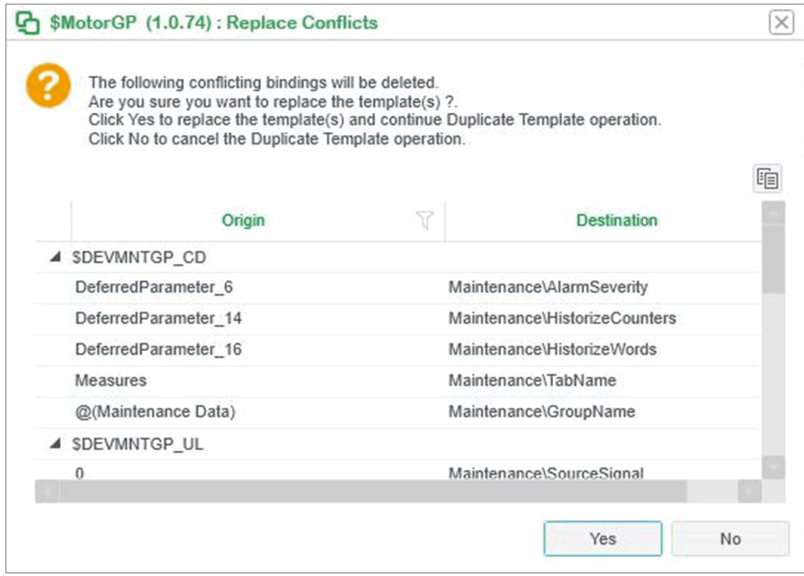
Step	Action
7	<p>Select Other button and save MotorDataGP interface without \$.</p>
8	<p>In Custom_MotorGP_UC, replace \$MotorDataGP by MotorDataGP and save the Custom_MotorGP_UC.</p>
9	<p>Open Custom_MotorGP_CS composite and edit the Custom_MotorDataGP_CD supervision composite.</p>
10	<p>Right-click the Interlock8ON_CD composite > Defer, and select the All unbound to defer the parameters. Then, update all the parent templates within the Custom_MotorGP template.</p>
11	<p>In Custom_MotorDataGP_CD, replace \$MotorDataGP by MotorDataGP and save Custom_MotorDataGP_CD.</p>

Step	Action
	
15	In Custom_MotorDataGP_CD, right-click the MotorStatus parameter > select Create Extended and rename the interface as MotorStatus.
16	Defer the Custom_MotorDataGP_CD and ensure that all the facet parameters are deferred till the top level.
17	In control module, Defer the control and supervision composite. Then, delete the parameters EPEFORASP, MotorSC in control composite and GroupSC in supervision composite.
18	Save the template with new build version.
19	Update the control module with the changes.

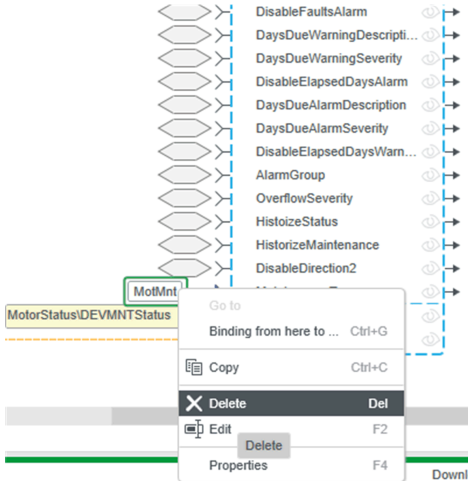
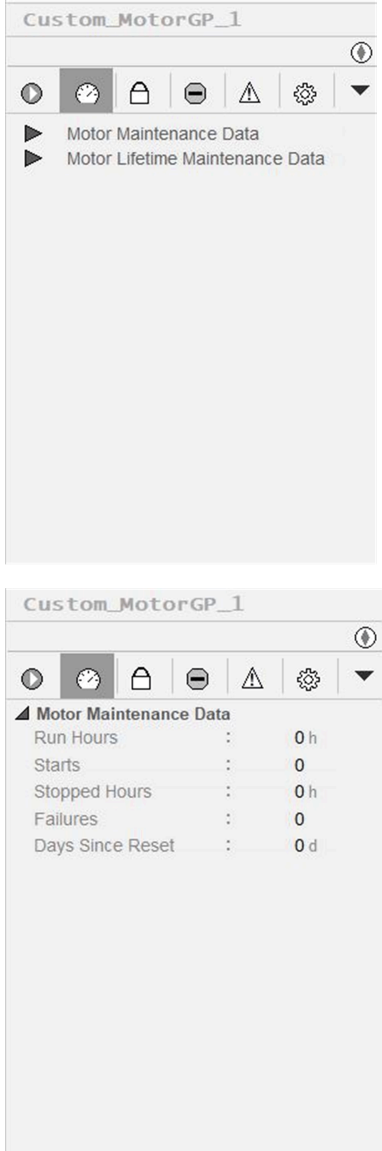
Step	Action
20	Drag and drop the template in application level, generate the control project and supervision project and to build, deploy the control project to controller.
21	<p>In the runtime, all the four groups will be available on the faceplate as shown in the following figure.</p> 

Replacing the Default Maintenance Service by a Different One

This section details the steps to replace the existing maintenance service by advance device maintenance service.

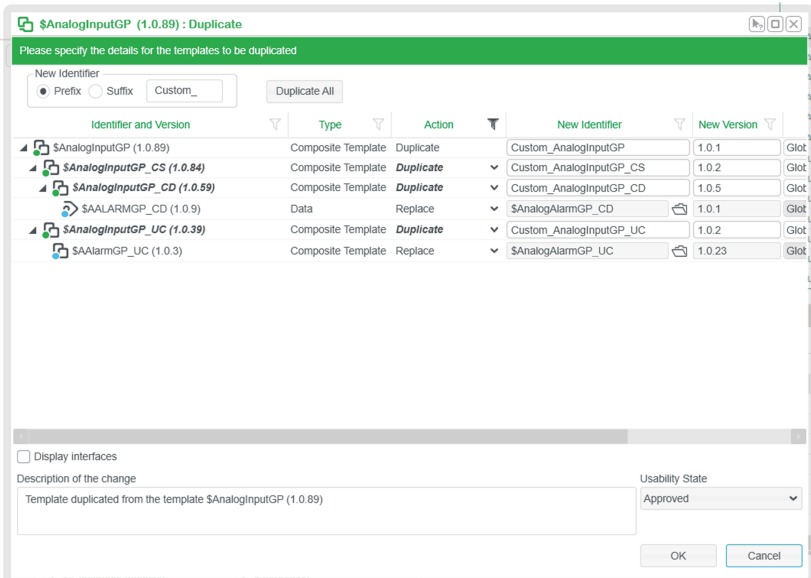
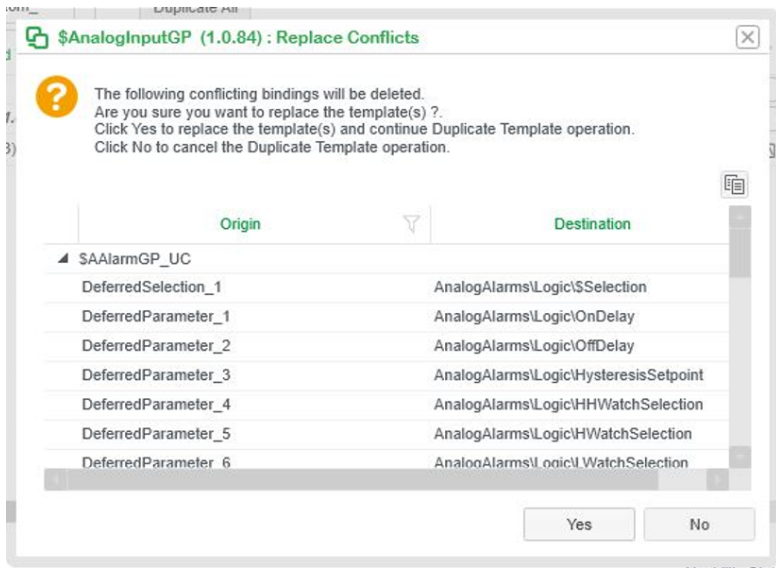
Step	Action
1	<p>Select duplicate option on the \$MotorGP Control Module.</p> <p>Select replace option for DevmntGP_UL by MotormntGP_UL for both UC and CD and the upper layers will get automatically duplicated.</p> 
2	<p>After clicking OK a message box about conflicting bindings opens. Click Yes to proceed.</p> 
3	<p>Edit the Custom_MotorGP_UC template and right-click \$MOTORMNTGP_UL and select Defer > All Unbound. Then, update all the parent templates within the Custom_MotorGP template.</p>

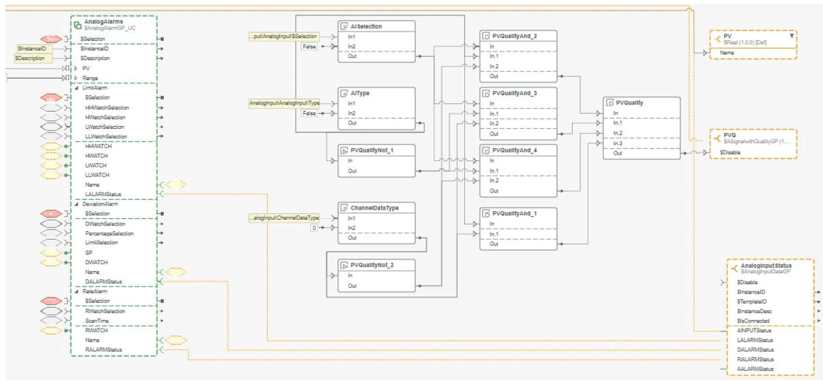
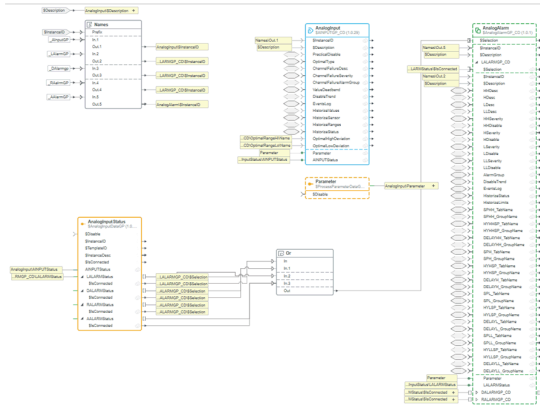
Step	Action
4	<p>Edit the Custom_motorgrp_CD template and right-click \$MOTORMNTGP_UL and select Defer > All Unbound and update the template as show in the following figure.</p> <p>Delete the input value of maintenance type interface and then defer it.</p>

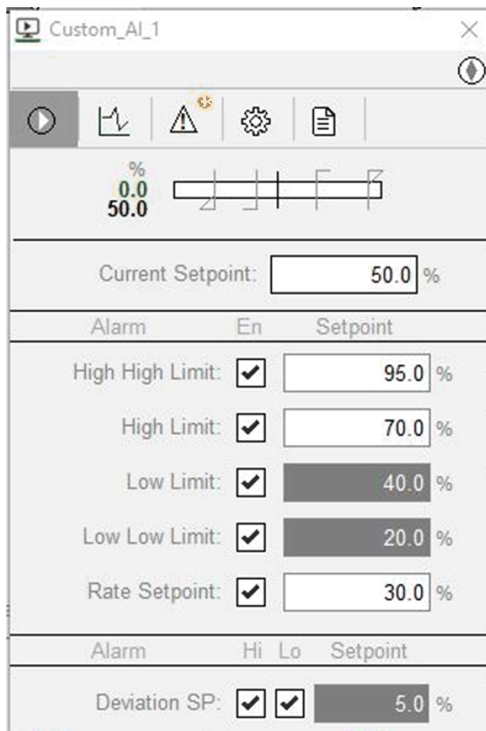
Step	Action
	
5	Update the control module with the changes.
6	<p>Drag and Drop the template in application level.</p> <p>Generate the control project and supervision project and build.</p> <p>Deploy the control project to controller.</p>
7	<p>In the runtime, two groups will be available on the faceplate as shown in the following figure.</p> 

Replacing with an AAlarm with Analog Alarms (LAlarm, DAlarm, RAlarm)

This section details the steps to change the existing maintenance service to advance device maintenance service.

Step	Action
1	<ul style="list-style-type: none"> Select Duplicate on the \$AnalogInputGP control module. Select Replace for AAlarmGP_UC by AnalogAlarmGP_UC for both UC and CD, the upper layers are automatically duplicated. 
2	<ul style="list-style-type: none"> After clicking OK, a message box about conflicting bindings opens. Click Yes to proceed. 
3	<ul style="list-style-type: none"> Edit Custom_AnalogInputGP template connect the Lalarm Status, Ralarm Status, Dalarm Status Pins to the corresponding interface. Defer the remaining pins in AnalogAlarms and update the template to top layer.

Step	Action
	
4	<ul style="list-style-type: none"> • Edit the Custom_AnalogInputGP_CD template and drag the LalarmGP_CD, RalarmGP_CD, DalarmGP_CD templates onto the workspace. • Connect and defer the links as shown in the following image. • Enter the prefix name _LalarmGP, _RAlarmGP, _DAlarmGP for concatenation as shown in the following image. • Enter the Input value for Alarm type interface as shown below. 
5	<ul style="list-style-type: none"> • Update the control module with the changes.

Step	Action																										
6	<p>Instantiate the template in application level.</p> <p>Generate the control and supervision projects and build.</p> <p>Deploy the control project.</p>																										
7	<p>In the runtime, all the four groups will be available on the faceplate as shown in the following figure.</p>  <p>The screenshot shows a faceplate titled "Custom_AI_1". It features a toolbar with icons for play, graph, alarm, settings, and document. Below the toolbar is a scale from 0.0 to 50.0 with a slider. The "Current Setpoint" is displayed as 50.0 %. Below this is a table of limits:</p> <table><tr><th>Alarm</th><th>En</th><th>Setpoint</th></tr><tr><td>High High Limit:</td><td><input checked="" type="checkbox"/></td><td>95.0 %</td></tr><tr><td>High Limit:</td><td><input checked="" type="checkbox"/></td><td>70.0 %</td></tr><tr><td>Low Limit:</td><td><input checked="" type="checkbox"/></td><td>40.0 %</td></tr><tr><td>Low Low Limit:</td><td><input checked="" type="checkbox"/></td><td>20.0 %</td></tr><tr><td>Rate Setpoint:</td><td><input checked="" type="checkbox"/></td><td>30.0 %</td></tr></table> <p>Below this table is another section with a table:</p> <table><tr><th>Alarm</th><th>Hi</th><th>Lo</th><th>Setpoint</th></tr><tr><td>Deviation SP:</td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td>5.0 %</td></tr></table>	Alarm	En	Setpoint	High High Limit:	<input checked="" type="checkbox"/>	95.0 %	High Limit:	<input checked="" type="checkbox"/>	70.0 %	Low Limit:	<input checked="" type="checkbox"/>	40.0 %	Low Low Limit:	<input checked="" type="checkbox"/>	20.0 %	Rate Setpoint:	<input checked="" type="checkbox"/>	30.0 %	Alarm	Hi	Lo	Setpoint	Deviation SP:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5.0 %
Alarm	En	Setpoint																									
High High Limit:	<input checked="" type="checkbox"/>	95.0 %																									
High Limit:	<input checked="" type="checkbox"/>	70.0 %																									
Low Limit:	<input checked="" type="checkbox"/>	40.0 %																									
Low Low Limit:	<input checked="" type="checkbox"/>	20.0 %																									
Rate Setpoint:	<input checked="" type="checkbox"/>	30.0 %																									
Alarm	Hi	Lo	Setpoint																								
Deviation SP:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5.0 %																								

Index

A

alarms	
alarms	78
analog values	
analog values	39

C

configure	
configure and run a supervision project	83
control modules	
control modules	43

D

datalogging	
datalogging	80

F

faceplate	
faceplate	64

G

genie	
genie	58

I

Inoperable Asset	
Inoperable Asset	32
interlocks	
interlocks	33
items	
items	82

M

maintenance	
maintenance	36
modes	
modes	29

N

Non-Confirmed Operation	
Non-Confirmed Operation	31

O

optional services	22
output	
output	30
owner/setpoint	
Owner/Setpoint	27

R

restart	
---------	--

restart	35
---------------	----

S

security	
security	76
situational awareness	
situational awareness	38
supervision	
Supervision	86

T

template	
template	98
template structure	13
templates	
alternate services	23
library structure	13
project standards	24

W

workspace	
workspace	44

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.

© 2023 Schneider Electric. All rights reserved.

EIO0000004045.05