



Process Expert - General Purpose Library Classic

Equipment Module Templates Reference Manual

EIO0000003016.06
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
Equipment Module Templates.....	12
Composition Strategy for Application Templates.....	12
Naming Convention	14
List of Equipment Module Templates	19
Common Services	20
Accessing General Purpose Library User Guides and Technical Documents using RTNS Feature	20
Hyperlink Services.....	21
Equipment Module Management	22
EMPattern - Equipment Module Pattern	22
Modifying the EMPattern Template	27
Pump Set Pattern Management	35
PumpSetPattern - Pump Set Pattern Equipment Module.....	35
Modifying the PumpSetPattern Template	43
Flow Control Pattern Management	48
FlowCtlPattern - Flow Control Pattern Equipment Module.....	48
Modifying the FlowCtlPattern Template	56
Index	57

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 Equipment Module templates, and the Control and Supervision services that they provide.

For a more detailed description of their associated Control and Supervision services and configuration parameters, refer to the user guides mentioned in this document.

To use Equipment Module templates, you need to have knowledge of EcoStruxure Process Expert, and of the Control and Supervision Participants.

Validity Note

This document has been updated for the release of EcoStruxure™ Process Expert 2023.

Related Documents

The characteristics that are described in the present document, as well as those described in the documents included in the Related Documents section below, can be found online. To access the information online, go to the Schneider Electric home page www.se.com/ww/en/download/.

The characteristics that are described in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Title of Documentation	Reference Number
EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual	EIO0000003014
EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual	EIO0000003015
EcoStruxure™ Process Expert - General Purpose Library Classic Device Templates Reference Manual	EIO0000001308
EcoStruxure™ Process Expert - General Purpose Library Classic Communication Templates Reference Manual	EIO0000001311
EcoStruxure Process Expert User Guide	EIO0000001114
EcoStruxure Process Expert Global Templates Reference Manual	EIO0000001986 (eng)
EcoStruxure Process Expert Runtime Navigation Services User Guide	EIO0000001574 (eng)

Technical Support

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

Product Related Information

WARNING

LOSS OF CONTROL

- 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.

Standard	Description
	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

Equipment Module Templates

Description

The EcoStruxure Process Expert General Purpose library provides resources:

- That have been pre-configured and tested by Schneider Electric.
- That are designed for automating a large variety of processes.

Equipment Module (EM) templates model generic functionalities of process equipment but also hardwired signals from the field, and variables. They represent the application of the system.

They are used to implement the Control and Supervision services that are provided by the software Participants, streamlining the engineering of systems.

The resources providing these services are encapsulated in dedicated facet references, which are then organized in composite references.

At the highest level, the Equipment Module Pattern templates represents the Equipment module.

You can easily customize the services that are provided by Equipment Module templates to fulfill your specific requirements. You can also link Equipment Module templates with compatible process templates, using interfaces.

Equipment Module Template Services

Facet templates referenced by Equipment Module templates provide the following services:

Control	Includes core services plus additional, optional services, which you can activate if needed. Function blocks and variables are the resources that are encapsulated in these facet references to provide such services. Also, it includes the customizable facet to create user defined logic.
Supervision	These services complement the Control services. Supervision services are optional and those corresponding to selected Control are activated by default. Tags, alarms, and genies are the resources that are encapsulated in these facet references to provide such services. Data is provided by the associated Control resources.

For both Control and Supervision services, you can configure parameters and also, you can duplicate the template and modify the Control and Supervision services to configure parameters to meet the requirements of your system during instantiation.

Composition Strategy for Application Templates

General

The following general rules are applied when designing application library templates:

- Decoupling between participants
- Coupling into the participant
- Composition approach
- Exposed interfaces

Decoupling Between Participants

Maximum decoupling between facets from different participants is expected. One participant does not need to know the internal implementation of the other one to be able to collaborate.

For instance, the Control Expert logic facets expose the OPC items to be required for the Citect SCADA data facets in order that the Citect SCADA data facets are not altered as a result of any change in the Control Expert logic facet data structures (DDT field names). So, the interfaces between participants contain so many items as required to respect this rule.

Coupling into the Participant

To minimize the data to be exchanged between facets from the same participant, the library designer can assume the details of the data structure that is shared between the facets by means of the interfaces.

For instance, the Citect SCADA Genie facets can assume which is the naming convention being implemented for Citect SCADA tags into the Citect SCADA data facets. That means that, by sharing the name of the object being implemented through the interface, the genie can assume the name of tags involved.

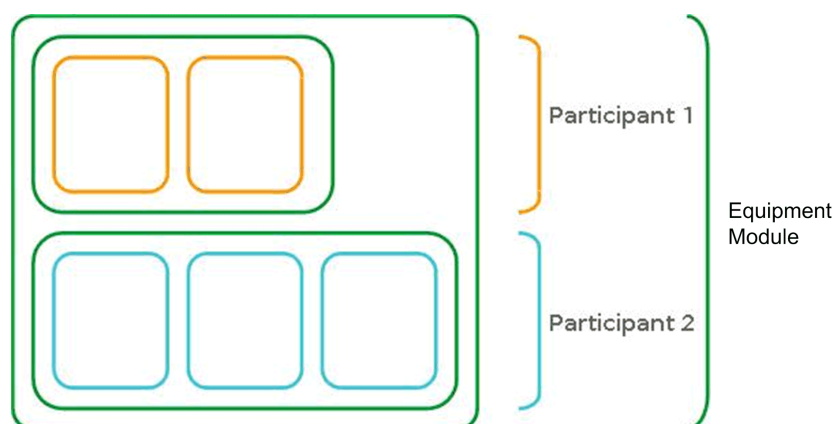
But, if the templates being designed could be used directly from the system application and linked by means of interface links, you need to provide the right connectivity between them (the user making the system application can only make interface links for establishing the relationship between different instances; whereas, the library designer has the opportunity to access the elements into the interfaces).

Composition Approach

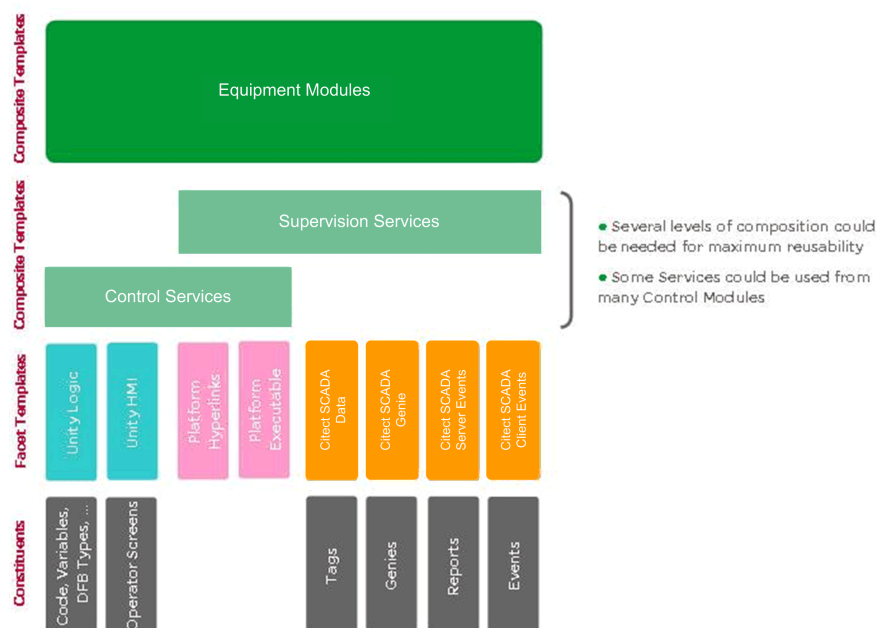
The objective is to provide composite templates which bring the features required for a control module (as defined in ISA-S88) that involves one or more participants when needed. Avoid the use of design composite templates with too many nested levels to avoid big changes propagation and to facilitate their maintainability. The libraries provide facet and composite templates which could be used to create instances or as part of other composite templates. This modular design allows the user to instantiate such templates in a consistent way for creating the application, independent of the participants involved and the required services (optional elements in the composite templates).

The first level of encapsulation is by participant. So, a horizontal approach is implemented. Composite templates offer the functionality from a concrete participant that can be combined for providing the complete functionality expected for a type of control module (that is, Motor 1 speed 1 direction).

The following figure shows general composition approach.



The following figure shows resources for application templates.



Another aspect to be considered is the perimeter for the control module templates.

The following capabilities are encapsulated by means of independent control module templates that the user can link in instantiation time:

- Foundation templates: Hardware Projection, Hardware Diagnosis, Peer to Peer, and so on.
- Communication templates.
- Device templates.
- Process templates: Optionally the Process templates include the projection of their related hardwired signals to facilitate the Equipment Module of creation of hardwired signals with a minimum effort.
- Equipment Module templates: These are open templates so user can create his own logic and configure the parameters as per system application requirement.

Naming Convention

General

A key point for harmonization of the templates is to use a standard naming convention.

The general rules are applied depending on the following type of template being defined:

- Any types of templates
- Any interface models
- Mapping interface models
- Application interface models
- Application facet templates
- Application composite templates

Any Types of Templates

The following rules are applied to any types of templates:

- **Prefix:** The character \$ is used as a prefix for templates that are not meant to be modified and templates without \$ prefix are meant to be modified.
- **Aliases:** The objects model is bringing the possibility to use aliases to hide the origin of the deferred selections, parameters and interfaces. In the standard library, we are not using such capability. So the user creating references (Definition time) or instances (Instantiation time) will see the real name of the related selections, parameters, and interfaces.
- **Documentation:** Templates need to include references for enabling the access to their related engineering documentation (that is, user manuals). Such documentation needs to be attached to the related function (that is, `EMPatten` explanation as part of the composite where the Control Expert Logic and the Control Expert HMI facets are encapsulated).

Even when such facets are not explicitly in some of the drawings included in this document, they are considered implicitly as included.

- **Constituent names:** To allow using multiple instances of the same template into the same participant project, it is commonly required to generate unique identifiers. The strategy to be applied is to add suffixes as in EcoStruxure Process Expert (for instance, `EM1_EMCTL`, `EM1_EMCTL_ST`, `EM1_IC_CONDSUM`, and so on). That is also confirming the compatibility with previously created resources (that is, Smart Genies for representing the Equipment Module implemented through the Control function `EMCTL`).

Verify that the templates are establishing similar suffixes for avoiding the names conflicts. The designed templates force the user to use the same naming convention when creating new composite templates. The designed templates are created by recombining the ones coming from the standard libraries. This process increases their reusability even for the standard templates.

So, the constituents names are calculated by means of bindings by the following rules:

- The root of the constituent names is generated by using the name of the instance entered by the user.
- The suffixes need to be added only when the fact of not using them is creating a name conflict at the template level that is being defined. This means that establishing suffixes need to be delayed as much as possible to the top composition levels.
- The suffixes should follow the naming convention applied in EcoStruxure Process Expert and need to be useful for understanding the role of the constituents in the concrete scenario.

For instance:

- The user creates an instance of the composite template `$EMPattern` and enters the name `EM1` for it. This is reducing (but not avoiding) names conflicts (in fact, they can be only avoided by the participant in generation time regardless the chosen naming convention).
- This name is propagated to encapsulated templates through bindings without any suffixes up to the level where that is causing a conflict (see the example in the next point).
- When combining the Control functionality for Equipment Module (`EMCTL`) and the detected failures management (`CONDSUM`), it is required to concatenate the related suffixes (`EM1_EMCTL` and `EM1_FC_CONDSUM`) to avoid creation of 2 DFBs with the same name, which is not allowed in Control projects.
- When the facet defines the required constituents for implementing the control part of the object (`EMCTL`), it is required to concatenate additional suffixes (`EM1_EMCTL_ST` for the status data, `EM1_EMCTL_CFG` for the configuration data) for avoiding to create duplicate names.
- In such case, if the user decides to instantiate directly the last mentioned facet template, it provides that no generated objects have duplicate names (into the scope of the instance). The template does not force the user to use the complete naming convention. For instance, `EM1 (EMCTL)`, `EM1_ST` and `EM1_CFG` rather than `EM1_EMCTL`, `EM1_EMCTL_ST` and `EM1_EMCTL_CFG` generated when instantiating the template `$EMPattern`.

Any Interface Models

The following rules are applied:

- **Roles representation:**
 - Unidirectional interfaces: Female representation is used for the role in charge of producing the values for elements whereas Male is used for the other role.

Mapping Interface Models

The following rules are applied:

- Role names:
 - CO: Is used from the application templates representing communications objects (CO) that receive the data provided from the system configuration. This role is represented graphically as male.
 - DO: Is used from the application templates representing device objects (DO) that receive the data provided from the system configuration. This role is represented graphically as male.
 - HO: Is used from the topological template (also called Hardware Objects (HO)) to expose the information provided by the hardware (for example, topological addresses, slave number, and so on) to the project facets. This role is represented graphically as female.
 - SO: Is used from the application templates representing the signal objects (SO) or I/O module objects that receive the data provided from the system configuration. This role is represented graphically as male.

Application Interface Models

As the same interface models can be used for several purposes and the objective of this key mechanism is to implement the collaboration between participants, their names need to be as generic as possible.

The naming convention is conditioned by the nature of the interface models.

The following types of interface models are considered for the standard library:

- Elementary interface models: The more elementary interface models are for exchanging just one element of an elementary type (not a nested Interface). These interfaces are for a generic usage as the interface name (the unique identifier) determines the type of the transported data.
- Single element interface models: Interfaces that contain just one element for a specific usage are named as the constituent type from the participant.
- Multiple elements interface models: Interfaces that contain several elements with a common objective. Instead of having several elementary interface models, one new interface model is created with many elements as needed.

The following rules are applied depending on the types:

- Any types of interface models
- Elementary interface models
- Single element interface models
- Multiple element interface models

Any Types of Interface Models

The following general rules are applied to any interface models:

- Interface model name: It is recommended to use upper and lowercase for better readability. The exception is when we use names that are directly used by the participant. In such case, the exact name is used.

- Role names:

Definition and references: For example, a template is used for defining some constituents (that is, definition of Control variables) that could be used by one or more similar participants which need to access to such constituents (that is, references to Control variables).

In such case, the role names to be used are:

- Definition: Is for the role in charge of defining the constituents. This role is represented graphically as female.
- Reference: Is for the role in charge of accessing (it does not matter for which kind of access: read, write, read/write, and so on) to such constituents. This role is represented graphically as male.

Elementary Interface Models

- Interface model name: The name of the real-time data type being exchanged is directly used for identifying the interface model (that is, \$Bool for exchanging the name of a boolean constituent). It is required to limit the usage of the data being exchanged.

For instance, to a concrete participant, they could be added some suffixes for avoiding. Such interface models could be used outside of its expected scope.

- Element name: The element is identified as name.

Single Element Interface Models

- Interface model name: The name needs to be identical to the type of the data being exchanged as defined in the participant (that is, \$EMCTLName for exchanging the name of a Control variable of type \$EMPATTERN).
- Element name: The element is identified as name.

Multiple Element Interface Models

- Interface definition name: The name should reflect the meaning of the set of elements (that is, \$EMPATTERNStatus for exchanging the names of the OPC items in the Control Expert Logic which are needed for accessing from Supervision).

Application Facet Templates

As much as possible, the name of the facet templates needs to be closer to the name of the encapsulated constituents or elements. As there could be several facet templates addressing the same functionality but from different points of view (for instance, different facet types) of the participant, consider the possibility to add some suffixes for distinguishing them.

- Suffixes: The following suffixes are used for identifying the templates:
 - _UL for Control Expert logic facet templates.
 - _UH for Control Expert HMI facet templates.
 - _CD for Citect SCADA data facet templates.
 - _CG for Citect SCADA genie facet templates.
 - _CR for Citect SCADA server event facet templates.
 - _CC for Citect SCADA client event facet templates.
 - _PH for platform Hyper link facet templates.

Application Composite Templates

As the composite templates are used to combine functionality provided by other facet or composite templates, it should reflect the complete functionality being encapsulated.

Basically, there are 2 use cases:

- **Equipment Module:** Typically the composite template combines functionality from several participants for representing the complete functionality required for a type of Equipment Module (that is, `EMPattern` - Equipment Module Pattern). In such case, the type of Equipment Module is abbreviated to determine the name of the template. It is recommended to use upper and lowercase for maximizing readability (that is, `$EMPattern`).
- **Equipment Module services:** The contained functionality is provided by the same participant for covering some services required for implementing one or more types of Equipment Module. In such case, a suffix is added for expressing their scope (that is, the suffix `_UC` for the template `$EquipmentModulePattern_UC` is expressing that it encapsulates functionality required from the perspective of the Control for implementing user defined Equipment Module logic).

Suffixes: The following suffixes are used for identifying the templates:

- `_UC` for Control Expert logic composite templates.
- `_UH` for Control Expert HMI composite templates.
- `_CS` for Citect SCADA data composite templates.
- `_CG` for Citect SCADA genie composite templates.
- `_CR` for Citect SCADA server event composite templates.
- `_CC` for Citect SCADA client event composite templates.
- `_CS` for Citect SCADA (data and genie) composite templates.

List of Equipment Module Templates

List of Families

The Equipment Module templates described in this document are grouped by family.

The table lists the templates of Equipment Module family:

Family name	Equipment Module templates	Description
Equipment Module	<code>EMPattern</code> , page 22	Equipment Module Pattern
	<code>PumpSetPattern</code> , page 35	Pump Set Pattern Equipment Module
	<code>FlowCtlPattern</code> , page 48	Flow Control Pattern Equipment Module

Common Services

Overview

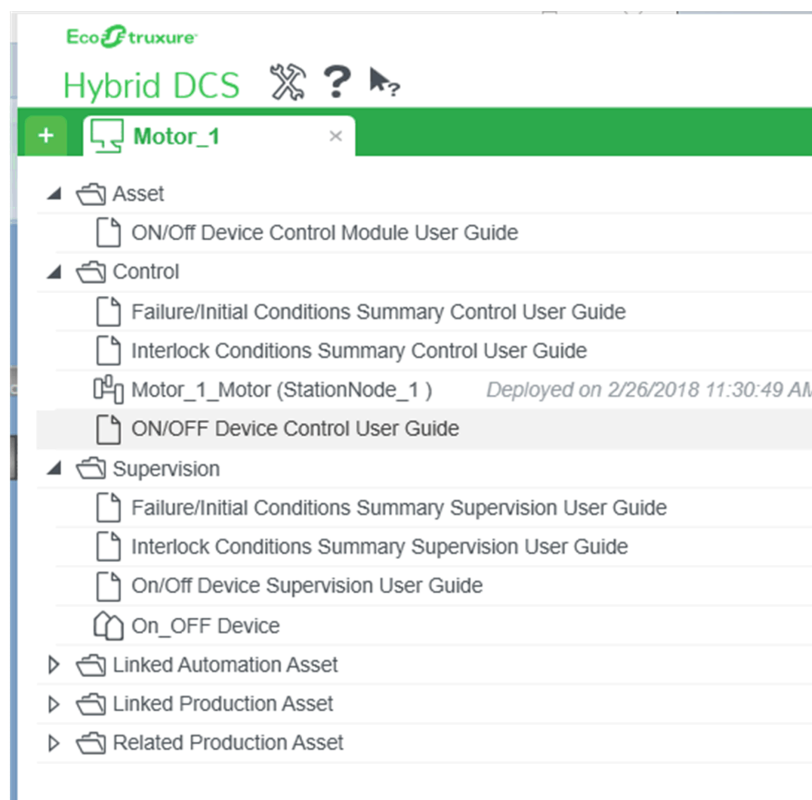
This chapter describes the different services that are common to the control modules.

Accessing General Purpose Library User Guides and Technical Documents using RTNS Feature

General Description

The General Purpose Library user guides and technical documents can be accessed using the Runtime Navigation Services (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide) provided by the EcoStruxure Process Expert.

In the Operation Client, the user guides are segregated as shown in the image below.

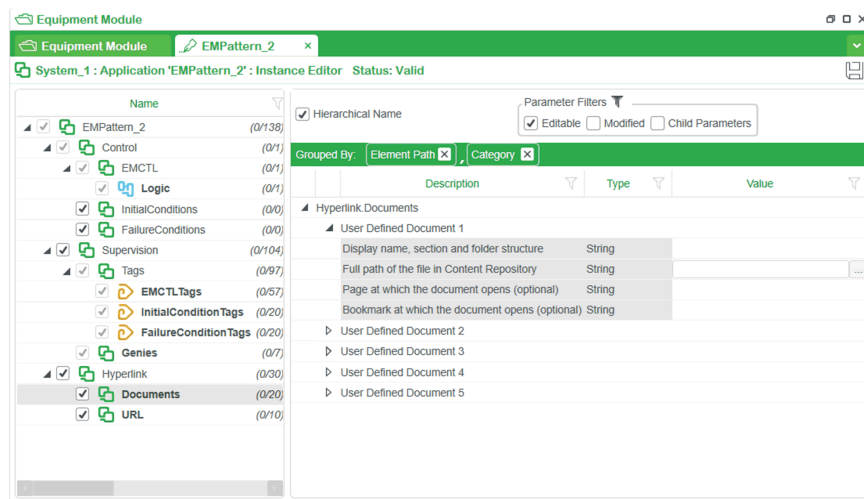


NOTE: The user guides of the optional Services (for example, *Failure Conditions*, *Interlock Conditions*) will be available in their respective sections (**Asset**, **Control** or **Supervision**) if you have enabled the optional services in the EcoStruxure Process Expert application.

Hyperlink Services

General Description

The General Purpose Library control modules allows you to link the documents and URL to its instances. The hyperlink service is disabled by default, you have to enable the hyperlink service and configure the parameters as shown in the below screen shot. For more details refer to the topic describing how to use the hyperlink service (see EcoStruxure™ Process Expert, Runtime Navigation Services , User Guide).



NOTE: A maximum of five documents and five URLs can be added to each instance.

Equipment Module Management

Overview

The Equipment Module Pattern (`EMPattern`) template is an open template which can be adjusted to suit the specific needs of Equipment Module Pattern being implemented.

⚠ WARNING

LOSS OF CONTROL

- 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.

EMPattern - Equipment Module Pattern

General Description

The `EMPattern` template is used to monitor and manage equipment module patterns.

The composite and facet names without the \$ prefix can be modified, while the composite and facet names with \$ prefix are standard and cannot be modified.

These templates do not reflect any specific installation.

Parameters

Configuration

The table describes the **Configuration** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Logic	AutoResetDis	Boolean	False	If True , auto reset is disabled.
Genies	EMCTL_x	Boolean	False	1 = Hides the corresponding button on the genie and faceplate. x represents the name of the command. For example, EMCTL_STOP .

Basic Alarm Configuration

The table describes the **Basic Alarm Configuration** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description					
EMCTLTags	Name_x	String	-	Name_x : Name of the physical device associated with the alarm. Corresponds to the <i>Alarm Name</i> property of advanced alarms of the Supervision Participant. For example, Name_IDLE is for EMCTLTags . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.					
EMCTLTags	Category	String	SGC_CAT_EVENT	Category of alarm/event for all states. Alarm category label or number. Corresponds to the category property of advanced alarms of the Supervision Participant.					
EMCTLTags	Help	String	-	Help of alarm/event for all states. Name of the graphics page that opens when the help function is called. Corresponds to the <i>Help</i> property of advanced alarms of the Supervision Participant.					
<table border="1"> <tr> <td>X</td><td colspan="4"> Suffix of the variable tag name or advanced alarm tag that is managed by the corresponding Supervision components. For example, Name_IDLE correspond to the <i>EM Name_EMCTL_IDLE</i> advanced alarm tag managed by the <i>EMCTL</i> Control resource. Refer to the description of tags of the Supervision component (see EcoStruxure™ Process Expert LVPC Library, Supervision Services User Guide). </td></tr> </table>					X	Suffix of the variable tag name or advanced alarm tag that is managed by the corresponding Supervision components. For example, Name_IDLE correspond to the <i>EM Name_EMCTL_IDLE</i> advanced alarm tag managed by the <i>EMCTL</i> Control resource. Refer to the description of tags of the Supervision component (see EcoStruxure™ Process Expert LVPC Library, Supervision Services User Guide).			
X	Suffix of the variable tag name or advanced alarm tag that is managed by the corresponding Supervision components. For example, Name_IDLE correspond to the <i>EM Name_EMCTL_IDLE</i> advanced alarm tag managed by the <i>EMCTL</i> Control resource. Refer to the description of tags of the Supervision component (see EcoStruxure™ Process Expert LVPC Library, Supervision Services User Guide).								

Advanced Alarm Configuration

The table describes the **Advanced Alarm Configuration** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
EMCTLTags	Privilege	String	Label with the syntax: <i>SGC_PRIV_<type of component>_<type of action></i>	Privilege level required to acknowledge or disable the alarm. Corresponds to the <i>Privilege</i> property of advanced alarms of the Supervision Participant. For example, <i>SGC_PRIV_EVENT</i> . NOTE: For advanced alarms that are process events in the label <i><type of component></i> and <i><type of action></i> are replaced by EVENT , for example SGC_PRIV_EVENT .

Historize

The table describes the **Historize** parameters that you can configure:

Element Name	Name	Type	Default value	Description
EMCTLTags	Historize_x	Boolean	False	Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . x represents the child elements of the <i>EMCTL_ST</i> and <i>EMCTL_CFG</i> data structures, and the <i>EMCTL.CSTEPPD</i> variable. For example, Historize_STW .

Element Name	Name	Type	Default value	Description
	Historize_y			Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . y represents the Strategy execution state alarms. For example, Historize_IDLE . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
InitialConditionTags and FailureConditionTags	Historize_BYPASSW	Boolean	False	Enable historize bypass word.
	Historize_CONDW	Boolean	False	Enable historize condition word.
	Historize_REARMREQW	Boolean	False	Enable historize rearm word.

Alarm Tag Disable

The table describes the **Alarm Tag Disable** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
EMCTLTags	Disabled_x	Boolean	True	Corresponds to the <i>Alarm Tag Disable</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Alarm Tag Disable</i> property of the corresponding tag to False. x represents the name of the state. For example, Disabled_RUNNING .
NOTE: Values of Boolean parameters are set by using check boxes: <ul style="list-style-type: none"> • Selected = True • Cleared = False 				

Variable Tag Disable

The table describes the **Variable Tag Disable** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
InitialConditionTags	BYPASSWordSelection	Boolean	False	Disable initial conditions bypass tags.
	REARMREQ-WordSelection	Boolean	False	Disable initial condition resetting (REARM) tags.
FailureConditionTags	BYPASSWordSelection	Boolean	False	Disable conditions bypass.
	REARMREQ-WordSelection	Boolean	False	Disable conditions manual resetting.
NOTE: Values of Boolean parameters are set by using check boxes: <ul style="list-style-type: none"> • Selected = True • Cleared = False 				

The following optional elements feature configurable parameters:

- **InitialConditionTags:** Messages associated to initial conditions, to appear on the Initial Conditions tab of the faceplate for Equipment Module Pattern.
- **FailureConditionTags:** Messages associated to diagnostic, to appear on the Diagnostic tab of the faceplate for Equipment Module Pattern.

Composition

The `EMPattern` template references composite and facet templates, which provide the following services:

Control	Core services, plus additional optional services, which you can activate if needed.
Supervision	These services complement the Control services. Supervision services are optional.

The table describes the services that are available from the `EMPattern` template and the corresponding facet, which implements the service:

Control services (EquipmentModulePattern_UC)	Corresponding facet template	Control service description	Associated Supervision services (EquipmentModulePattern_CS) *	Corresponding facet template	Supervision service description
Core services					
Logic	\$EMCTL_UL	Refer to the EMCTL DFB*.	EMCTLTags	EMCTL_CD	Refer to tags of Equipment Module management**.
-	-	-	Genies	\$EMPanel-State_CG	Refer to genie representation of Equipment Module Management**.
				\$EMPanel-WithButtons_CG	
				\$EMPanelAllData2_CG	
				\$EMPanelAllData4_CG	
EMLogic	EMLogic_UL	Refer to EMLOGIC DFB*.	Parameter-Tags	EMPAR_CD	Refer to tags of Equipment Module management**.
Optional Services					
InitialConditions	\$CONDSUM_UL	Refer to the CONDSUM DFB***.	InitialConditionTags	\$INIT-CONDSUM_CD	-
FailureConditions	\$CONDSUM_UL	Refer to the CONDSUM DFB***.	FailureConditionTags	\$CON-DSUM_CD	
* See EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual					
** See EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual					
*** See EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual					

NOTE: When you activate an optional Control service, the corresponding Supervision service is automatically activated, if available.

The control module template exposes the following interface.

Interface Identifier	Type/Role	Description
<code>EMCTLName</code>	<code>\$EMCTLName/Def</code>	Equipment module control name.

Below table shows the list of templates which are available for modification.

Name of Facets, Composites and Templates	Description	Comment
EMLogic_UL	To create user logic and configure input and output parameters for control component as per requirement.	To modify this facet template you have to Save As and rename it as per requirement.
EquipmentModulePattern_UC	If EMLogic_UL is modified then EquipmentModulePattern_UC has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
EMPAR_CD	To configure the input and output parameters for supervision component as per requirement.	To modify this composite template you have to Save As and rename it as per requirement.
EMCTL_CD	To configure the Custom1...8, Paging, PagingG parameters of alarms of Equipment Module states and strategy execution states.	To modify this composite template you have to Save As and rename it as per requirement.
EquipmentModulePattern_CD	To configure the Strategy and EMstate variables. Also, to configure Disable, Historize and Name parameters of alarms of Equipment Module states and strategy execution states. If EMPAR_CD or EMCTL_CD is modified then EquipmentModulePattern_CD has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
EquipmentModulePattern_CS	To configure the DATA1 to DATA4 parameters. The parameters DATA1 to DATA4 can be any input/output parameters which can be viewed in genie for monitoring purpose. If EquipmentModulePattern_CD is modified then EquipmentModulePattern_CS has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
EMPattern	Main Equipment Module Pattern template. If EquipmentModulePattern_UC or EquipmentModulePattern_CS is modified then EMPattern has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.

Input/Output Parameter Configuration for Equipment Module

To configure the Input/Output parameters, you can create the variable tag and use these variables for configuring the customized data type as per the requirement.

Below table show the various data types supported and their corresponding Control and Supervision data types.

Data Type (user)	Control Data Type	Value entered in the Variable Tag Eng Units Field	Variable Tag Data Type
Boolean	BOOL		DIGITAL
Duration	TIME	<D>	ULONG
Date/Time	DINT	<T>	ULONG
Enumeration	INT	<E>	INT
Numeric (Real)	REAL	Format (for example, 0.00)	REAL

Data Type (user)	Control Data Type	Value entered in the Variable Tag Eng Units Field	Variable Tag Data Type
Numeric (Signed integer)	INT	Format (for example, 0)	INT
Numeric (Double signed integer)	DINT	Format (for example, 00000)	LONG

Default Configuration available for EMPattern Template

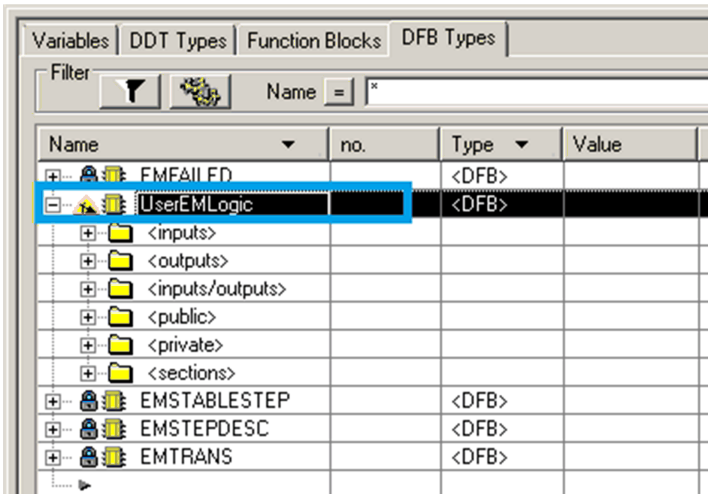
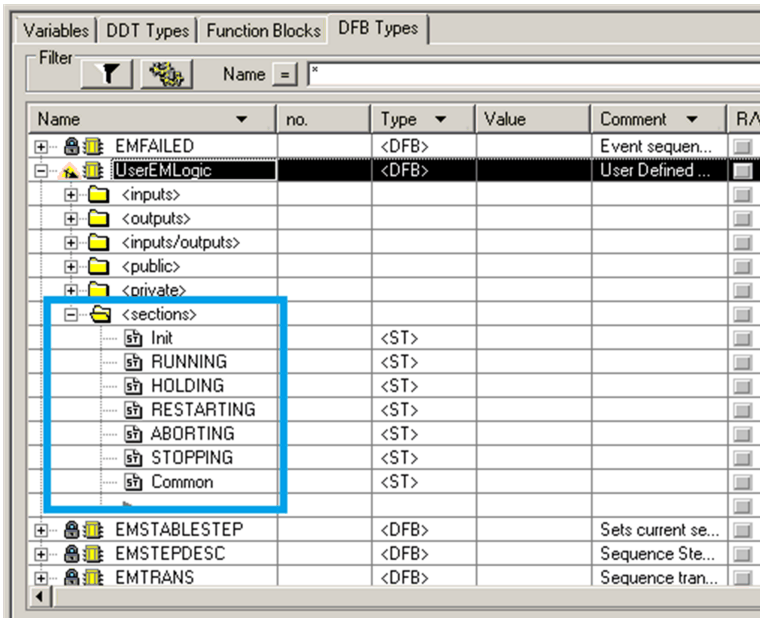
The EMPattern template is provided with the default Strategy, user defined Equipment Module States, input, output parameters and configurable DATA1 to DATA 4 parameters configuration for reference and better understanding of user so that the template can be modified as per its application requirements.

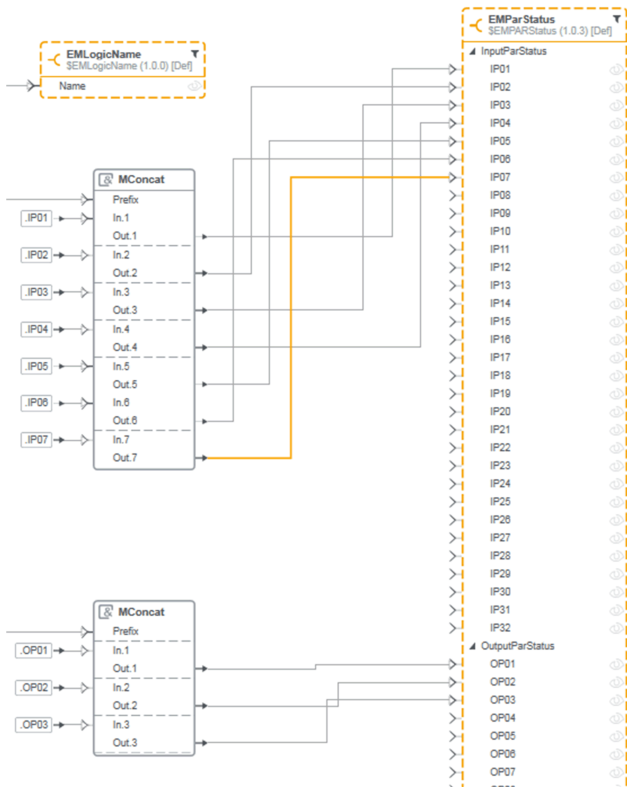
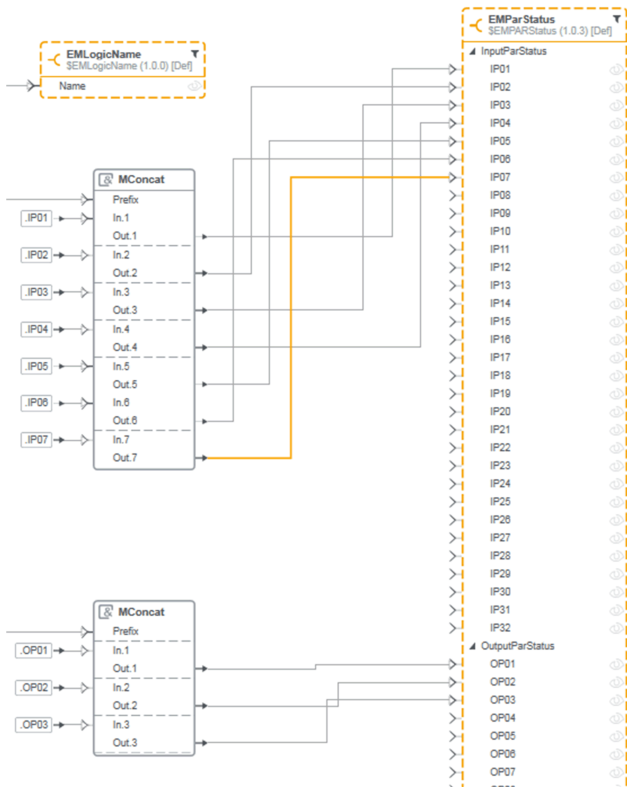
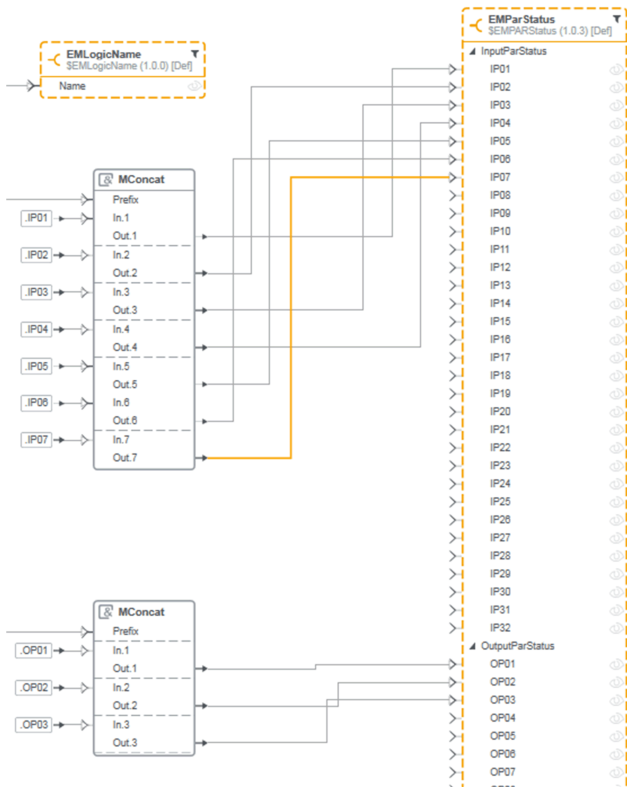
- The applicable parameters for default Strategy are (S1):1,2,3,4,5,6; @ (S2):1,3,5;@ (S3):2,4,6;@ (S9):2,3,4.
- The default configured user defined Equipment Module states are EMState16, EMState17, EMState18 and EMState24.
- Six input parameters are configured by default with the data types BOOL, DURATION, TIME, ENUM, INT and REAL. The input parameter with ENUM data type is configured with default values @ (Item1):1;@ (Item2):2;@ (Item3):3;@ (Item4):9.
- Four output parameters are configured by default with the data types REAL, DURATION, BOOL and TIME.
- The default value configured for DATA 1 to DATA 4 parameters are Output parameter 1 to Output Parameter 4 respectively.

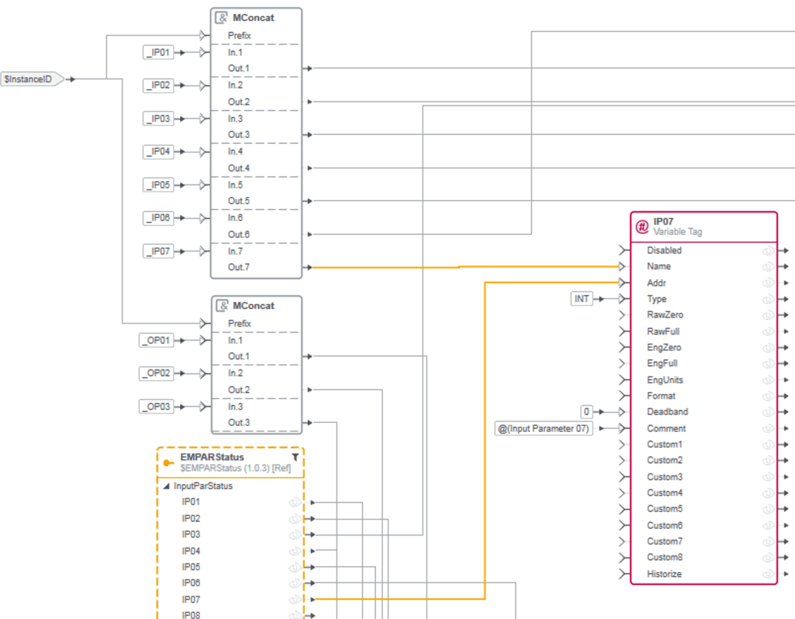
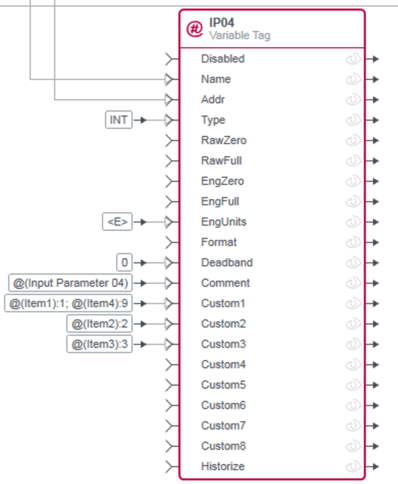
Modifying the EMPattern Template

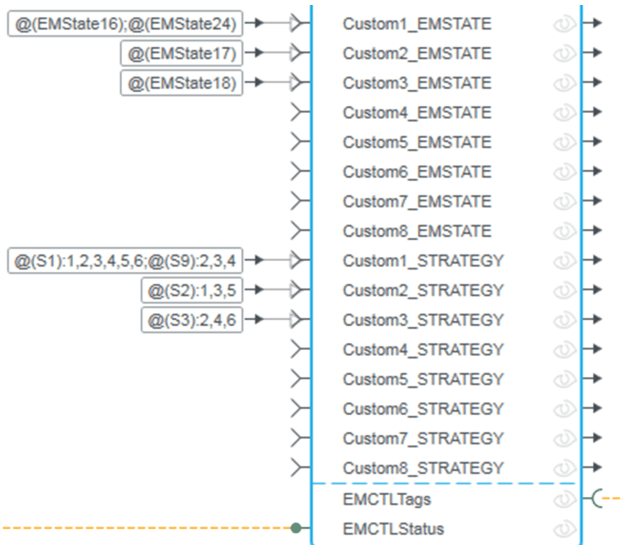
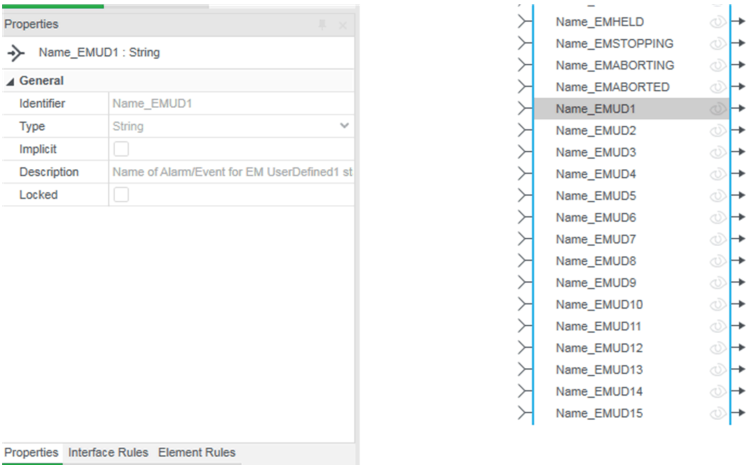
Steps to modify EMPattern template for the Equipment Module Operation


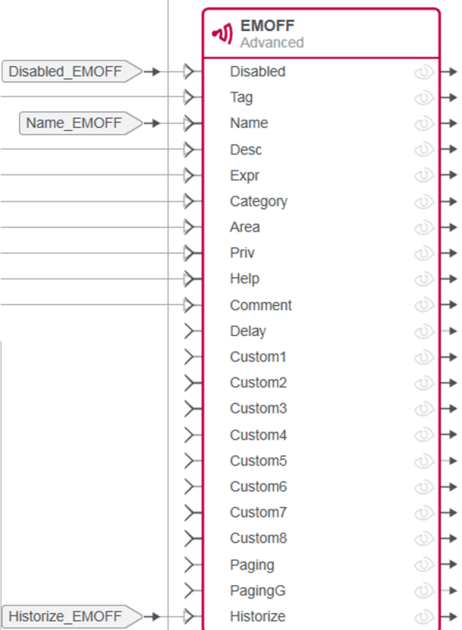
Step	Action
1	Navigate to the EMPattern template from Global Templates , this template is available under General Purpose Library folder with separate folder name as Equipment Module Patterns . Global Templates → General Purpose Library → Equipment Module Patterns → Equipment Modules .
2	You have to Save As the EMPattern template with different name (for example, UserEMPattern) so that the template can be used for specific needs.
3	You can write an own logic for specific requirement in the EMLogic_UL template in templatizer. Follow the path, UserEMPattern (newly created template from step 2) → EquipmentModulePattern_UC → EMLogic_UL).
4	Navigate to EMLogic_UL then go to templatizer, rename EMLogic (for example, UserEMLogic) DFB type as per the requirement.

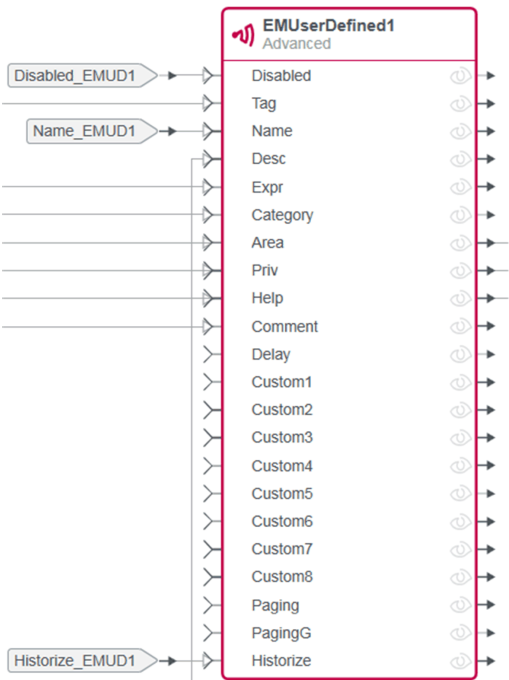
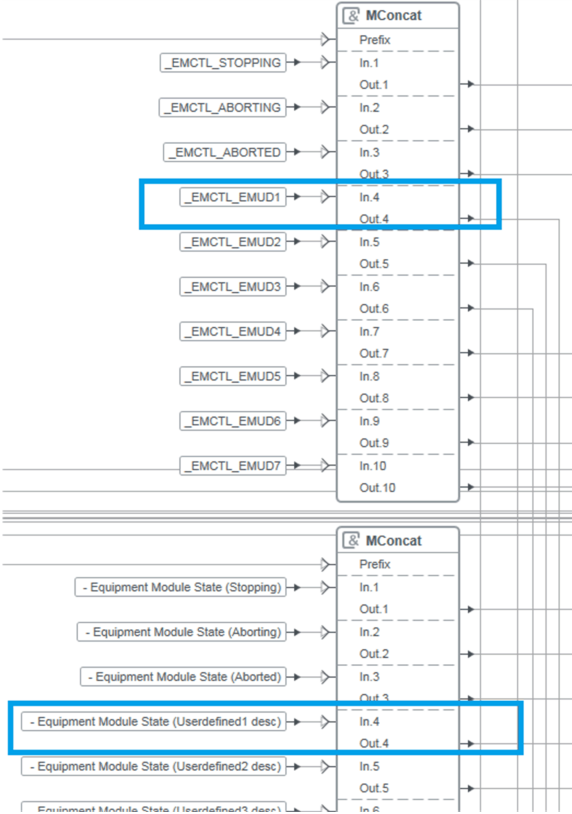
Step	Action
	 <p>You can import your <code>EMLogic</code> DFB into the templatizer. In this case if the DFB name is different than the existing name (<code>EMLogic</code>) then the links related to the DFB will be lost, you have to re-instantiate the new DFB and reconnect all the links to this DFB in the UL</p>
5	<p>You can write your own logic in the <code>EMLogic</code> DFB sections as shown below.</p> 
6	<p>It is recommended to rename the DDT Type <code>EM_IOPAR_DDT</code> (for example, <code>UserEM_IOPAR_DDT</code>), to avoid the DDT type conflict if you are using Equipment Module Pattern template to create two different templates with different number of input and output parameters.</p> <p>Also if you are importing <code>EMLogic</code> DFB in templatizer, then Verify that the DDT type name is different than <code>EM_IOPAR_DDT</code>.</p> <p>Below example shows how parameters can be added and removed from the existing template. For the data type information of Input/Output parameters (see <i>EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual</i>).</p> <p>a In templatizer, go to <code>UserEM_IOPAR_DDT</code>, add a new input parameter <code>IP07</code> and remove the output parameter <code>OP04</code> from DDT.</p>

Step	Action																																																			
	<div><div><div><div>Variables</div><div>DDT Types</div><div>Function Blocks</div><div>DFB Types</div></div><div><div>Filter</div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div><div>Name</div><div>=</div><div>x</div></div></div><div><table><thead><tr><th>Name</th><th>Type</th><th>Comment</th></tr></thead><tbody><tr><td>EMCTL_CFG_DDT</td><td><Struct></td><td>Provides the necessary data for t</td></tr><tr><td>EMCTL_SC_DDT</td><td><Struct></td><td>Provides the frequently needed d</td></tr><tr><td>EMCTL_ST_DDT</td><td><Struct></td><td>Provides the necessary data for r</td></tr><tr><td>UserEM_IOPAR_DDT</td><td><Struct></td><td>Equipment module parameters</td></tr><tr><td>IP01</td><td>BOOL</td><td>Input parameter 01</td></tr><tr><td>IP02</td><td>TIME</td><td>Input parameter 02</td></tr><tr><td>IP03</td><td>DINT</td><td>Input parameter 03</td></tr><tr><td>IP04</td><td>INT</td><td>Input parameter 04</td></tr><tr><td>IP05</td><td>REAL</td><td>Input parameter 05</td></tr><tr><td>IP06</td><td>INT</td><td>Input parameter 06</td></tr><tr><td>IP07</td><td>INT</td><td>Input parameter 07</td></tr><tr><td>OP01</td><td>REAL</td><td>Output parameter 01</td></tr><tr><td>OP02</td><td>TIME</td><td>Output parameter 02</td></tr><tr><td>OP03</td><td>BOOL</td><td>Output parameter 03</td></tr></tbody></table></div></div></div> <div>b</div> <div>After modifying the EMLogic DFB, input and output parameters in templatizer, analyze, build changes and save the templatizer.</div> <tr><td>c</td><td><div><div>Now in EMLogic_UL, add the new concatenation for input parameter IP07 and remove output parameter OP04 in MConcat. Link the newly added input parameter IP07 to EMParStatus as shown below.</div><div></div><div>Verify the concatenated name IP07 (with . as prefix) has to be same as defined in the EM_IOPAR_DDT type as mentioned in step 6 a.</div><div>NOTE: To link a process template with Equipment module template, you have to export the data structure (DDT) of respective process template (for example, DEVCTL_SC_DDT) from the respective process template UL and import that DDT in templatizer and create a new pin and assign the respective DDT structure in data type.</div><div>In UL, you have to instantiate the interface of corresponding process object (for example, \$DEVCTL_IF) and link it to the respective DFB pin. Now you need to defer the interface to application level. Refer PumpSetPattern template for additional information.</div></div><div>d</div><div>After modifying the EMLogic_UL, Save As the template with a different name (for example, UserEMLogic_UL).</div><tr><td>e</td><td>Open EquipmentModulePattern_UC, to replace the EMLogic_UL with newly created UL, drag and drop the newly created UL (UserEMLogic_UL) from Browser over the existing EMLogic_UL. Verify all the links with the existing EMLogic_UL are intact after replacing with new UL.</td></tr><tr><td>f</td><td>Now Save As the EquipmentModulePattern_UC with a different name (for example, UserEquipmentModulePattern_UC).</td></tr></td></tr>	Name	Type	Comment	EMCTL_CFG_DDT	<Struct>	Provides the necessary data for t	EMCTL_SC_DDT	<Struct>	Provides the frequently needed d	EMCTL_ST_DDT	<Struct>	Provides the necessary data for r	UserEM_IOPAR_DDT	<Struct>	Equipment module parameters	IP01	BOOL	Input parameter 01	IP02	TIME	Input parameter 02	IP03	DINT	Input parameter 03	IP04	INT	Input parameter 04	IP05	REAL	Input parameter 05	IP06	INT	Input parameter 06	IP07	INT	Input parameter 07	OP01	REAL	Output parameter 01	OP02	TIME	Output parameter 02	OP03	BOOL	Output parameter 03	c	<div><div>Now in EMLogic_UL, add the new concatenation for input parameter IP07 and remove output parameter OP04 in MConcat. Link the newly added input parameter IP07 to EMParStatus as shown below.</div><div></div><div>Verify the concatenated name IP07 (with . as prefix) has to be same as defined in the EM_IOPAR_DDT type as mentioned in step 6 a.</div><div>NOTE: To link a process template with Equipment module template, you have to export the data structure (DDT) of respective process template (for example, DEVCTL_SC_DDT) from the respective process template UL and import that DDT in templatizer and create a new pin and assign the respective DDT structure in data type.</div><div>In UL, you have to instantiate the interface of corresponding process object (for example, \$DEVCTL_IF) and link it to the respective DFB pin. Now you need to defer the interface to application level. Refer PumpSetPattern template for additional information.</div></div> <div>d</div> <div>After modifying the EMLogic_UL, Save As the template with a different name (for example, UserEMLogic_UL).</div> <tr><td>e</td><td>Open EquipmentModulePattern_UC, to replace the EMLogic_UL with newly created UL, drag and drop the newly created UL (UserEMLogic_UL) from Browser over the existing EMLogic_UL. Verify all the links with the existing EMLogic_UL are intact after replacing with new UL.</td></tr> <tr><td>f</td><td>Now Save As the EquipmentModulePattern_UC with a different name (for example, UserEquipmentModulePattern_UC).</td></tr>	e	Open EquipmentModulePattern_UC, to replace the EMLogic_UL with newly created UL, drag and drop the newly created UL (UserEMLogic_UL) from Browser over the existing EMLogic_UL. Verify all the links with the existing EMLogic_UL are intact after replacing with new UL.	f	Now Save As the EquipmentModulePattern_UC with a different name (for example, UserEquipmentModulePattern_UC).
Name	Type	Comment																																																		
EMCTL_CFG_DDT	<Struct>	Provides the necessary data for t																																																		
EMCTL_SC_DDT	<Struct>	Provides the frequently needed d																																																		
EMCTL_ST_DDT	<Struct>	Provides the necessary data for r																																																		
UserEM_IOPAR_DDT	<Struct>	Equipment module parameters																																																		
IP01	BOOL	Input parameter 01																																																		
IP02	TIME	Input parameter 02																																																		
IP03	DINT	Input parameter 03																																																		
IP04	INT	Input parameter 04																																																		
IP05	REAL	Input parameter 05																																																		
IP06	INT	Input parameter 06																																																		
IP07	INT	Input parameter 07																																																		
OP01	REAL	Output parameter 01																																																		
OP02	TIME	Output parameter 02																																																		
OP03	BOOL	Output parameter 03																																																		
c	<div><div>Now in EMLogic_UL, add the new concatenation for input parameter IP07 and remove output parameter OP04 in MConcat. Link the newly added input parameter IP07 to EMParStatus as shown below.</div><div></div><div>Verify the concatenated name IP07 (with . as prefix) has to be same as defined in the EM_IOPAR_DDT type as mentioned in step 6 a.</div><div>NOTE: To link a process template with Equipment module template, you have to export the data structure (DDT) of respective process template (for example, DEVCTL_SC_DDT) from the respective process template UL and import that DDT in templatizer and create a new pin and assign the respective DDT structure in data type.</div><div>In UL, you have to instantiate the interface of corresponding process object (for example, \$DEVCTL_IF) and link it to the respective DFB pin. Now you need to defer the interface to application level. Refer PumpSetPattern template for additional information.</div></div> <div>d</div> <div>After modifying the EMLogic_UL, Save As the template with a different name (for example, UserEMLogic_UL).</div> <tr><td>e</td><td>Open EquipmentModulePattern_UC, to replace the EMLogic_UL with newly created UL, drag and drop the newly created UL (UserEMLogic_UL) from Browser over the existing EMLogic_UL. Verify all the links with the existing EMLogic_UL are intact after replacing with new UL.</td></tr> <tr><td>f</td><td>Now Save As the EquipmentModulePattern_UC with a different name (for example, UserEquipmentModulePattern_UC).</td></tr>	e	Open EquipmentModulePattern_UC, to replace the EMLogic_UL with newly created UL, drag and drop the newly created UL (UserEMLogic_UL) from Browser over the existing EMLogic_UL. Verify all the links with the existing EMLogic_UL are intact after replacing with new UL.	f	Now Save As the EquipmentModulePattern_UC with a different name (for example, UserEquipmentModulePattern_UC).																																															
e	Open EquipmentModulePattern_UC, to replace the EMLogic_UL with newly created UL, drag and drop the newly created UL (UserEMLogic_UL) from Browser over the existing EMLogic_UL. Verify all the links with the existing EMLogic_UL are intact after replacing with new UL.																																																			
f	Now Save As the EquipmentModulePattern_UC with a different name (for example, UserEquipmentModulePattern_UC).																																																			

Step	Action
g	<p>Navigate and open EMPAR_CD template, UserEMPPattern (newly created template from above point 2) → EquipmentModulePattern_CS → EquipmentModulePattern_CD → EMPAR_CD.</p> <p>Add the new variable tag with name IP07, add concatenation for IP07, link name and address, also provide the description and other parameters like range and units for the variable and remove the OP04 variable tag with all its references as shown in below figure. For the data type information of Input/Output parameters (see EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual).</p> 
h	<p>To configure the input parameters as the Enumeration data type, you have to configure the EngUnit and Custom fields accordingly as shown.</p> <p>Format is, <item name>:<item value> (for example, @ (Item1) :1; @ (Item4) :9).</p> <p>EngUnit - <E></p> <p>Below figure shows the example of Input parameter IP04 is configured for the Enumeration data type.</p>  <p>For additional information about configuring the Enumeration data type (see EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual).</p>
i	<p>After modification, Save As the EMPAR_CD with a different name (for example, UserEMP_CD) and to replace the EMPAR_CD with newly created CD, drag and drop the newly created CD (UserEMP_CD) from Browser over the existing EMPAR_CD in EquipmentModulePattern_CD.</p>
7	<p>In EquipmentModulePattern_CD template, navigate to the EMCTL_CD and locate the Custom field parameters of Strategy and EMState, configure the Strategy and EMState by creating an input value for the element as shown.</p> <p>Format is, <Strategy text>:<list of applicable input parameters separated by a comma> (for example, @ (S1) :1,2,3,4,5,6; @ (S9) :2,3,4).</p> <p>The Equipment Module states 0 to 15 are predefined and fixed. Equipment Module states 16 to 31 states are user defined states. To configure Equipment Module states you have to configure the EMState variable as <State Description>;<State Description> (for example, @ (EMState16) ; @ (EMState24)).</p>

Step	Action
	<p>For additional information about configuring the Strategy and EMState management (see EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual).</p> 
8	<p>You can Disable, Historize and configure the name of strategy execution state alarms in the Application Explorer. Privilege, Help and Category parameters are common for all the execution states and Equipment Module states.</p> <p>User can Disable, Historize and configure the name of Equipment Module state alarms from the EMCTL_CD by configuring its input values as per requirement.</p> 
9	<p>Parameters like Delay, Custom1...Custom8, Paging and PagingG) of Equipment Module states, Strategy execution states alarms has to be configured inside the EMCTL_CD.</p> <p>Below figure shows the parameters for Strategy execution state alarms.</p>

Step	Action
	<div></div> <p>Below figure shows the parameters for predefined EMstates alarms.</p> <div></div> <p>Below figure shows the parameters for user defined Equipment Module state alarms.</p>

Step	Action
	
10	<p>While configuring the user defined Equipment Module states you can rename the parameters and EMstate alarm variable names as per requirement. You can modify the tag name and description of the user defined Equipment Module states.</p>  <p>NOTE: You have to Verify that the predefined Equipment Module state alarm variables, tag names and descriptions are not to be modified. You can Defer required configuration parameters to Equipment Module level so that it can be available for modification in Application Explorer. If certain alarm tags are not used, then those tags can be deleted in EMCTL_CD.</p>
11	<p>After modification, Save As the EMCTL_CD template with a different name (for example, UserEMCTL_CD) and to replace the EMCTL_CD with newly created CD, drag and drop the newly created CD (UserEMCTL_CD) from Browser over the existing EMCTL_CD in EquipmentModulePattern_CD.</p>
12	<p>Now Save As the EquipmentModulePattern_CD with a different name (for example, UserEquipmentModulePattern_CD) and to replace the EquipmentModulePattern_CD with newly created CD, drag and drop the newly created CD (UserEquipmentModulePattern_CD) from Browser over the existing EquipmentModulePattern_CD in EquipmentModulePattern_CS.</p>

Step	Action
13	<p>To configure genie parameters DATA1 to DATA4, you have to go to EquipmentModulePattern_CS and concatenate the input/output parameter tag as per requirement as shown below.</p> <p>The figure shows the output parameter1 to output parameter4 that are configured for DATA1 to DATA4.</p>
14	<p>Now save as the EquipmentModulePattern_CS with different name (for example, UserEquipmentModulePattern_CS). Now open and to replace the EquipmentModulePattern_CS with newly created CS, drag and drop the newly created CS (UserEquipmentModulePattern_CS) from Browser over the existing EquipmentModulePattern_CS in EMPattern .</p>
15	<p>Open UserEMPPattern template (created from step 2) to replace UC and CS with the newly created UC and CS.</p> <p>Now, to replace the EquipmentModulePattern_CS with newly created CS, drag and drop the newly created CS (UserEquipmentModulePattern_CS) from Browser over the existing EquipmentModulePattern_CS.</p> <p>To replace the EquipmentModulePattern_UC with newly created UC (UserEquipmentModulePattern_UC created from step 6 f), drag and drop the newly created UC (UserEquipmentModulePattern_UC) from Browser over the existing EquipmentModulePattern_UC</p>
16	<p>Close all the open templates (if any). Go to Global Templates and navigate to the newly created main template UserEMPPattern. Right click and Update.</p>
17	<p>Verify that this newly created template is updated with the all user modified composite templates. Now, the template is ready to use.</p>

Pump Set Pattern Management

PumpSetPattern - Pump Set Pattern Equipment Module

General Description

The `PumpSetPattern` template is used to monitor and manage the Pump Set operation. It is provided with the configuration of 5 pumping assets. Each asset includes one Pump and three Valves (Inlet, Outlet & Drain).

The composite and facet names without the \$ prefix can be modified, while the composite and facet names with \$ prefix are standard and cannot be modified.

These templates do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

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

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

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

Parameters

Configuration

The table describes the **Configuration** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Logic	AutoResetDis	Boolean	False	If <code>True</code> , auto reset is disabled.
PumpSet-Pattern	EnabledStrategies	Enum	Priority and Balance	Determines strategies to be enabled. 1 - Priority only 2 - Balance only Other value – Both

Element Name	Name	Type	Default value	Description
	OutValveFirst	Boolean	False	Outlet Valve is opened before starting the pump and closed after stopping the pump.
Genies	EMCTL_x	Boolean	False	1 = Hides the corresponding button on the genie and faceplate. x represents the name of the command. For example, EMCTL_STOP .
NOTE: Values of Boolean parameters are set by using check boxes: <ul style="list-style-type: none"> Selected = True Cleared = False 				

Time

The table describes the **Time** parameters that you can configure:

Element Name	Name	Type	Default value	Description
PumpSet-Pattern	SwitchOnDelay	Duration	00:00:03	Delay between the end of the starting sequence of a pumping asset and start of the starting sequence of the next pumping asset.
	DrainingTime	Duration	00:00:05	Time for which drain valve will be open when stopping the pumping asset.
	DelayAfterValveOpen	Duration	00:00:03	Delay time before next action after opening valves.
	DelayAfterValveClose	Duration	00:00:03	Delay time before next action after closing valves.
	DelayAfterPumpStart	Duration	00:00:03	Delay time before next action after starting pumps.
	DelayAfterPumpStop	Duration	00:00:03	Delay time before next action after stopping pumps.

Basic Alarm Configuration

The table describes the **Basic Alarm Configuration** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Name_x	String	-	Name_x: Name of the physical device associated with the alarm. Corresponds to the <i>Alarm Name</i> property of advanced alarms of the Supervision Participant. For example, Name_IDLE is for ControlTags and Name_EM_ALARM is for ParameterTags Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
ParameterTags				
ControlTags	Category	String	SGC_CAT_EVENT	Category of alarm/event for all states. Alarm category label or number. Corresponds to the category property of advanced alarms of the Supervision Participant.
ParameterTags	CAT_x	String	Label with the syntax: <i>SGC_CAT_<type of component>_<type of alarm></i>	CAT_x: Alarm category label or number. Corresponds to the <i>Category</i> property of advanced alarms of the Supervision Participant. For example, for the parameter Category_EM_ALARM the label is SGC_CAT_PS_ALARM . Refer to Alarm Category Labels and Numbers (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Supervision Services Reference Manual).
ControlTags	Help	String	-	Help of alarm/event for all states. Name of the graphics page that opens when the help function is called. Corresponds to the <i>Help</i> property of advanced alarms of the Supervision Participant.
ParameterTags	Help_x	String	-	Help_x: Name of the graphics page that opens when the help function is called. Corresponds to the <i>Help</i> property of advanced alarms of the Supervision Participant. For example, Help_EM

Element Name	Name	Type	Default value	Description
ParameterTags	Delay_X	Duration	00:00:00	Delay_x : Alarm delay period. Corresponds to the <i>Delay</i> property of advanced alarms of the Supervision Participant. For example, Delay_EM_Fail .
<div> <div>X</div> <div> <p>Suffix of the variable tag name or advanced alarm tag that is managed by the corresponding Supervision components. For example, Name_EM_ALARM correspond to the <i>EM Name_PumpSet_EM_ALARM</i> advanced alarm tag managed by the <i>PUMPSETPATTERN</i> Control resource.</p> <p>Refer to the description of tags of the Supervision component (see EcoStruxure™ Process Expert LVPC Library, Supervision Services User Guide).</p> </div> </div>				

Advanced Alarm Configuration

The table describes the **Advanced Alarm Configuration** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Privilege	String	Label with the syntax: <i>SGC_PRIV_<type of component>_<type of action></i>	Privilege level required to acknowledge or disable the alarm. Corresponds to the <i>Privilege</i> property of advanced alarms of the Supervision Participant. For example, <i>SGC_PRIV_EVENT</i> . NOTE: For advanced alarms that are process events in the label <i><type of component></i> and <i><type of action></i> are replaced by EVENT , for example SGC_PRIV_EVENT .
ParameterTags	Privilege_x	String	Label with the syntax: <i>SGC_PRIV_<type of component>_<type of action></i>	Privilege_x : Privilege level required to acknowledge or disable the alarm. Corresponds to the <i>Privilege</i> property of advanced alarms of the Supervision Participant. For example, for Privilege_EM_ALARM , the label is SGC_PRIV_PS_ALARM . Refer to Privilege Levels and Labels (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Supervision Services Reference Manual).
	Custom1_x to Custom8_x	String	-	Custom1_x to Custom8_x : String to filter active alarms (64 characters maximum). Corresponds to the <i>Custom 1 to Custom 8</i> property of advanced alarms of the Supervision Participant. For example, Custom1_EM_ALARM Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
	Paging_x	Boolean	False	Paging_x : Read/write property, which indicates the paging status of the alarm. Corresponds to the <i>Paging</i> property of advanced alarms of the Supervision Participant. For example, Paging_EM_ALARM . 1 = The alarm is paged.
	PagingG_x	String	-	PagingG_x : A read-only text string that indicates the paging group to which the alarm belongs (80 characters maximum). Corresponds to the <i>Paging Group</i> property of advanced alarms of the Supervision Participant. For example, PagingG_EM_ALARM .
<div> <div>X</div> <div> <p>Suffix of the variable tag name or advanced alarm tag that is managed by the corresponding Supervision components. For example, Paging_EM_ALARM correspond to the <i>EM Name_PumpSet_EM_ALARM</i> advanced alarm tag managed by the <i>PUMPSETPATTERN</i> Control resource.</p> <p>Refer to the description of tags of the Supervision component (see EcoStruxure™ Process Expert LVPC Library, Supervision Services User Guide).</p> </div> </div>				

Historize

The table describes the **Historize** parameters that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Historize_x	Boolean	False	Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . x represents the child elements of the <i>EMCTL_ST</i> and <i>EMCTL_CFG</i> data structures, and the <i>EMCTL.CSTEPD</i> variable. For example, Historize_STW .
	Historize_y			Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . y represents the Strategy execution state alarms. For example, Historize_IDLE . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
Parameter-Tags	Historize_EM_ALARM	Boolean	False	Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
	Historize_EM_FAIL			
	Historize_x_ALARM			Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> .
	Historize_x_FAIL			x represents the Pumping Asset 01 to 05 (ASSET01 to ASSET05). For example, Historize_ASSET01_ALARM and Historize_ASSET01_FAIL . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
InitialConditionTags and FailureConditionTags	Historize_BYPASSW	Boolean	False	Enable historize bypass word.
	Historize_CONDW	Boolean	False	Enable historize condition word.
	Historize_REARMREQW	Boolean	False	Enable historize rearm word.

Alarm Tag Disable

The table describes the **Alarm Tag Disable** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Disabled_x	Boolean	True	Corresponds to the <i>Alarm Tag Disable</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Alarm Tag Disable</i> property of the corresponding tag to False. x represents the name of the state. For example, Disabled_RUNNING .
ParameterTags	Disabled_EM_ALARM	Boolean	True	Corresponds to the <i>Alarm Tag Disable</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Alarm Tag Disable</i> property of the corresponding tag to False. Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
	Disabled_EM_FAIL			
	Disabled_x_ALARM			Corresponds to the <i>Alarm Tag Disable</i> property of advanced alarm tags of the Supervision Participant.
	Disabled_x_FAIL			1 = Sets the <i>Alarm Tag Disable</i> property of the corresponding tag to False.

Element Name	Name	Type	Default value	Description
				<p>x represents the Pumping Asset 01 to 05 (ASSET01 to ASSET05). For example, Disabled_ASSET01_ALARM and Disabled_ASSET01_FAIL.</p> <p>Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.</p>
<p>NOTE:</p> <p>Values of Boolean parameters are set by using check boxes:</p> <ul style="list-style-type: none"> Selected = True Cleared = False 				

Variable Tag Disable

The table describes the **Variable Tag Disable** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
InitialCondition- Tags	BYPASSWordSe- lection	Boolean	False	Disable initial conditions bypass tags.
	REARMREQ- WordSelection	Boolean	False	Disable initial condition resetting (REARM) tags.
FailureCondi- tionTags	BYPASSWordSe- lection	Boolean	False	Disable conditions bypass.
	REARMREQ- WordSelection	Boolean	False	Disable conditions manual resetting.
<p>NOTE:</p> <p>Values of Boolean parameters are set by using check boxes:</p> <ul style="list-style-type: none"> Selected = True Cleared = False 				

The following optional elements feature configurable parameters:

- **InitialConditionTags:** Messages associated to initial conditions, to appear on the Initial Conditions tab of the faceplate for pump set pattern.
- **FailureConditionTags:** Messages associated to diagnostic, to appear on the Diagnostic tab of the faceplate for pump set pattern.

Composition

The `PumpSetPattern` template references composite and facet templates, which provide the following services:

Control	Core services, plus additional optional services, which you can activate if needed.
Supervision	These services complement the Control services. Supervision services are optional.

The table describes the services that are available from the `PumpSetPattern` template and the corresponding facet, which implements the service:

Control services (PumpSetPattern_UC)	Corresponding facet template	Control service description	Associated Supervision services (PumpSetPattern_CS)*	Corresponding facet template	Supervision service description
Core services					
Logic	\$EMCTL_UL	Refer to the EMCTL DFB*.	ControlTags	PumpSet-Pat-ternCTL_CD	Refer to tags of Equipment Module Management*.
-	-	-	Genies	\$EMPan-elState_CG	Refer to the genie representation of the Equipment Module Management**.
				\$EMPan-elWith-Buttons_CG	
				\$EMPane-lAllDa-ta2_CG	
				\$EMPane-lAllDa-ta4_CG	
PumpSetPattern	PumpSetPattern_UL	Refer to PUMPSETPATTERN DFB***.	ParameterTags	PumpSet-Pattern-PAR_CD	Refer to tags of Pump Set Pattern Management**.
Optional Services					
InitialConditions	\$CONDSUM_UL	Refer to the CONDSUM DFB***.	InitialCondition-Tags	\$INIT-CONDSUM_CD	-
FailureConditions	\$CONDSUM_UL	Refer to the CONDSUM DFB***.	FailureCondi-tionTags	\$CON-DSUM_CD	
* See EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual					
** See EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual					
*** See EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual					

NOTE: When you activate an optional Control service, the corresponding Supervision service is automatically activated, if available.

The control module template exposes the following interface.

Interface Identifier	Type/Role	Description
EMCTLName	\$EMCTLName/Def	Equipment module control name.
Pump01	\$DEVCTL_IF/Ref	Pumping asset 0x pump.
InletValve0x	\$DEVCTL_IF/Ref	Pumping asset 0x inlet valve.
OutletValve0x	\$DEVCTL_IF/Ref	Pumping asset 0x outlet valve.
DrainValve0x	\$DEVCTL_IF/Ref	Pumping asset 0x drain valve.
NOTE: x indicates values from 1 to 5.		

Below table shows the list of templates which are available for modification.

Name of Facets, Composites and Templates	Description	Comment
PumpSetPattern_UL	To create pump set user logic and configure input and output parameters for control component as per requirement.	To modify this facet template you have to Save As and rename it as per requirement.
PumpSetPattern_UC	If PumpSetPattern_UL is modified then PumpSetPattern_UC has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.

Name of Facets, Composites and Templates	Description	Comment
PumpSetPatternPAR_CD	To configure the input and output parameters pump set for supervision component as per requirement.	To modify this composite template you have to Save As and rename it as per requirement.
PumpSetPatternCTL_CD	To configure the Custom1...8, Paging, PagingG parameters of alarms of Pump Set equipment module states and strategy execution states.	To modify this composite template you have to Save As and rename it as per requirement.
PumpSetPattern_CD	To configure the Strategy and EMstate variables. Also, to configure Disable, Historize and Name parameters of alarms of Equipment Module states and strategy execution states. If PumpSetPatternPAR_CD or PumpSetPatternCTL_CD is modified then PumpSerPattern_CD has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
PumpSetPattern_CS	To configure the DATA1 to DATA4 parameters. The parameters DATA1 to DATA4 can be any input/output parameters which can be viewed in genie for monitoring purpose. If PumpSetPattern_CD is modified then PumpSetPattern_CS has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
PumpSetPattern	Main Pump Set Pattern template. If PumpSetPattern_UC or PumpSerPattern_CS is modified then PumpSetPattern has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.

Default Configuration available for PumpSetPattern Template

The template is provided with strategies, user defined Equipment Module States, input and output parameters, and four configurable data parameters. The template can be modified as per its application requirements.

The Input parameters and their data types are as follows:

Input Parameter	Data Type
Required Number of Active Pumps	Numeric (Signed integer)
Reset Pumps Operation Time	Boolean
Priority Pump 1	Numeric (Signed integer)
Priority Pump 2	Numeric (Signed integer)
Priority Pump 3	Numeric (Signed integer)
Priority Pump 4	Numeric (Signed integer)
Priority Pump 5	Numeric (Signed integer)

The output parameters and their data types are as follows:

Output Parameter	Data Type (user)
Equipment Module Diagnosis State	Enumeration
Requested Pumps	Numeric (Signed integer)
Active Pumps	Numeric (Signed integer)
Pump 01 Operation Time	Duration
Pumping Asset 01 State	Enumeration
Pump 02 Operation Time	Duration
Pumping Asset 02 State	Enumeration
Pump 03 Operation Time	Duration

Output Parameter	Data Type (user)
Pumping Asset 03 State	Enumeration
Pump 04 Operation Time	Duration
Pumping Asset 04 State	Enumeration
Pump 05 Operation Time	Duration
Pumping Asset 05 State	Enumeration

NOTE: For the Control and Supervision data types of input and output parameters, page 26.

For the Output parameters with the Enumeration data type, their variable tags are configured as:

- Custom1 - @(Normal):0
- Custom2 - @(Warning):1
- Custom3 - @(Failure):2

The default value configured for DATA 1 to DATA 4 genie parameters are Output parameter 1 to Output Parameter 4 respectively.

The applicable parameters for default Strategy are:

Strategy	Applicable Input Parameters
Program SP/Balance	Reset pumps operation time
Operator SP/Balance	Reset pumps operation time and Required number of active pumps set point
Program SP/Priority	Reset pumps operation time, Priority pump 01, Priority pump 02, Priority pump 03, Priority pump 04, Priority pump 05.
Operator SP/Priority	Required number of active pumps set point, Reset pumps operation time, Priority Pump 01, Priority Pump 02, Priority Pump 03, Priority Pump 04, Priority Pump 05.
Recalculate	Reset pumps operation time
Stop Equipment	Reset pumps operation time

The Equipment Module states are:

Equipment Module State	Description
Predefined Equipment Module states	
EMState0	Off
EMState1	Stopped
EMState10	Stopping
User defined Equipment Module states	
EMState16	Running Program SP/Balance
EMState17	Running Operator SP/Balance
EMState18	Running Program SP/Priority
EMState19	Running Operator SP/Priority
EMState20	Configuring
EMState21	Running Externally

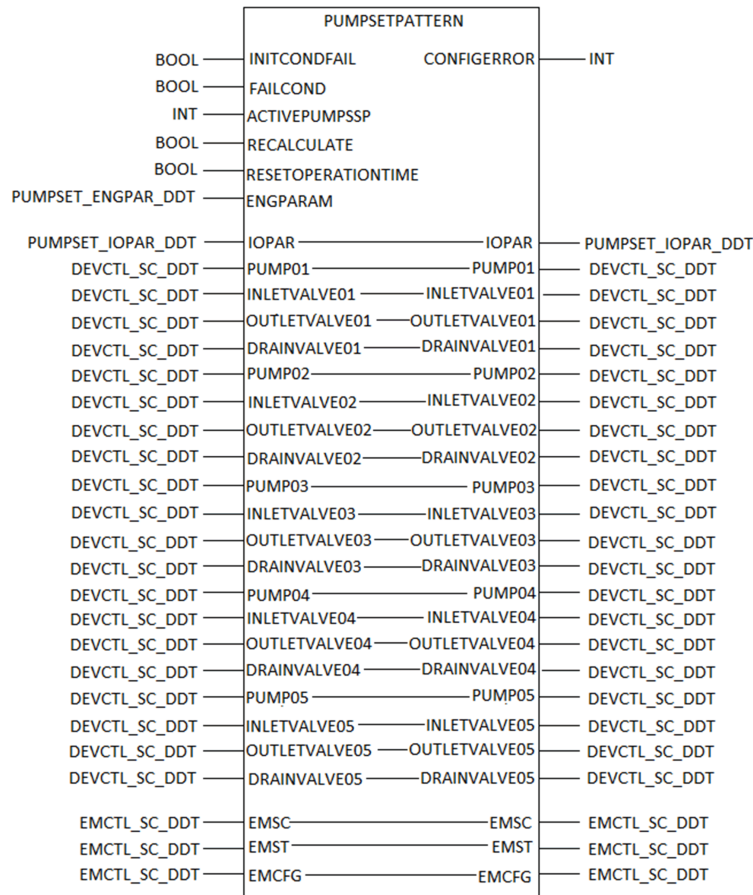
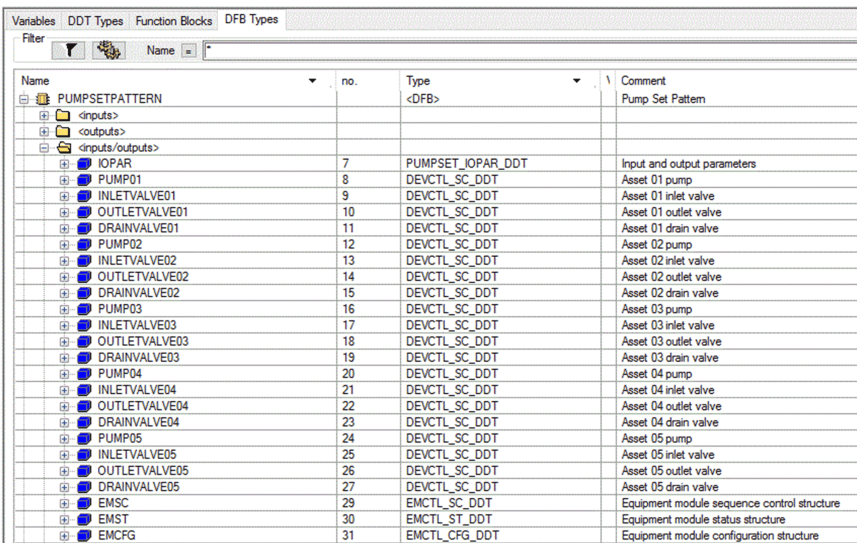
Modifying the PumpSetPattern Template

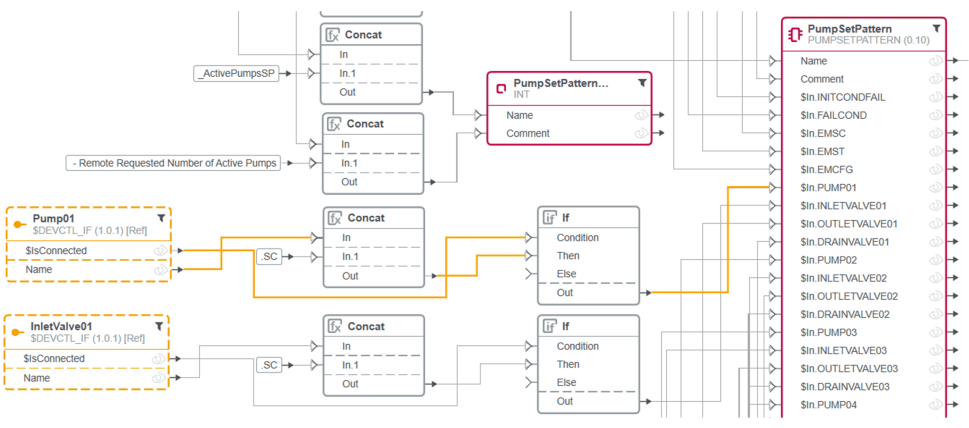
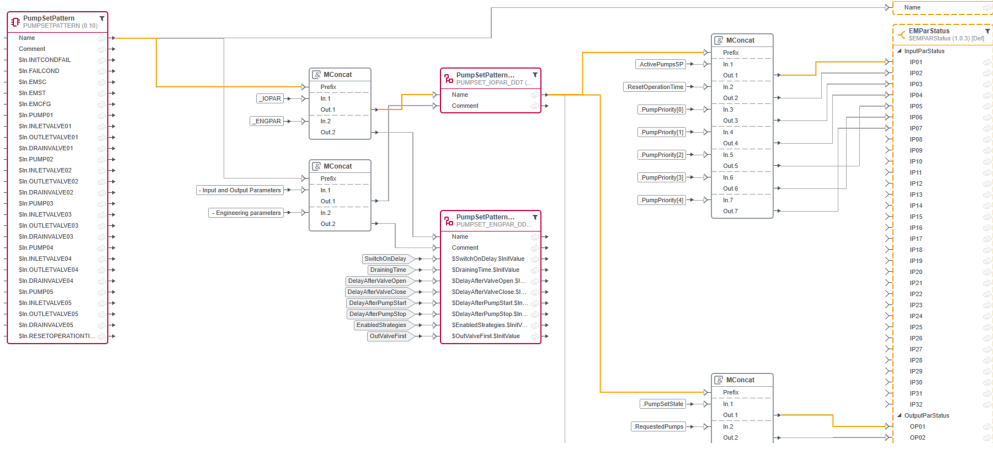
Steps to modify PumpSetPattern template for the Pump Set Operation

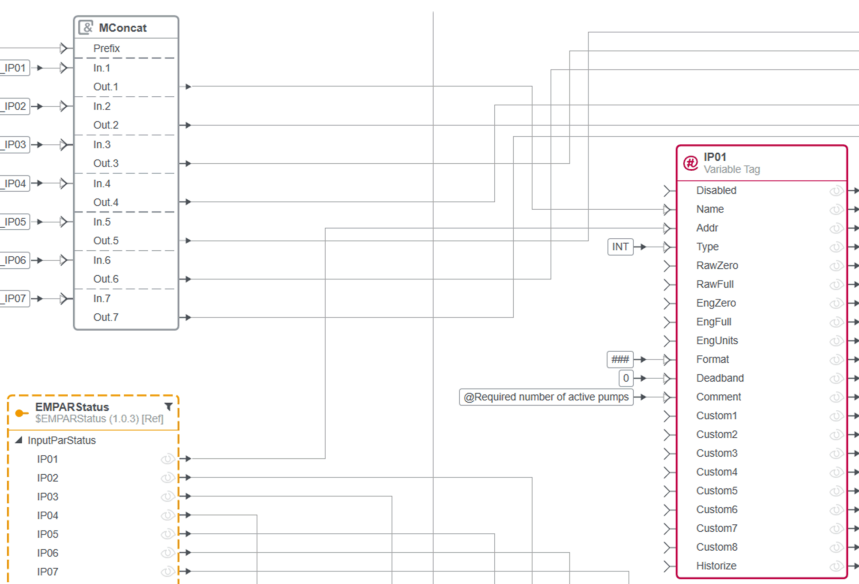
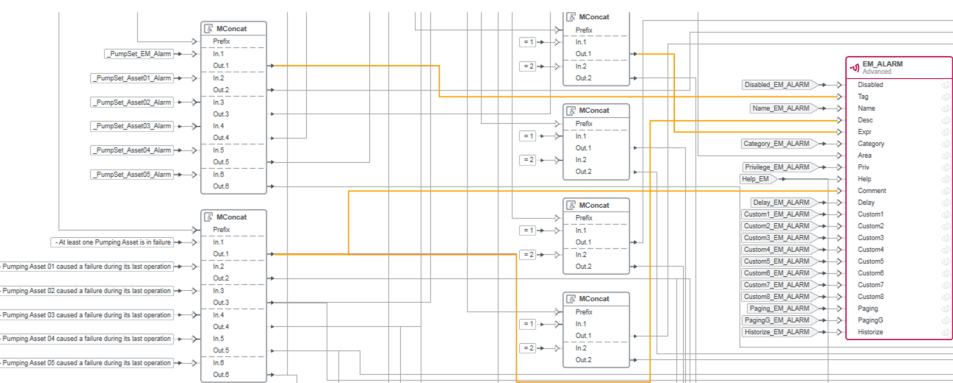
The PumpSetPattern template is provided with 5 pumping assets. Each asset includes one pump and three valves (Inlet, Outlet and Drain valve).

Step	Action																																																												
1	<p>Navigate to the PumpSetPattern template from Global Templates, this template is available under Global Templates → General Purpose Library → Equipment Module Patterns → Equipment Modules.</p> <p>Save As the facet and composite templates as per application specific requirement, page 39. Also replace the existing facet and composite templates with the newly created facet and composite templates (Refer to Modifying the EMPattern Template, page 27 to follow template replacement procedure).</p>																																																												
2	<p>Open PumpSetPattern_UL facet template from PumpSetPattern template, in templatizer. The EMLogic DFB is modified with the Pump Set logic and its DFB type is changed to PUMPSETPATTERN.</p> <p>The modified DFB type name and sections are as shown below:</p> <div><div>VariablesDDT TypesFunction BlocksDFB Types</div><div>Filter<input type="text"/></div><table><thead><tr><th>Name</th><th>no.</th><th>Type</th><th>Comment</th></tr></thead><tbody><tr><td> PUMPSETPATTERN</td><td></td><td><DFB></td><td>Pump Set Pattern</td></tr><tr><td> <inputs></td><td></td><td></td><td></td></tr><tr><td> <outputs></td><td></td><td></td><td></td></tr><tr><td> <inputs/outputs></td><td></td><td></td><td></td></tr><tr><td> <public></td><td></td><td></td><td></td></tr><tr><td> <private></td><td></td><td></td><td></td></tr><tr><td> <sections></td><td></td><td></td><td></td></tr><tr><td> INIT</td><td></td><td><ST></td><td>Read inputs, calculate configuration error and asset state as well reset the operation time counters</td></tr><tr><td> RUNNING</td><td></td><td><ST></td><td>Execute the strategy</td></tr><tr><td> HOLDING</td><td></td><td><ST></td><td>Execute the Holding sequence for different strategy</td></tr><tr><td> RESTARTING</td><td></td><td><ST></td><td>Execute the Restarting sequence for different strategy</td></tr><tr><td> ABORTING</td><td></td><td><ST></td><td>Execute the Aborting sequence for different strategy</td></tr><tr><td> STOPPING</td><td></td><td><ST></td><td>Execute the Stopping sequence for different strategy</td></tr><tr><td> EMSTATE</td><td></td><td><ST></td><td>Execute the EM state code and start and stop pumps in configuring case of the EM state</td></tr></tbody></table></div>	Name	no.	Type	Comment	PUMPSETPATTERN		<DFB>	Pump Set Pattern	<inputs>				<outputs>				<inputs/outputs>				<public>				<private>				<sections>				INIT		<ST>	Read inputs, calculate configuration error and asset state as well reset the operation time counters	RUNNING		<ST>	Execute the strategy	HOLDING		<ST>	Execute the Holding sequence for different strategy	RESTARTING		<ST>	Execute the Restarting sequence for different strategy	ABORTING		<ST>	Execute the Aborting sequence for different strategy	STOPPING		<ST>	Execute the Stopping sequence for different strategy	EMSTATE		<ST>	Execute the EM state code and start and stop pumps in configuring case of the EM state
Name	no.	Type	Comment																																																										
PUMPSETPATTERN		<DFB>	Pump Set Pattern																																																										
<inputs>																																																													
<outputs>																																																													
<inputs/outputs>																																																													
<public>																																																													
<private>																																																													
<sections>																																																													
INIT		<ST>	Read inputs, calculate configuration error and asset state as well reset the operation time counters																																																										
RUNNING		<ST>	Execute the strategy																																																										
HOLDING		<ST>	Execute the Holding sequence for different strategy																																																										
RESTARTING		<ST>	Execute the Restarting sequence for different strategy																																																										
ABORTING		<ST>	Execute the Aborting sequence for different strategy																																																										
STOPPING		<ST>	Execute the Stopping sequence for different strategy																																																										
EMSTATE		<ST>	Execute the EM state code and start and stop pumps in configuring case of the EM state																																																										

For more information on pump set operation logic, DFB pins and DDT, (see EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual).

Step	Action
	<p>The DFB Layout is as shown below:</p>  <p>The diagram shows the PUMPSETPATTERN DFB layout. It includes several input pins on the left (BOOL, INT, PUMPSET_ENGPARG_DDT, PUMPSET_IOPAR_DDT, and multiple DEVCTL_SC_DDT pins) and output pins on the right (INT, PUMPSET_IOPAR_DDT, and multiple DEVCTL_SC_DDT pins). Internal connections are shown between these pins and various internal blocks like INITCONDFAIL, CONFIGERROR, ACTIVEPUMPSSP, RECALCULATE, RESETOPERATIONTIME, ENGPARAM, and multiple pump/valve control blocks (PUMP01-05, INLETVALVE01-05, OUTLETVALVE01-05, DRAINVALVE01-05). At the bottom, there are EMCTL_SC_DDT pins connected to EMSC, EMST, and EMCFCG blocks.</p>
3	<p>To link the motor and valves to the PUMPSETPATTERN DFB, the DEVCTL_SC_DDT type is used as data type for Input and Output pins (PUMP01, INLETVALVE01, OUTLETVALVE01, DRAINVALVE01 to PUMP05, INLETVALVE05, OUTLETVALVE05, DRAINVALVE05).</p>  <p>The screenshot shows the 'Variables' tab in the software interface. It displays a table with columns: Name, no., Type, and Comment. The table lists various variables used in the PUMPSETPATTERN DFB, including IOPAR, PUMP01-05, INLETVALVE01-05, OUTLETVALVE01-05, DRAINVALVE01-05, EMSC, EMST, and EMCFCG. The 'Type' column shows the data types assigned to each variable, such as PUMPSET_IOPAR_DDT, DEVCTL_SC_DDT, EMCTL_SC_DDT, and EMCTL_CFG_DDT.</p>
4	<p>To link the motor and valve control modules, in UL with the Pump Set operation, the name interface of \$DEVCTL_IF template is instantiated as a REF type and is connected to the DFB as shown below.</p>

Step	Action
	 <p>The motor and valve interfaces are deferred to the application level of equipment module in order to link the motor and valve control modules to the pump set operation in Edit Links of PumpSetPattern instance.</p> <p>If user wishes to add or remove the pumping asset to the existing pump set, then accordingly has to modify the logic and add or remove asset input output pins from DFB. For details, (see EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual).</p> <p>In UL, user have to instantiate corresponding name interfaces \$DEVCTL_IF and link them to the DFB pins identical to existing interface links and expose it till the application level of equipment module.</p>
5	<p>The element names of PUMPSET_IOPAR_DDT are concatenated with the variable name InstanceID_IOPAR and are linked to the \$EMPARStatus interface as shown below.</p>  <p>To add the pumping asset, user should add concatenation for the respective input and output parameters in MConcat (for example PUMPPRIORITY [5], PUMPOPERATIONTIME [5], PUMPINGASSETSTATE [5]) and link it to \$EMPARStatus interface.</p> <p>The engineering parameters are created and are deferred to the application level of equipment module. For more details, page 35.</p>
6	<p>In PumpSetPatternPAR_CD, descriptions and data type of input output parameters are configured as per the requirement of pump set operation.</p>

Step	Action
	 <p>The alarm tags are created for the Alarm or detected failure conditions of Pump set and each asset. These alarms can be configured from the application explorer.</p>  <p>To add or remove pumping asset, user should add or remove the input output variable tags with reference to the tags created in the point 4. Also user can add and configure the alarm tags for the alarm and detected failure condition of the newly added pumping assets identical to the existing alarm tags and defer its all configuration parameters to the application level of equipment module.</p>
7	<p>The EMState and Strategies are configured as input values for PumpSetPatternCTL_CD in PumpSetPattern_CD as shown below. The user can configure more EMStates or Strategies as per the requirement, but accordingly should modify the logic in PUMPSETPATTERN DFB. For more details, (see EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual).</p>

Step	Action
	<p>The diagram illustrates the configuration of pump set patterns. It shows a mapping from specific pump set states (e.g., Running program SP/Balance, Running operator SP/Balance, Configuring, Running externally, Program SP/Balance, Operator SP/Balance, Program SP/Priority, Operator SP/Priority, Recalculate, Stop equipment) to custom EMSTATE and STRATEGY parameters (Custom1_EMSTATE through Custom8_STRATEGY). The states are grouped into two main categories: EMSTATE and STRATEGY. The EMSTATE parameters are mapped to the first six states, and the STRATEGY parameters are mapped to the remaining states. The diagram uses arrows to show the flow from the states to the parameters and then to the icons representing the states.</p>
8	<p>In PumpSetPatternCTL_CD, the basic alarm parameters (including Privilege parameter) for strategy execution state alarms are available to modify at the application explorer level.</p> <p>The EM state alarms (EM OFF, STOPPED, STOPPING, RUN PROGRAM SP/BALANCE, RUN OPERATOR SP/BALANCE, RUN PROGRAM SP/PRIORITY, RUN OPERATOR SP/PRIORITY, CONFIGURING and RUNNING EXTERNALLY) related to the pump set operation are available for configuration. The basic alarm parameters for these alarms can be configured in PumpSetPattern_CD as input values to the PumpSetPatternCTL_CD.</p> <p>To configure the advanced alarm parameters for EMState alarms and strategy execution state alarms, user has to go inside the PumpSetPatternCTL_CD. Other predefined EM state alarms which are not required for the pump set operation are removed.</p>
9	To configure genie parameters DATA1 to DATA4, user has to go to PumpSetPattern_CS and concatenate the input/output parameter tag as per requirement.
10	Update the PumpSetpattern template with the updated PumpSetPattern_CS and PumpSetPattern_UC composite templates. Now the template is ready to use.

Flow Control Pattern Management

FlowCtlPattern - Flow Control Pattern Equipment Module

General Description

The `FlowCtlPattern` template is used to monitor and manage the Flow Control operation. It is provided with the configuration of 5 pumping assets. Each asset includes one variable speed pump and three valves (Inlet, Outlet and Drain).

The composite and facet names without the \$ prefix can be modified, while the composite and facet names with \$ prefix are standard and cannot be modified.

These templates do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

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

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

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

Parameters

Configuration

The table describes the **Configuration** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Logic	AutoResetDis	Boolean	False	If <code>True</code> , auto reset is disabled.
Flow-CtlPattern	MinimumSP	Float	40.0	Minimum pump setpoint (valid range is 0 -100%).
	OptimalSP	Float	80.0	Optimal pump setpoint (valid range is 0 -100%).
	MaximumSP	Float	90.0	Maximum pump setpoint (valid range is 0 -100%).

Element Name	Name	Type	Default value	Description
	EnabledStrategies	Enum	Priority and Balance	Determines strategies to be enabled.
	OutValveFirst	Boolean	False	Open outlet valve before starting the pump and closed after stopping the pump.
Genies	EMCTL_x	Boolean	False	1 = Hides the corresponding button on the genie and faceplate. x represents the name of the command. For example, EMCTL_Stop .

Time

The table describes the **Time** parameters that you can configure:

Element Name	Name	Type	Default value	Description
Flow-CtlPattern	SwitchOnDelay	Duration	00:00:03	Delay between the end of the starting sequence of a pumping asset and start of the starting sequence of the next pumping asset.
	DrainingTime	Duration	00:00:05	Time for which drain valve will be open when stopping the pumping asset.
	DelayAfterValveOpen	Duration	00:00:03	Delay time before next action after opening valves.
	DelayAfterValveClose	Duration	00:00:03	Delay time before next action after closing valves.
	DelayAfterPumpStart	Duration	00:00:03	Delay time before next action after starting pumps.
	DelayAfterPumpStop	Duration	00:00:03	Delay time before next action after stopping pumps.

Basic Alarm Configuration

The table describes the **Basic Alarm Configuration** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Name_x	String	-	Name_x : Name of the physical device associated with the alarm. Corresponds to the <i>Alarm Name</i> property of advanced alarms of the Supervision Participant. For example, Name_IDLE is for ControlTags and Name_EM_ALARM is for ParameterTags Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
ParameterTags				
ControlTags	Category	String	SGC_CAT_EVENT	Category of alarm/event for all states. Alarm category label or number. Corresponds to the category property of advanced alarms of the Supervision Participant.
ParameterTags	CAT_x	String	Label with the syntax: SGC_CAT_<type of component>_<type of alarm>	CAT_x : Alarm category label or number. Corresponds to the <i>Category</i> property of advanced alarms of the Supervision Participant. For example, for the parameter Category_EM_ALARM the label is SGC_CAT_FC_ALARM . Refer to Alarm Category Labels and Numbers (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Supervision Services Reference Manual).
ControlTags	Help	String	-	Help of alarm/event for all states. Name of the graphics page that opens when the help function is called. Corresponds to the <i>Help</i> property of advanced alarms of the Supervision Participant.
ParameterTags	Help_x	String	-	Help_x : Name of the graphics page that opens when the help function is called. Corresponds to the <i>Help</i> property of advanced alarms of the Supervision Participant. For example, Help_EM

Element Name	Name	Type	Default value	Description
ParameterTags	Delay_X	Duration	00:00:00	Delay_x : Alarm delay period. Corresponds to the <i>Delay</i> property of advanced alarms of the Supervision Participant. For example, Delay_EM_Fail .
<div> <div>X</div> <div> <p>Suffix of the variable tag name or advanced alarm tag that is managed by the corresponding Supervision components. For example, Name_EM_ALARM correspond to the <i>EM Name_FLOWCTL_EM_ALARM</i> advanced alarm tag managed by the <i>FLOWCTL</i> Control resource.</p> <p>Refer to the description of tags of the Supervision component (see EcoStruxure™ Process Expert LVPC Library, Supervision Services User Guide).</p> </div> </div>				

Advanced Alarm Configuration

The table describes the **Advanced Alarm Configuration** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Privilege	String	Label with the syntax: <i>SGC_PRIV_<type of component>_<type of action></i>	Privilege level required to acknowledge or disable the alarm. Corresponds to the <i>Privilege</i> property of advanced alarms of the Supervision Participant. For example, <i>SGC_PRIV_EVENT</i> . NOTE: For advanced alarms that are process events in the label <i><type of component></i> and <i><type of action></i> are replaced by EVENT , for example SGC_PRIV_EVENT .
ParameterTags	Privilege_x	String	Label with the syntax: <i>SGC_PRIV_<type of component>_<type of action></i>	Privilege_x : Privilege level required to acknowledge or disable the alarm. Corresponds to the <i>Privilege</i> property of advanced alarms of the Supervision Participant. For example, for Privilege_EM_ALARM , the label is SGC_PRIV_D_ALAR . Refer to Privilege Levels and Labels (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Supervision Services Reference Manual).
	Custom1_x to Custom8_x	String	-	Custom1_x to Custom8_x : String to filter active alarms (64 characters maximum). Corresponds to the <i>Custom 1 to Custom 8</i> property of advanced alarms of the Supervision Participant. For example, Custom1_EM_ALARM Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
	Paging_x	Boolean	False	Paging_x : Read/write property, which indicates the paging status of the alarm. Corresponds to the <i>Paging</i> property of advanced alarms of the Supervision Participant. For example, Paging_EM_ALARM . 1 = The alarm is paged.
	PagingG_x	String	-	PagingG_x : A read-only text string that indicates the paging group to which the alarm belongs (80 characters maximum). Corresponds to the <i>Paging Group</i> property of advanced alarms of the Supervision Participant. For example, PagingG_EM_ALARM .
<div> <div>X</div> <div> <p>Suffix of the variable tag name or advanced alarm tag that is managed by the corresponding Supervision components. For example, Paging_EM_ALARM correspond to the <i>EM Name_FLOWCTL_EM_ALARM</i> advanced alarm tag managed by the <i>FLOWCTLPATTERN</i> Control resource.</p> <p>Refer to the description of tags of the Supervision component (see EcoStruxure™ Process Expert LVPC Library, Supervision Services User Guide).</p> </div> </div>				

Historize

The table describes the **Historize** parameters that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Historize_x	Boolean	False	Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . x represents the child elements of the <i>EMCTL_ST</i> and <i>EMCTL_CFG</i> data structures, and the <i>EMCTL_CSTEPPD</i> variable. For example, Historize_STW .
	Historize_y			Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . y represents the Strategy execution state alarms. For example, Historize_IDLE . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
ParameterTags	Historize_EM_ALARM	Boolean	False	Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
	Historize_EM_FAIL			
	Historize_x_ALARM			Corresponds to the <i>Historize</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Historize</i> property of the corresponding tag to <i>True</i> .
	Historize_x_FAIL			x represents the Pumping Asset 01 to 05 (ASSET01 to ASSET05). For example, Historize_ASSET01_ALARM and Historize_ASSET01_FAIL . Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
InitialConditionTags and FailureConditionTags	Historize_BYPASSW	Boolean	False	Enable historize bypass word.
	Historize_CONDW	Boolean	False	Enable historize condition word.
	Historize_REARMREQW	Boolean	False	Enable historize rearm word.

Alarm Tag Disable

The table describes the **Alarm Tag Disable** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
ControlTags	Disabled_x	Boolean	True	Corresponds to the <i>Alarm Tag Disable</i> property of advanced alarm tags of the Supervision Participant. 1 = Sets the <i>Alarm Tag Disable</i> property of the corresponding tag to False. x represents the name of the state. For example, Disabled_RUNNING .
ParameterTags	Disabled_EM_ALARM	Boolean	True	Corresponds to the <i>Alarm Tag Disable</i> property of advanced alarm tags of the Supervision Participant.
	Disabled_EM_FAIL			1 = Sets the <i>Alarm Tag Disable</i> property of the corresponding tag to False. Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.
	Disabled_x_ALARM			Corresponds to the <i>Alarm Tag Disable</i> property of advanced alarm tags of the Supervision Participant.
	Disabled_x_FAIL			1 = Sets the <i>Alarm Tag Disable</i> property of the corresponding tag to False.

Element Name	Name	Type	Default value	Description
				<p>x represents the Pumping Asset 01 to 05 (ASSET01 to ASSET05). For example, Disabled_ASSET01_ALARM and Disabled_ASSET01_FAIL.</p> <p>Refer to <i>Advanced Alarm Properties</i> in the help of the Supervision Participant for details.</p>
<p>NOTE:</p> <p>Values of Boolean parameters are set by using check boxes:</p> <ul style="list-style-type: none"> Selected = True Cleared = False 				

Variable Tag Disable

The table describes the **Variable Tag Disable** parameters for Supervision component that you can configure:

Element Name	Name	Type	Default value	Description
InitialCondition- Tags	BYPASSWordSe- lection	Boolean	False	Disable initial conditions bypass tags.
	REARMREQ- WordSelection	Boolean	False	Disable initial condition resetting (REARM) tags.
FailureCondi- tionTags	BYPASSWordSe- lection	Boolean	False	Disable conditions bypass.
	REARMREQ- WordSelection	Boolean	False	Disable conditions manual resetting.
<p>NOTE:</p> <p>Values of Boolean parameters are set by using check boxes:</p> <ul style="list-style-type: none"> Selected = True Cleared = False 				

The following optional elements feature configurable parameters:

- **InitialConditionTags:** Messages associated to initial conditions, to appear on the Initial Conditions tab of the faceplate for flow control pattern.
- **FailureConditionTags:** Messages associated to detected failure conditions, to appear on the detected failure Conditions tab of the faceplate for flow control pattern.

Composition

The `FlowCtlPattern` template references composite and facet templates, which provide the following services:

Control	Core services, plus additional optional services, which you can activate if needed.
Supervision	These services complement the Control services. Supervision services are optional.

The table describes the services that are available from the `FlowCtlPattern` template and the corresponding facet, which implements the service:

Control services (FlowCtlPattern_UC)	Corresponding facet template	Control service description	Associated Supervision services (FlowCtlPattern_CS)*	Corresponding facet template	Supervision service description
Core services					
Logic	\$EMCTL_UL	Refer to the EMCTL DFB*.	ControlTags	FlowCtlPatternCTL_CD	Refer to tags of Equipment Module Management**.
-	-	-	Genies	\$EMPANELState_CG	Refer to the genie representation of the Equipment Module Management**.
				\$EMPANELWithButtons_CG	
				\$EMPANELAllData2_CG	
				\$EMPANELAllData4_CG	
FlowCtlPattern	FlowCtlPattern_UL	Refer to the FLOWCTLPATTERN DFB*.	ParameterTags	FlowCtlPatternPAR_CD	Refer to tags of Flow Control Pattern Management.
Optional Services					
InitialConditions	\$CONDSUM_UL	Refer to the CONDSUM DFB***.	InitialConditionTags	\$INITCONDSUM_CD	-
FailureConditions	\$CONDSUM_UL	Refer to the CONDSUM DFB***.	FailureConditionTags	\$CONDSUM_CD	
* See EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Control Services Reference Manual					
** See EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual					
*** See EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual					

NOTE: When you activate an optional Control service, the corresponding Supervision service is automatically activated, if available.

The control module template exposes the following interface.

Interface Identifier	Type/Role	Description
EMCTLName	\$EMCTLName/Def	Equipment module control name.
FlowSP	\$Real/Ref	Flow set point.
FlowPV	\$Real/Ref	Flow process value.
Pump0x	\$SDDEVCTL_IF/Ref	Pumping asset 0x pump.
InletValve0x	\$DEVCTL_IF/Ref	Pumping asset 0x inlet valve.
OutletValve0x	\$DEVCTL_IF/Ref	Pumping asset 0x outlet valve.
DrainValve0x	\$DEVCTL_IF/Ref	Pumping asset 0x drain valve.
NOTE: x indicates values from 1 to 5.		

Below table shows the list of templates which are available for modification.

Name of Facets, Composites and Templates	Description	Comment
FlowCtlPattern_UL	To create flow control user logic and configure input and output parameters for control component as per requirement.	To modify this facet template you have to Save As and rename it as per requirement.
FlowCtlPattern_UC	If FlowCtlPattern_UL is modified then FlowCtlPattern_UC has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
FlowCtlPatternPAR_CD	To configure the input and output parameters Flow Control for supervision component as per requirement.	To modify this composite template you have to Save As and rename it as per requirement.
FlowCtlPatternCTL_CD	To configure the Custom1...8, Paging, PagingG parameters of alarms of Flow Control equipment module states and strategy execution states.	To modify this composite template you have to Save As and rename it as per requirement.
FlowCtlPattern_CD	To configure the Strategy and EMstate variables. Also, to configure Disable, Historize and Name parameters of alarms of Equipment Module states and strategy execution states. If FlowCtlPatternPAR_CD or FlowCtlPatternCTL_CD is modified then FlowCtlPattern_CD has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
FlowCtlPattern_CS	To configure the DATA1 to DATA4 parameters. The parameters DATA1 to DATA4 can be any input/output parameters which can be viewed in genie for monitoring purpose. If FlowCtlPattern_CD is modified then FlowCtlPattern_CS has to be modified.	To modify this composite template you have to Save As and rename it as per requirement.
FlowCtlPattern	Main Flow Control Pattern template. If FlowCtlPattern_UC or FlowCtlPattern_CS is modified then FlowCtlPattern has to be modified	To modify this composite template you have to Save As and rename it as per requirement.

Default Configuration available for FlowCtlPattern Template

The template is provided with strategies, user defined Equipment Module States, input and output parameters, and four configurable data parameters. The template can be modified as per its application requirements.

The Input parameters and their data types are as follows:

Input Parameter	Data Type (user)
FlowSP	Numeric (Real)
Reset Pumps Operation Time	Boolean
Priority Pump 1	Numeric (Signed integer)
Priority Pump 2	Numeric (Signed integer)
Priority Pump 3	Numeric (Signed integer)
Priority Pump 4	Numeric (Signed integer)
Priority Pump 5	Numeric (Signed integer)

The output parameters and their data types are as follows:

Output Parameter	Data Type (user)
Equipment Module Diagnosis State	Enumeration
Flow Present Value	Numeric (Real)
Current Flow Setpoint	Numeric (Real)
Pumps Required To Achieve Flow setpoint	Numeric (Signed integer)

Output Parameter	Data Type (user)
Active Pumps To Achieve Flow Setpoint	Numeric (Signed integer)
Total active pumps	Numeric (Signed integer)
Pump 01 Speed Setpoint	Numeric (Real)
Pump 01 Operation Time	Duration
Pumping Asset 01 State	Enumeration
Pump 02 Speed Setpoint	Numeric (Real)
Pump 02 Operation Time	Duration
Pumping Asset 02 State	Enumeration
Pump 03 Speed Setpoint	Numeric (Real)
Pump 03 Operation Time	Duration
Pumping Asset 03 State	Enumeration
Pump 04 Speed Setpoint	Numeric (Real)
Pump 04 Operation Time	Duration
Pumping Asset 04 State	Enumeration
Pump 05 Speed Setpoint	Numeric (Real)
Pump 05 Operation Time	Duration
Pumping Asset 05 State	Enumeration

NOTE: For the Control and Supervision data types of input and output parameters, page 26.

For the Output parameters with the Enumeration data type, their variable tags are configured as:

- *Custom1* - *@(Normal):0*
- *Custom2* - *@(Warning):1*
- *Custom3* - *@(Failure):2*

The default value configured for DATA 1 to DATA 4 genie parameters are Output parameter 1 to Output Parameter 4 respectively.

The applicable parameters for default Strategy are:

Strategy	Applicable Input Parameters
Program SP/Balance	Reset Pumps Operation Time
Operator SP/Balance	Flow SP and Reset Pumps Operation Time
Program SP/Priority	Reset Pumps Operation Time, Priority Pump 01, Priority Pump 02, Priority Pump 03, Priority Pump 04, Priority Pump 05.
Operator SP/Priority	Flow SP, Reset Pumps Operation Time, Priority Pump 01, Priority Pump 02, Priority Pump 03, Priority Pump 04, Priority Pump 05.
Recalculate	Reset Pumps Operation Time
Stop Equipment	Reset Pumps Operation Time

The Equipment Module states are:

Equipment Module State	Description
Predefined Equipment Module states	
EMState0	Off
EMState1	Stopped
EMState10	Stopping
User defined Equipment Module states	

Equipment Module State	Description
EMState16	Running Program SP/Balance
EMState17	Running Operator SP/Balance
EMState18	Running Program SP/Priority
EMState19	Running Operator SP/Priority
EMState20	Configuring
EMState21	Running externally

Modifying the FlowCtlPattern Template

Steps to modify FlowCtlPattern template for the Flow Control Operation

The FlowCtlPattern template is provided with 5 pumping assets. Each asset includes one variable speed pump and three valves (Inlet, Outlet and Drain valve). Steps to modify the FlowCtlPattern template are identical to the modification of PumpSetPattern template. For additional information, page 43.

Index

E

EMPattern	
template description	22, 35
Equipment Module Pattern	
EMPattern	22, 35
Equipment Module templates	
composition strategy	12
description	12
list of templates	19

F

Flow Control Pattern	
FlowCtlPattern	48
FlowCtlPattern	
template description	48

P

process templates	
naming convention	14

T

templates	
list of Equipment Module templates	19

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.

EIO0000003016.06