

GPL Classic >> GPL Situational Awareness

Guidelines to transition an application

Ecostruxure Hybrid DCS 2019 released with a new general process library as its default library. This is a new library and while functionally like previous general process libraries there is no direct upgrade between the libraries.

If projects or customers want to leverage the new general process library, they must transition their project to replace the libraries and re-draw their graphics screens. This document provides a description of the process the user must undertake to handle the transformation of their project to get it running with the new library.

Transition Process

The aim of the transition process is to provide a new system with the functionality of the existing system. A second system is created to ensure the project team continue to have the existing project as a reference and validation source.

New System Libraries

The engineer creates a new system for the upgraded project. The transformation process will progressively engineer this system based on the content of the existing system.

As the process progresses it is necessary to display content from both systems at the same time to simplify the transformation of refinement between the system. It is therefore recommended that the transition be done on a computer with two monitors where the two systems can be displayed side by side for comparison.

Install Libraries

To start the transition process it is important that both libraries are available within the global library repository. Ecostruxure Hybrid DCS supports the existence of two items with the same name provided they have different templates and so it is possible to generate parallel objects based on each library.

To import libraries into Ecostruxure Process Expert open the Global Templates and select Import. For more information see the Ecostruxure Process Expert Help for "Import Templates". The libraries for both GPL and GPL SA are available in the Program Data directory of the installation.

Project Libraries

The GPL-SA library contains parameters which are designed to be set for each instance of an asset (description, failure conditions, interlock conditions) and others which are parameters to support customization but which are usually set once for the whole project (alarm description text, interlock group names, trend history settings) and some which can change based on the project and asset (security settings).

All the parameters are exposed by default within the library to show the capabilities of the library. It is intended that customers will choose to fix some of these parameters within the template. Doing this will ensure their value on all future assets as well as reducing the engineering effort on individual assets.

Library

Open the Global Template Editor

Create a library folder for the current project

Duplicate

Select the template for which parameters should be fixed.

Right click and duplicate the template (top level only)

Save the new template within the project library

Set

Open the project template and replace the “Deferred” symbol associated with the parameters to be set with a fixed value.

Save the template into the library folder

All assets should now be created with the library project template (and not the standard template)

The project should consider fixing the following parameters

- Alarm Descriptions
- Alarm Groups
- Security Settings
- Trend Historical Selection (excluding sample rate)
- Historization Selections

If a parameter is required to be available later in the project the deferred selection can be restored, and its value can be set via the bulk generation process. The value of fixed parameters is the reduction in knowledge required to add new assets to the system after the initial project engineering as there are less settings required by the engineer.

Transition Assets

This step replicates each asset in the existing control system within the new control system. It also enables the engineer to ensure that the parameters are maintained. While there is a high degree of overlap with control parameters between the two libraries most supervision parameters should either be fixed or require additional configuration.

Existing Assets

In the existing system use the Application Browser interface and select the grid view (not tree view) via the selection at the top right of the window. The display will now show all the assets defined as a list. Click on the Template to sort the templates into alphabetical order. Export a file for all assets associated to each template.

New Assets

In the new system use the Application Browser interface and select the tree view. Generate a sample asset for each template exported from the existing project but using the project library templates. The sample templates will be deleted at the end of the process so they should be created in a central location. Export each asset as per the existing assets

Create New Assets

Open the existing asset export file (with all the assets) and the new asset export file with the new template. Set each line in the new asset template to create an asset based on the project template. Include the name, description, location, area (all properties beginning with a \$ sign).

Replicate Parameters

Use the template guides below to copy relevant parameters between the two sets of templates. If the parameter does not exist within the GPL Classic asset, then duplicate it from the sample asset.

Import

Import the file into the new system. It will generate assets with matching names in the correct locations (beside) the existing assets. The names and structure will be identical, but the template and parameters will be different.

Transition Asset Links

With all the new assets complete it is now possible to replicate the links (edit links) which have been created to orchestrate the connection between the assets. The aim is to replicate all the existing links.

Export

Select all the assets within the existing project and export from the system. The links will appear at the end of the export file. Each link is represented by line in the file.

Migrate Links

Section :interfaceLink

\$SourceInstanceTemplateIdentifier is updated to the new template type

\$DestinationInstanceTemplateIdentifier is updated to the new template type

For the transition from GPL Classic to GPL SA this should simply require adding a GP as the suffix for both templates to create the new link. If the project template has a different name this needs to be used instead.

Refer to the template guide below to identify if any interface names also require modification

Delete

Remove all asset information as the asset configuration does not change.

Import

Import the file into the new system. It will generate links between assets matching the existing project

Assignment

With all the new assets created and linked its possible to assign the asset to the section and tag groups within the new project.

Code Section

Assign the new assets to controller sections. The simplest approach is to follow the assignment of the existing sections, but this is not required.

Tag Container

Create a new Tag Container and assign the new asset to the new container.

Refinement

Code Section

Open the code section from both the existing and new systems (two monitors). The exiting code can be used as a template to configure the new asset. The new GPL has different asset names (based on their updated DFB) so the variables will be different and need to be selected from the available assets.

Code segments which are used within the existing project as refinement can be copied into the new project using Copy & Paste. Types can be exported from the existing system and re-imported into the new project to generate DFBs and DDTs for refinement.

Sequence, Phase and Equipment blocks refer to internal blocks within their code. The new library adds a GP to these blocks and so the code itself must be updated to use these references.

Graphics

Create new Graphics Pages for SA graphics. Create the page hierarchy within the supervision. The existing graphics page can be displayed on the second monitor to speed development. The assets are dragged from the process tree onto the screen.

Build and Deploy

See existing documentation.

Templates

Analog Input

\$AnalogInput
\$AnalogInput1

>> None
>> \$AnalogInputGP

Control

AnalogInput1.AnalogInputSignal.Enabled

>> AnalogInput.AnalogInput.Enabled

Parameter.Control

AnalogAlarms.Logic.DWATCH Selection
AnalogAlarms.Logic.HHWATCH Selection
AnalogAlarms.Logic.HWATCH Selection
AnalogAlarms.Logic.LWATCH Selection
AnalogAlarms.Logic.LLWATCH Selection

>> AnalogAlarms.Logic.DWATCHSelection
>> AnalogAlarms.Logic.HHWATCHSelection
>> AnalogAlarms.Logic.HWATCHSelection
>> AnalogAlarms.Logic.LWATCHSelection
>> AnalogAlarms.Logic.LLWATCHSelection

AnalogAlarms.Logic.HysteresisSP
AnalogAlarms.Logic.TOFFSP
AnalogAlarms.Logic.TONSP

>> AnalogAlarms.Logic.HysteresisSetpoint
>> AnalogAlarms.Logic.OffDelay
>> AnalogAlarms.Logic.OnDelay

AnalogInput1.AnalogInputSignal.*
AnalogInput1.Logic.*

>> AnalogInput.AnalogInput.*
>> AnalogInput.Logic.*

Supervision

Tag.*

>> Data.*

Interfaces

Variables