طرح نگهداشت و افزایش تولید ۲۷ مخزن

Functional Design Specification-DCS/ESD Software

نگهداشت و افزایش تولید میدان نفتی بینک

V00	Mar. 2025	IFA	IGK	M.Fakharian	S.Faramarzpour	
Rev.	**Date**	**Purpose of Issue/Status**	**Prepared by:**	**Checked by:**	**Approved by:**	**CLIENT Approval**

Status:

IFA: Issued For Approval
AFC: Approved For Construction
IFI: Issued For Information

REVISION RECORD

Page	V00	01	02	03	04	05	06	07	Rev.	V00	01	02	03	04	05	06	07
1	x								51	x							
2	x								52	x							
3	x								53	x							
4	x								54	x							
5	x								55	x							
6	x								56	x							
7	x								57	x							
8	x								58	x							
9	x								59	x							
10	x								60	x							
11	x								61	x							
12	x								62	x							
13	x								63	x							
14	x								64	x							
15	x								65	x							
16	x								66	x							
17	x								67	x							
18	x								68	x							
19	x								69	x							
20	x								70	x							
21	x								71	x							
22	x								72	x							
23	x								73	x							
24	x								74	x							
25	x								75	x							
26	x								76	x							
27	x								77	x							
28	x								78	x							
29	x								79	x							
30	x								80	x							
31	x								81	x							
32	x								82	x							
33	x								83	x							
34	x								84	x							
35	x								85	x							
36	x								86	x							
37	x								87	x							
38	x								88	x							
39	x								89	x							
40	x								90	x							
41	x								91	x							
42	x								92	x							
43	x								93	x							
44	x								94	x							
45	x								95	x							
46	x								96	x							
47	x								97	x							
48	x								98	x							
49	x								99	x							
50	x								100	x							

REVISION RECORD

Page	00	01	02	03	04	05	06	07	Rev.	00	01	02	03	04	05	06	07
101	X								151	X							
102	X								152	X							
103	X								153	X							
104	X								154	X							
105	X								155	X							
106	X								156	X							
107	X								157	X							
108	X								158	X							
109	X								159	X							
110	X								160	X							
111	X								161	X							
112	X								162	X							
113	X								163	X							
114	X								164	X							
115	X								165	X							
116	X								166	X							
117	X								167	X							
118	X								168	X							
119	X								169	X							
120	X								170	X							
121	X								171	X							
122	X								172	X							
123	X								173	X							
124	X								174	X							
125	X								175	X							
126	X								176	X							
127	X								177	X							
128	X								178	X							
129	X								179	X							
130	X								180	X							
131	X								181	X							
132	X								182	X							
133	X								183	X							
134	X								184	X							
135	X								185	X							
136	X								186	X							
137	X								187	X							
138	X								188	X							
139	X								189	X							
140	X								190	X							
141	X								191	X							
142	X								192	X							
143	X								193	X							
144	X								194	X							
145	X								195	X							
146	X								196	X							
147	X								197	X							
148	X								198	X							
149	X								199	X							
150	X								200	X							

REVISION INDEX

Rev.	00	01	02	03	04	05	06	07	Rev.	00	01	02	03	04	05	06	07
Page									Page	X							
201	X								251	X							
202	X								252	X							
203	X								253	X							
204	X								254	X							
205	X								255	X							
206	X								256	X							
207	X								257	X							
208	X								258	X							
209	X								259	X							
210	X								260	X							
211	X								261	X							
212	X								262	X							
213	X								263	X							
214	X								264	X							
215	X								265	X							
216	X								266	X							
217	X								267	X							
218	X								268	X							
219	X								269	X							
220	X								270	X							
221	X								271	X							
222	X								272	X							
223	X								273	X							
224	X								274	X							
225	X								275	X							
226	X								276	X							
227	X								277	X							
228	X								278	X							
229	X								279	X							
230	X								280	X							
231	X								281	X							
232	X								282	X							
233	X								283	X							
234	X								284	X							
235	X								285	X							
236	X								286	X							
237	X								287	X							
238	X								288	X							
239	X								289	X							
240	X								290	X							
241	X								291	X							
242	X								292	X							
243	X								293	X							
244	X								294	X							
245	X								295	X							
246	X								296	X							
247	X								297	X							
248	X								298	X							
249	X								299	X							
250	X								300	X							

REVISION INDEX

Rev.	00	01	02	03	04	05	06	07	Rev.	00	01	02	03	04	05	06	07
Page									Page	X							
301	X								351	X							
302	X								352	X							
303	X								353	X							
304	X								354	X							
305	X								355	X							
306	X								356	X							
307	X																
308	X																
309	X																
310	X																
311	X																
312	X																
313	X																
314	X																
315	X																
316	X																
317	X																
318	X																
319	X																
320	X																
321	X																
322	X																
323	X																
324	X																
325	X																
326	X																
327	X																
328	X																
329	X																
330	X																
331	X																
332	X																
333	X																
334	X																
335	X																
336	X																
337	X																
338	X																
339	X																
340	X																
341	X																
342	X																
343	X																
344	X																
345	X																
346	X																
347	X																
348	X																
349	X																
350	X																

Contents

1 Proprietary Statement

Binak oilfield in Bushehr province is a part of the southern oilfields of Iran, is located 20 km northwest of Genaveh city.

With the aim of increasing production of oil from Binak oilfield, an EPC/EPD Project has been defined by NIOC/NISOC and awarded to Petro Iran Development Company (PEDCO). Also PEDCO (as General Contractor) has assigned the EPC-packages of the Project to "Hirgan Energy - Design and Inspection" JV..

1.1. DEFINITIONS

The following terms shall be used in this document.

CLIENT:	National Iranian South Oilfields Company **(NISOC)**
PROJECT:	DCS and ESD Control System of Binak Gas Booster Station
EPD/EPC CONTRACTOR (GC):	Petro Iran Development Company **(PEDCO)**
EPC CONTRACTOR/PURCAHSER:	Joint Venture of: Hirgan Energy – Design & Inspection Companies **(HE/DI)**
VENDOR:	IDEH GLOBAL KISH (IGK)
EXECUTOR:	Executor is the party which carries out all or part of construction and/or commissioning for the project.
TPI:	Third-Party Inspector
SHALL:	Is used where a provision is mandatory.
SHOULD:	Is used where a provision is advisory only.
MAY:	Is used where a provision is completely discretionary.

1.2. FDS Purpose

The purpose of the FDS is to specify how the requirements for the IPCS for the **BINAK Projec**t will be met by the HONEYWELL &EMERSON system.

1.3. FDS Scope

The scope of the overall FDS encompasses the functional design of the complete IPCS. This FDS will:

- Provide full details of functionality, performance and self-checking that will be available from the system.
- Provide a full definition of all interfaces between the IPCS and all interconnected subsystems.

1.4. Project Scope

The project scope encompasses system design and testing of the complete Integrated Process Control System (IPCS) to monitor and control the plant and interface to a number of third party packages. IGK will be supplied with standard Honeywell & EMERSON hardware packages for system configuration. Include GPS, operator stations, desks & chairs, large screens, engineering station, network cabinet that will be supplied by IGK as per contract accordingly.

1.5. Abbreviations

Abbreviations, used in connection with FDS Document, are defined in the following table:

Acronym	Description
ACE	Application Control Environment
ACM	Alarm Configuration Manager
AE	Alarm & Event
AI	Analogue Input
AMS	Asset Management System
AO	Analogue Output
APC	Advanced Process Control
ASM	Abnormal Situation Management
BIN	Business Information Network
CAB	Custom Algorithm Block
CCR	Central Control Room
CDA	Control Data Access
CDB	Custom Data Block
CEE	Control Execution Environment
CE-STN	Console Extension Stations
C-STN	Console Station
DC	Domain Controller
DCS	Distributed Control System
DI	Digital Input
DO	Digital Output
DSA	Distributed System Architecture
DTM	Device Type Manager
EBR	Experion Backup and Restore
EMDB	Enterprise Model Database
ERDB	Engineering Repository Database

Acronym	Description
ESD	Emergency Shutdown
FAR	Field Auxiliary Room
FDS	Functional Design Specification
FDM	Field Device Manager
FED	Front End Design
	Fire & Gas System
FIM	Field Interface Module
FTA	Field Termination Assembly
FTE	Fault Tolerant Ethernet
FS	File Server
F-STN	Flex Station
GB	Giga Byte
GPS	Global Positioning System
HART	Highway Addressable Remote Transducer
HCSL	Honeywell Control Systems Ltd.
HMI	Human Machine Interface
IOM	Input Output Module
IOTA	Input Output Termination Assembly
IP	Internet Protocol
IPS	Instrument Protective System
IPCS	Integrated Process Control System
IS	Intrinsic Safety
IT	Information Technology
KBps	Kilo Bits per Second
KVM	Keyboard, Video and Monitor
LAN	Local Area Network
MOS	Maintenance Override Switch
MOV	Motor Operated Valve
MTTSP	Methods, Techniques, Tools, Standards and Products
NIC	Network Interface Card
IOLIM	I/O Link Interface Module
ODS	Operational Data Supervision System
OPC	OLE for Process Control
OPM	On Process Migration
OTS	Operator Training System
PCN	Process Control Network

Acronym	Description
PI	Processing Instruction
PKS	Process Knowledge System
PM	Process Management
PPS	Parameters per Second
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RCI	Remote Communication Interface
ROM	Read only Memory
RTU	Remote Telemetry Unit
RUIP	Removal and Insertion Under Power
SCADA	Supervisory Control and Data Acquisition
SER	Sequence of Events Recorder
SM	Safety Manager
SNTP	Simple Network Time Protocol
SOE	Sequence of Events
TDAS	Tank Data Acquisition System
TPC	Total Plant Configurator
TPI	Tap Position Indication
UAPC	Universal Advanced Process Control
UTC	Universal Time Coordinated
VAC	Volts Alternating Current
VDC	Volts Direct Current
VLAN	Virtual Local Area Network
WAN	Wide Area Network
WMI	Windows Management Instrumentation

2 PKS Software Overview

2.1 About Experion applications

The Experion R501 system includes the Configuration Studio, which provides a central location from which you can configure your Experion system. The individual tools required to configure parts of your system are launched from Configuration Studio. These tools include the following applications.

• Control Builder

• HMIWeb Display Builder

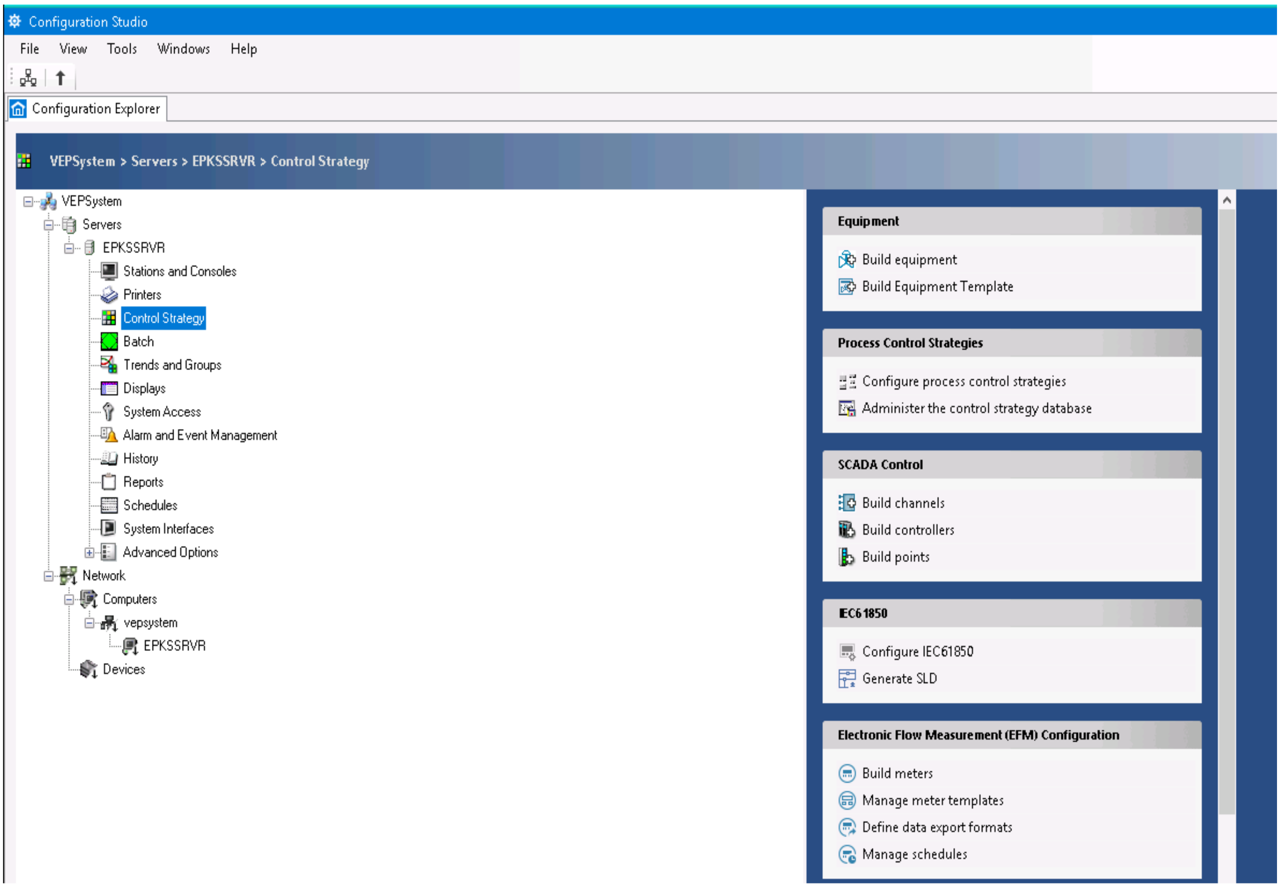
• Quick Builder

• Station

In order to implement a control application, several Experion Tools are required. These tools are located under the umbrella of the Configuration Studio application.

Configuration Studio provides a centralized location, from where the Experion PKS system is configured. It bundles all required tools and applications for the system configuration in general and for the control application implementation in particular. As soon as the user is selecting a configuration task, the appropriate configuration tool is launched.

Configuration Studio is installed as part of the Experion PKS installation.



Top Figure: Configuration Studio Main Window

Control Builder is a graphical function plan oriented software implementation package according to the guidelines of IEC61131.3, supporting Experion Controller system design, documentation and monitoring of the application.

All functions are implemented by using standard function blocks, which are provided in several libraries for several purposes (like logic, math, regulatory control, data acquisition etc.). The user just picks the appropriate function blocks and configures those in order to adjust their function to meet the project specific requirement. In case required custom algorithm blocks can be created in order to implement a more complex algorithm into one customized function block.

Control Builder provides three different representations of the project database. These representations are called “views”. In the Project View, the hierarchical representation of the offline structure of the application project the initial control application configuration is implemented. The Monitoring View provides the capability to monitor the application downloaded into a controller on-line. The third view is the Library, which contains all installed function blocks libraries and their function blocks.

In Project and Monitoring View the user can toggle between Assignment View (hardware oriented hierarchical representation of the software structure) and a Containment View (logical derivation and containment representation of the software structure). For the Library a similar selection exists and it called Containment View (which block belongs to which library) and Derivation View (parent – child relationship representation in a hierarchical manner).

Control Strategies are realized as a combination of function blocks which contain several configurable parameters per function block are which are connected via “soft-wires” in order to form the required function. Function blocks are either contained in the hardware hierarchy (like controllers, I/O cards etc.) or container in a CFC or SFC (for sequences). In Control Builder those container function plans are called Control Module and Sequential Control Module. Complex control strategies can span across multiple CM and/or SCM.

Each object in the Control Builder hierarchy is independent and will not affect other objects unless those objects are contained in it or those objects have connections to it. This is called control module independence principle and allows downloading any object of a control strategy during run time of the process without effects to other unrelated objects.

Object parameters in Control Builder could be referenced by other objects using an object oriented addressing structure like:

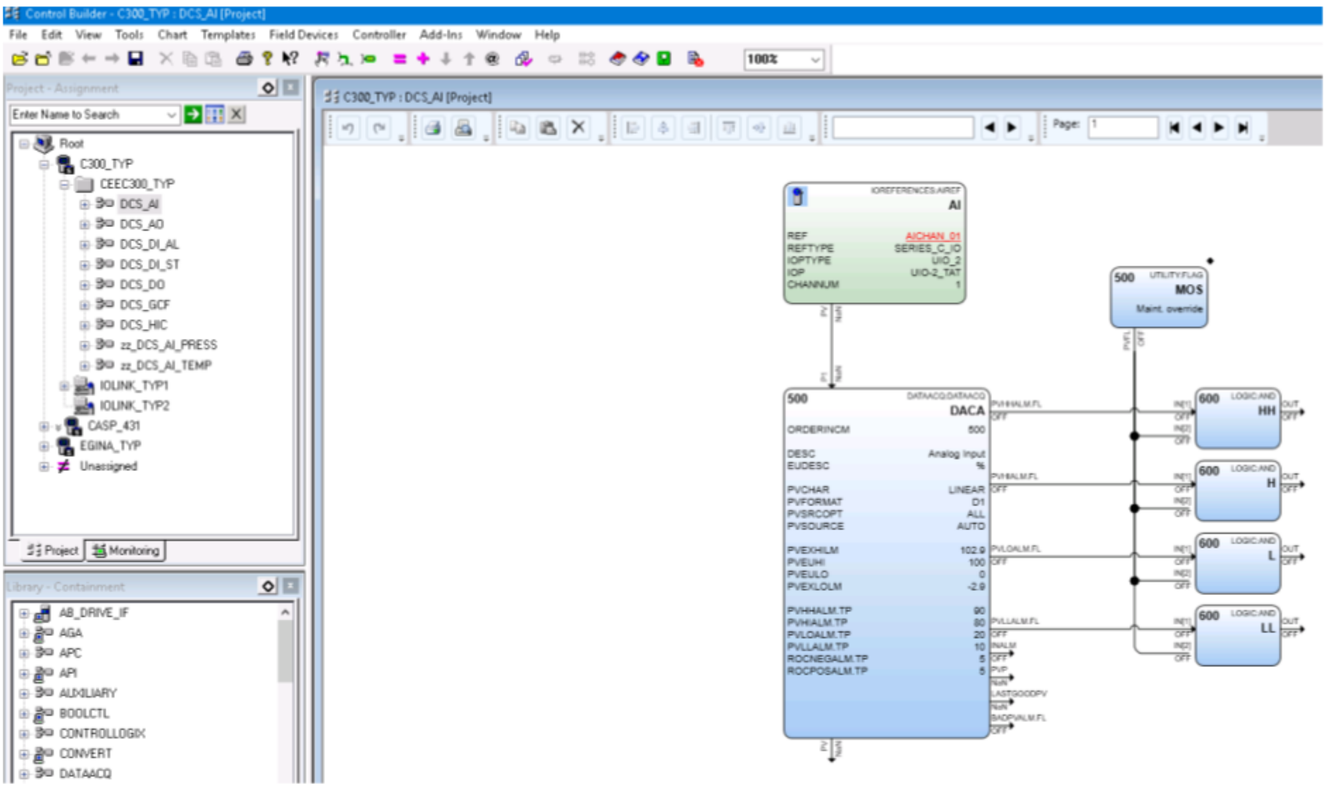
<Object Name>.<Contained Object Name>.<..>. Parameter

In case of a PID-function block (PIDA) set point (SP) contained in a CM called TC4711 the reference to this set point is: TC4711.PIDA.SP.

Control Builder provides a comprehensive set of tools for handling I/Os, Logic, Motor, Sequential, Batch and advanced control functions through a library of Function Blocks (FBs). FBs are basic objects provided by Honeywell to execute different control functions. FBs are interconnected using "soft wires" to construct control applications or strategies. FBs are grouped together and contained in CM and in case of sequential FBs, SCM are present.

All logic programming is done using FBs. Each block contains parameters, which are used to define certain attributes of that function block. All real hardware I/Os (AI, DI, AO, and DO) are also function blocks of component type. The parameters of these function blocks are connected to other function blocks (PID, AND, OR, FLAG, NUMERIC etc.) as per logic requirement to build a control strategy. All these blocks are contained in CMs. The control strategy can span across multiple CMs in multiple controllers. Since Experion PKS is based on Object Oriented system environment, it accordingly follows the rules for referring variables and parameters. The parameters of the CMs are referred as **<CM name>.<Function Block Name>.<Parameter Name>**.

Control Builder implements a transparent peer-to-peer handling. As a result any function block can reference any other function block / parameter in the whole system in the same manner as it would be done locally. The controller takes care of establishing the required connectivity. Peer-to-peer is slower than a local data transportation (usually once per 500ms) and limited in the number of connections (500 per CEE). It shall be used as rarely as possible.



Top Figure: Typical Control Module Window

For full description of Control Module Loading Procedure, please see attachment #2 at the end of this document.

2.2 HMIWeb Display Builder

The HMIWeb display builder is the package used to build graphical displays for Experion PKS in an HTML format. HMIWeb Display Builder is an object based fully integrated custom display builder for development of application specific graphics. It is used to configure the graphics for Operator Interface. The graphics can be animated and linked to plant processes to represent real-time events.

HMIWeb Display Builder is a powerful tool for developing html-based operator graphics. It is used to create custom process graphics used to control a plant, mill, process area etc.

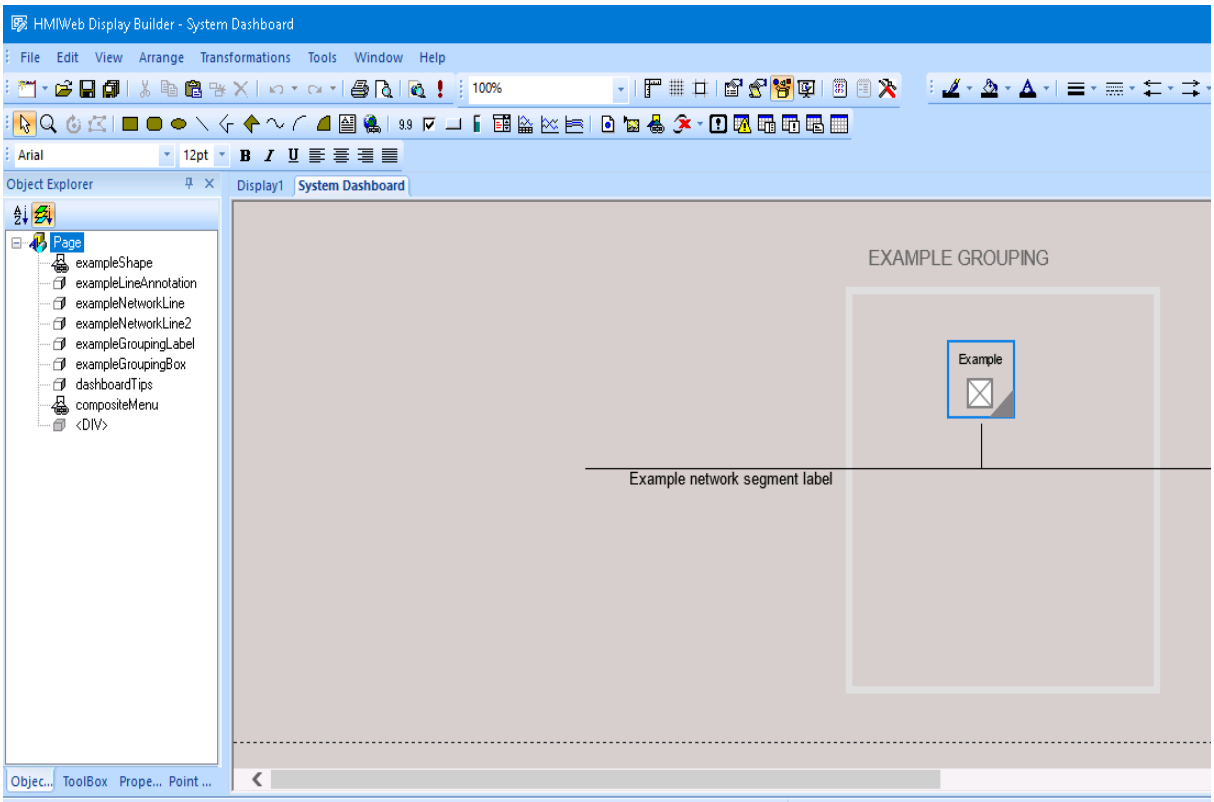
The HMI Web Display Builder tool provides a Display Area for the user to modify and create custom graphics.

An object explorer is provided to present a tree view of all the objects in a display.

This tool is equipped with standard functions for building, modifying and saving the display. Although majority of the animation requirements is met with native functions provided by the HMI Web Display Builder, display scripts can also be employed when required.

A shape gallery is used to make it easy to insert a shape sequence or a dynamic shape. Popup and Faceplates are also created using standard tools in the HMIWeb Display Builder.

Popup is the secondary window that appears when user clicks the object to which it is attached. Faceplates are specialized type of popup that shows critical information of the object to which it is linked.



2.3 Quick Builder

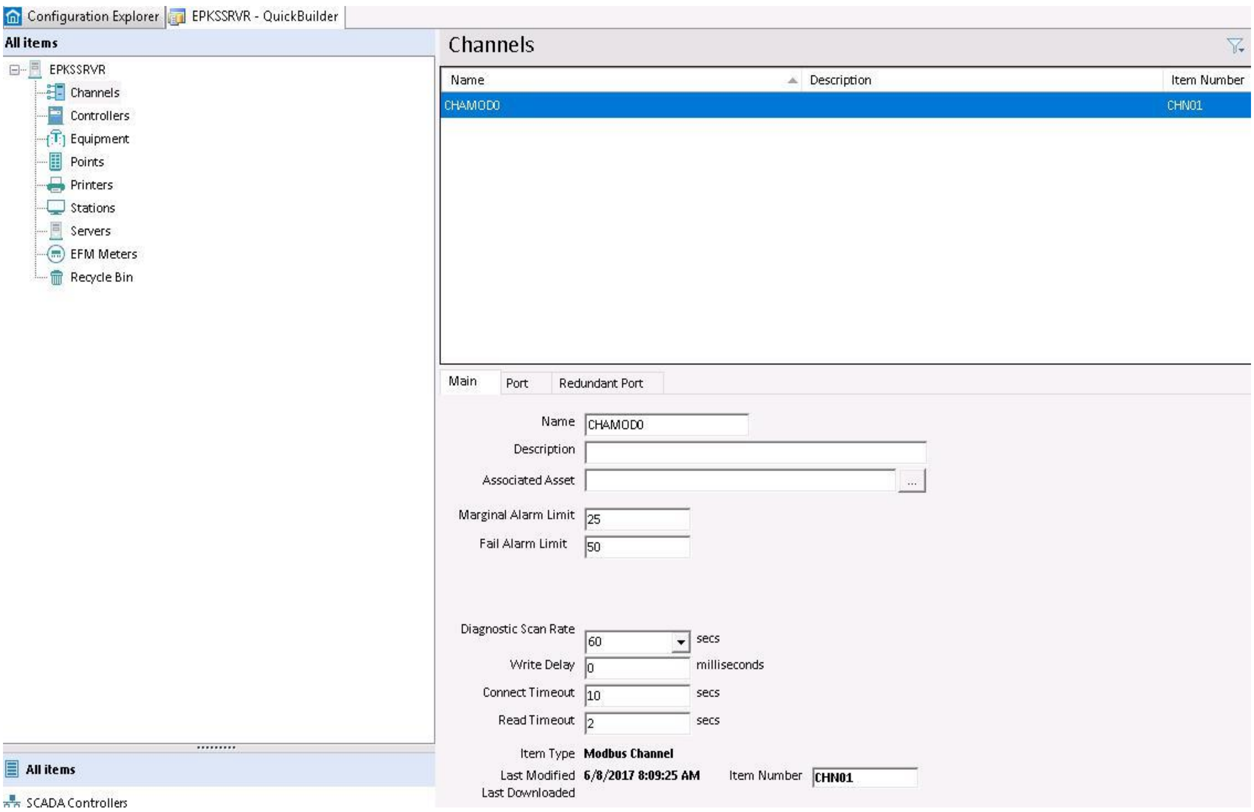
Quick builder is an Experion PKS software utility. Quick Builder is used for configuring SCADA points, Stations, Printers and Channel and controller for PLC and third party interface. Quick Builder utility can be opened from server.

After defining points with Quick Builder, user has to download these definitions from Quick Builder to the server database.

When user downloads a project or part of it to the server, it becomes part of the configuration database. The configuration database defines how each component in the system is configured.

If necessary, user can update the project and repeat the download process. Alternatively, user can upload data from the configuration database into Quick Builder, edit it and then download it back to the server.

Following Figure shows the general graphic orientation and Windows look-and-feel of the Quick Builder application for reference



Top Figure : Quick Builder Application Window

SCADA Points are configured for interfacing PLCs and 3rd Party packages to the Experion PKS Server. There are a number of types of SCADA points:

- Analogue
- Status
- Accumulator
- Container

Each of the Analogue, Status and Accumulator types has a predefined set of parameters available, many of which can be assigned source or destination addresses referencing data resident in 3rd party controllers.

Container points provide a means of grouping multiple SCADA points into a single Container Point, with configurable parameter references. These structures may be used for repeated functional units to promote reusable shapes and detail displays.

2.4 Station

Station is Experion's user interface. You use Station to monitor and control your system. Station displays are used to notify operators of alarm conditions. Station works in conjunction with Experion server, which:

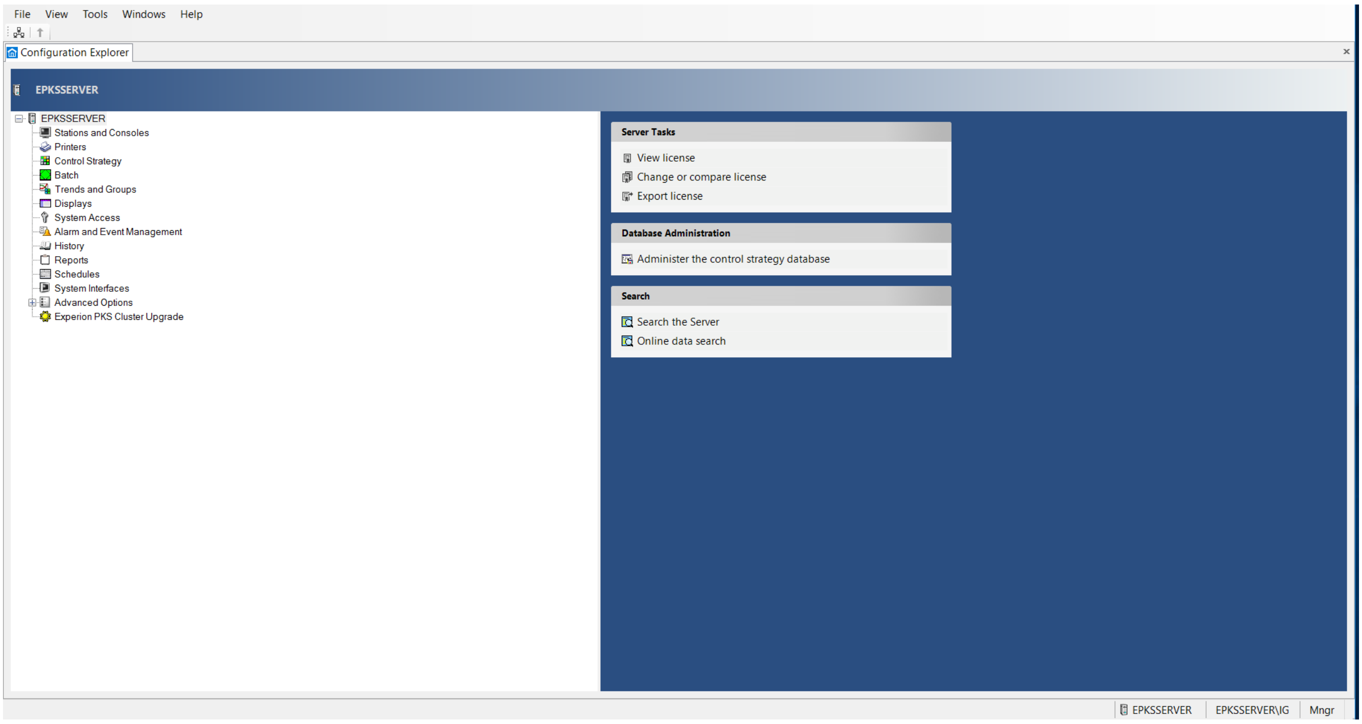
- Collects data from your system and displays it in a manner that you can easily understand
- Allows you to control your system by sending appropriate commands
- Automatically performs scheduled tasks
- Notifies you of system activities, including alarms and system events
- Generates comprehensive reports

2.5 Experion PKS Knowledge Builder

Knowledge Builder is the online documentation library for the Experion PKS system. It is provided on a compact disc and will be installed by default on all machines configured with an Experion PKS system.

2.6 Configuration Studio

Configuration studio provides a central location from which you can configure Experion PKS system. The individual tools required to configure parts of the system are launched from configuration studio. Configuration studio provided with customized list of tasks that are required to complete the configuration of the system. When you click the task the appropriate tool is launched so that you can complete the task.



Experion PKS, Process Control System is being supplied to HXXXX. The current software version of this System is ***R501*** that runs under Microsoft Windows10 operating system. The system contains the redundant servers.

Experion PKS is a control system that utilizes the **Experion C300 Controller** and also includes a full library of interfaces to programmable 3rd party controllers that is supported by Experion SCADA.

Both Experion Servers consist of the following main components:

Control Builder- a graphical, object-oriented tool that supports Experion C300 Controller system design, documentation, and monitoring.

HMI Web Display Builder- a tool, which is used to create custom HMI Web displays using Web-based features and it saves the displays in HTML format.

Quick Builder- to create and modify configuration databases, which define how system items, such as Stations, printers and points, are set up. In our project, quick builder is used for the station and printer setup.

Knowledge Builder is an electronic on-line Documentation using HTML Frame Browser that provides users quick access to system information.

Server scripting

You can extend the functionality of Experion by creating server scripts. As you configure your system, you may want to create server scripts for:

- Servers
- Points
- Reports

The type of script you want to create determines the tool you use. For example, if you want to create a script that runs at specified times, you use Station to create the script. If you want to create a script that performs a task when the status of a point changes, you use Quick Builder to create the script. For more information about server scripting, see the Server Scripting Reference.

2.7 Configuration procedures

The table below shows the tasks for configuring Experion in their recommended sequence. The table also shows:

- Which procedures relate to licensable options.
- Which procedures are optional (Optional procedure).
- Where to find the information, you need to complete the procedures.

Task	Licensable Option	Tool
Add operators to Experion so that configuration engineers can log on to Configuration Studio.	No	Station
Build your system model and asset model	No	Configuration Studio - Enterprise Model Builder
Build your Network tree	No	Configuration Studio
Configure performance monitoring	Yes	Configuration Studio
Build Flex Station	Yes	Configuration Studio - Quick Builder
Build printers	No	Configuration Studio - Quick Builder
Define Station setup details, and complete Station and printer configuration	No	Station
Configure server-wide Station settings	No	Configuration Studio
Define point servers to the server	No	Configuration Studio
Build flexible points and, if required, create server scripts	Yes (Flexible points)	Appropriate point server tool Configuration Studio - Quick Builder Station
Build your control strategy		
Build a control strategy for Process Controllers	Yes	Configuration Studio - Control Builder

Task	Licensable Option	Tool
Build equipment templates	No	Quick Builder
Build equipment	Yes (Equipment points)	Quick Builder
Build SCADA controllers and the communications paths from the controllers to the server	Yes	Configuration Studio - Quick Builder
Test communications between the server and the controllers	No	Controller-specific test utilities
(Optional procedure) Build Electronic Flow Measurement (EFM) meters, data export formats, and schedules	Yes	Configuration Studio - Quick Builder
Build SCADA points (analog, status, and accumulator) on controllers and, if required, create server scripts	Yes	Configuration Studio - Quick Builder
Configure history collection	No	Configuration Studio
Configure system security	No	Configuration Studio - Quick Builder
Configure a redundant server system	Yes	Configuration Studio
Define the servers that will form a Distributed System Architecture (DSA)	Yes	Configuration Studio - Enterprise Model Builder
Configure Console Station	Yes	Configuration Studio
Configure eServer	Yes	Station
Configure Mobile Station	Yes	Station
(Optional procedure) Configure dashboard hierarchy	No	Configuration Studio
(Optional procedure) Configure groups and trends	No	Configuration Studio
(Optional procedure) Configure pre-formatted reports and create server scripts to run on completion of reports	No	Configuration Studio
Configure Point Control Schedules	Yes	Configuration Studio
Configure Recipes	Yes	Configuration Studio
Configure Event Archiving	No	Configuration Studio

Task	Licensable Option	Tool
(Optional procedure) Configure Microsoft Excel Data Exchange and the ODBC options. Develop applications that use the Application Programming Interface (API) and Network API.	Yes	Microsoft application development tools
If required, create server scripts	No	Configuration Studio
(Optional procedure) Build custom displays and, if required, create display scripts	No	Configuration Studio
Configure Collaboration Station Workspaces	Yes	Collaboration Station
Perform backups	Yes	Experion Backup and Restore

Configuration procedures checklist

2.8 The Experion standard features

Alarm and event management that provides the operator with comprehensive alarm and event detection, management, and reporting facilities.

History Collection that is available over a wide range of frequencies in both average and snapshot/production formats. A large amount of history can be retained on line, with automatic archiving, retention and access to unlimited quantities of historical data.

Trending allows viewing of historical point data.

Reporting can be configured in many different types and generated either by demand or periodically.

Security that provides configurable security levels, control levels and area assignments. These may be configured for each individual operator or alternatively for each operator station.

To allow the different Experion Servers to communicate with each other, and allow data sharing, it utilizes Honeywell state-of-the-art **Distributed Server Architecture (DSA),** will be used to integrate multiple servers as a single system.

Server Redundancy

Following Redundant Experion Servers with FTE are being provided to HXXXX. The redundant server configuration will provide a warm fail over architecture with on-line database replication. The Primary and backup Server will communicate with each other over FTE.

The primary server will update the backup server with all pertinent information- the server and database information, so if there is a failover, the backup will take over in the primary's place. The Primary's (Experion) Server service must be able to communicate with the Backup's Server Service.

2.9 Diagnostic Software

The System Status display is similar to the Alarm Summary, however the System Status display shows system alarms for system and network components only.

You can use the System Status display to:

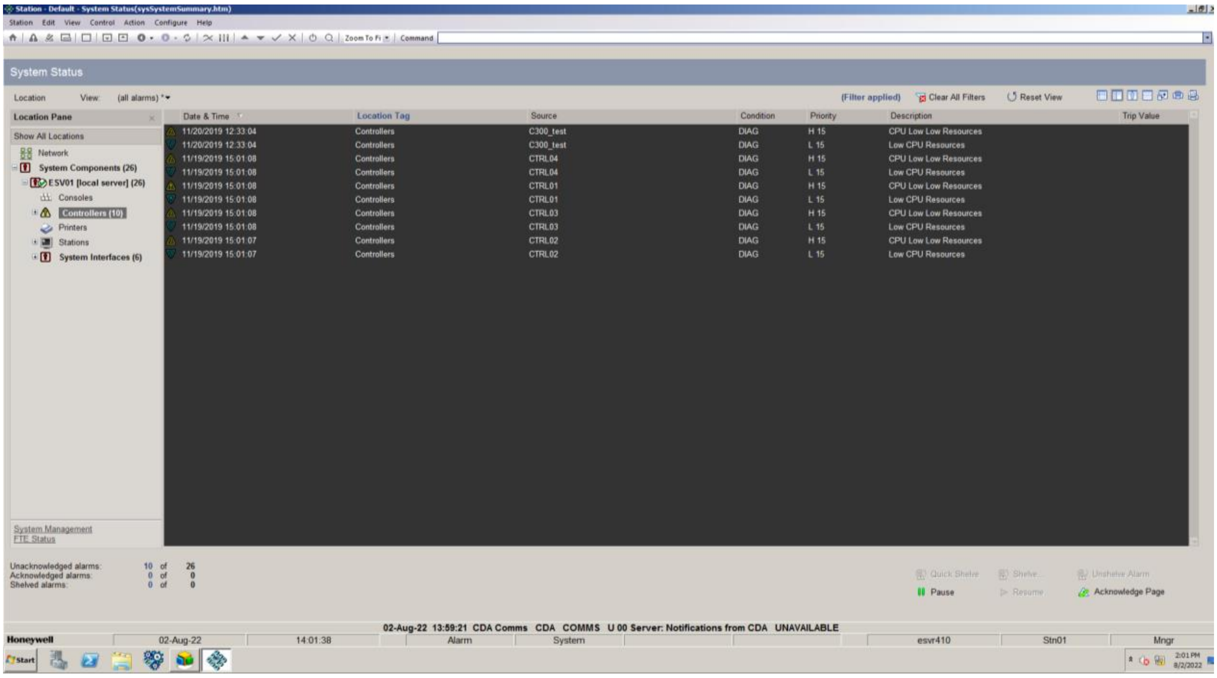
- Check the status of components to verify that they have been configured correctly and are operating correctly.
- Respond to system alarms such as:
  - Communications failures
  - Station failures
  - Operator logon failures
  - Printer failures.
- Monitor the status of the system to prevent problems occurring.
- Diagnose problems in the system.
- Review firmware versions for C300 controllers.

2.9.1 Calling up the System Status display

There are several ways to call up the System Status display:

- Using the Station menu
- Using the Station Status Bar, usually in response to a system alarm

To call up the System Status display using the Station menu In Station choose View > System Status Display.
To call up the System Status display using the Status bar In Station, click the System box in the Status bar. If there is a system alarm, the System box is flashing.



2.10 The Experion hardware node configurations

Enterprise Model

An enterprise model provides a means of organizing your system around key entities in your enterprise, such as plant equipment. An enterprise model provides: A hierarchical structure that makes it easier for users to navigate their way through your system. A simple and intuitive means of implementing scope of responsibility—that is, systematically managing the access rights of operators (or Stations) to various parts of your system. An enterprise model is a framework that includes a set of specialized models, such as the asset and system models, each of which represents one aspect of your system.

Configuration Studio->[SystemName]->System Task->Configure Assets…

Asset naming : A_ [xx]

Where, [xx] : Unit code

- Asset Tag bName : Same as Item Name
- Asset Item Name :

The assets of this project will be defined as below:

Asset Tag Name	Asset Item Name	Description
A_12	A_12	

The server name in EMDB (Enterprise Model Database) will be defined as below：

- Configuration Studio Server naming：CCRSVR

Station

Configuration Studio->SERVERS->[SERVERNAME]->Stations and Consoles

Quick Builder：

- Name：[uu]STN[NN]

[uu]：WHERE XX: UNIT CODE

[NN]：Station no.

Printer

Configuration Studio->SERVERS->[SERVERNAME]->Printers

- Name：\\[HOST Name]\PRN[NN]

[HOST Name]：Will be the station computer name if the printer is local printer of a station. Will be network printer path name or IP address if the printer is a network printer.

[NN]：Printer no.

Note: The printer name must be same as the network sharing name in windows.

SCADA Channel/Controller/Point definition

Quick Builder is a tool to create and modify configuration databases, which define how system items, such as controllers, point and Operation Stations are set up. It is used for configuring third party device controllers, points etc. After defining hardware and points with Quick Builder, one has to download these definitions from Quick Builder to the Experion PKS ES database.
Only Modbus RTU interface was used in this project.

2.11 The Network tree

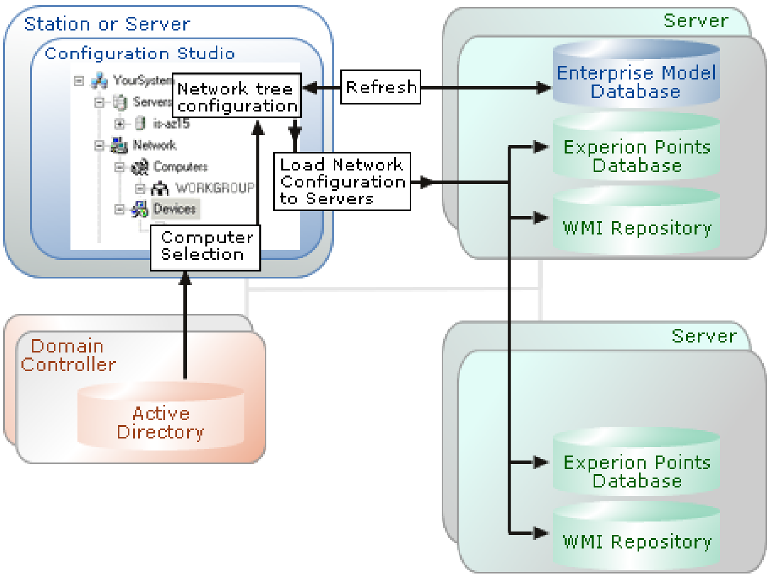
The Network tree is a graphical view of the nodes in your network, which can be viewed on the System Status display. (The System Status display also contains the System Components tree, which is automatically configured.) You configure a Network tree so that you can view the status and health of network items from the System Status Display. From the network tree in the System Status display, you can then access detail displays for the network tree items that display performance and configuration data. In the Network tree, you can configure the following to appear:

- Computers that host applications or software such as ACE, Station or Server.
- Collections of computers such as workgroups, domains, and Organizational Units.
- Devices such as switches and routers.
- Collections of devices and computers called FTE Communities.

The Network tree works in conjunction with the System Event Server and the System Performance Server to display system errors on the System Status display. Operators and engineers can use this information to

diagnose or troubleshoot faults within the system. Details about the items in your Network tree, such as computers, devices, and FTE Communities, are stored in system databases. The initial definition of the tree (using the Computer Add/Remove task) displays the latest view of computers, workgroups, and domains from the Active Directory. The Network tree configuration, as displayed in the Configuration Explorer, resides in the Enterprise Model database (EMDB).

When you add (or remove) Network tree items, you must also load the configuration to the servers in the system so that the tree items can be stored as points in the Experion point database, in addition, switch and FTE Community information is stored in the WMI Repository. The following figure shows the relationship of user configuration tasks to the databases you update or load so that you can support your Network tree configuration.



If there is an inconsistency of the Network tree configuration between the EMDB and the Experion database, icons appear in the Network tree in Configuration Studio.

The Honeywell System Performance Server provides performance and configuration information that is internally maintained on a per-node basis using Microsoft's WMI (Windows Management Instrumentation). This data is accessible as OPC data thus allowing it to be easily integrated into operator displays and process applications in a manner consistent with process data access.

2.11.1 Network tree configuration tasks

The following topics describe how to manage the Network tree on the System Status display.

Prerequisites

- Before you can load your Network tree configuration, you must first have added servers to your system.

- To view Network tree items from remote Experion servers, DSA must be enabled between the servers.

When you first define a server and download this server definition to other servers in the system, the default subscription for the server is disabled and it does not participate in the DSA.

Prerequisites

• The computer must have system management software installed. This is installed by default on all Experions, however, for non-Experion nodes you will need to install this software manually.

- The computer has the correct multicast address.
- One server node hosting SES per FTE community has been determined to be the SES for the FTE community.

To add a computer to the Network tree

1. In the Configuration Explorer in Configuration Studio, expand the System tree you are configuring and click Networks.

2 .Click the Add/Remove Computers task.

The Computer Selection dialog box opens.

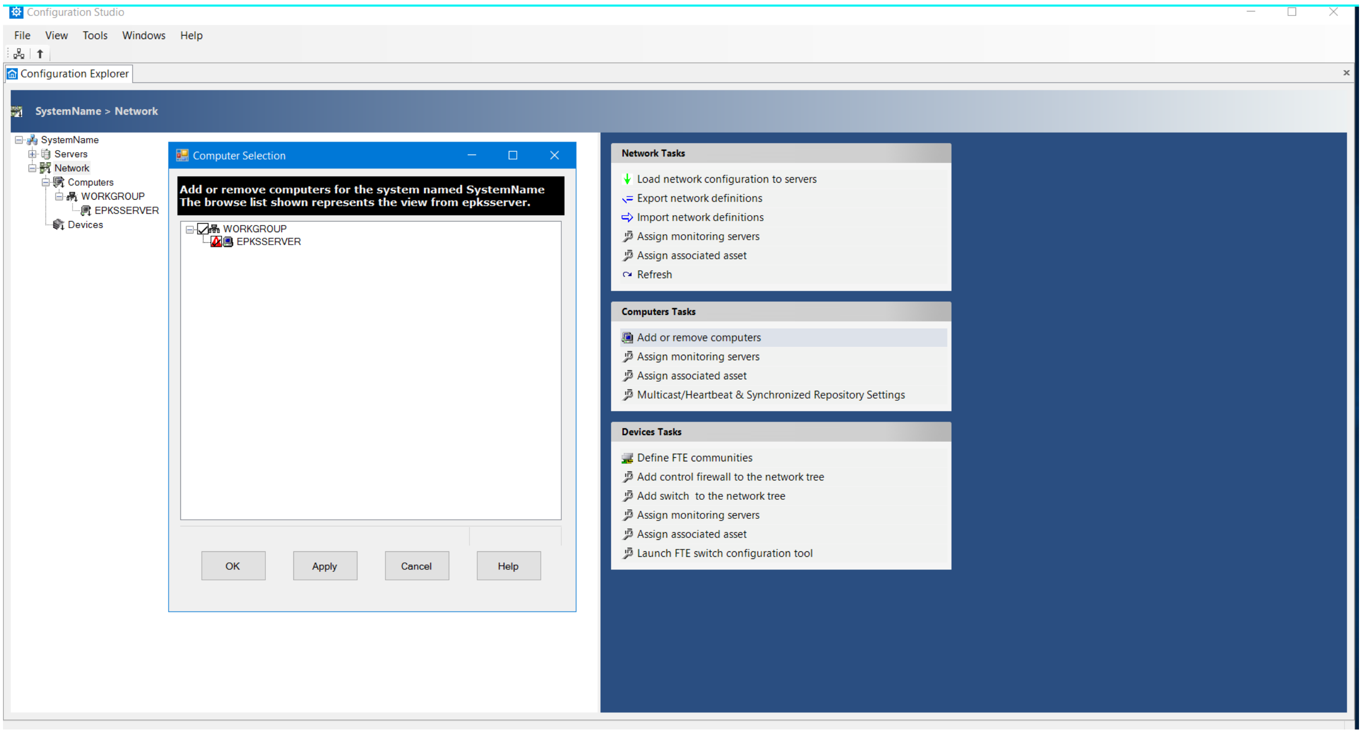
3 .Select the check box for each computer you want to add to the Network tree and click Apply.

The computers are added to the Network tree in Configuration Studio.

4 .Click the Load Network Configuration to Servers task.

5 .Select the servers you want to download to and click OK.

You should download the network tree to all servers in the system.



Adding FTE communities to the Network tree

Consider the following when you add FTE communities:

- Nodes planned for the FTE community have been identified.
- One server node hosting the System Event Server (SES) per FTE community has been determined to be the SES for the FTE community.
- The FTE community name you specify should be meaningful to your operators, for example, the name can identify the grouping of FTE devices.

Prerequisites

- Multicast address for the community has been defined using Installation Builder.
- Nodes in the FTE community have been configured to use the same multicast address using Installation Builder.

To add an FTE community to the Network tree

1. In the Configuration Explorer in Configuration Studio, expand the System tree you are configuring and click Devices.

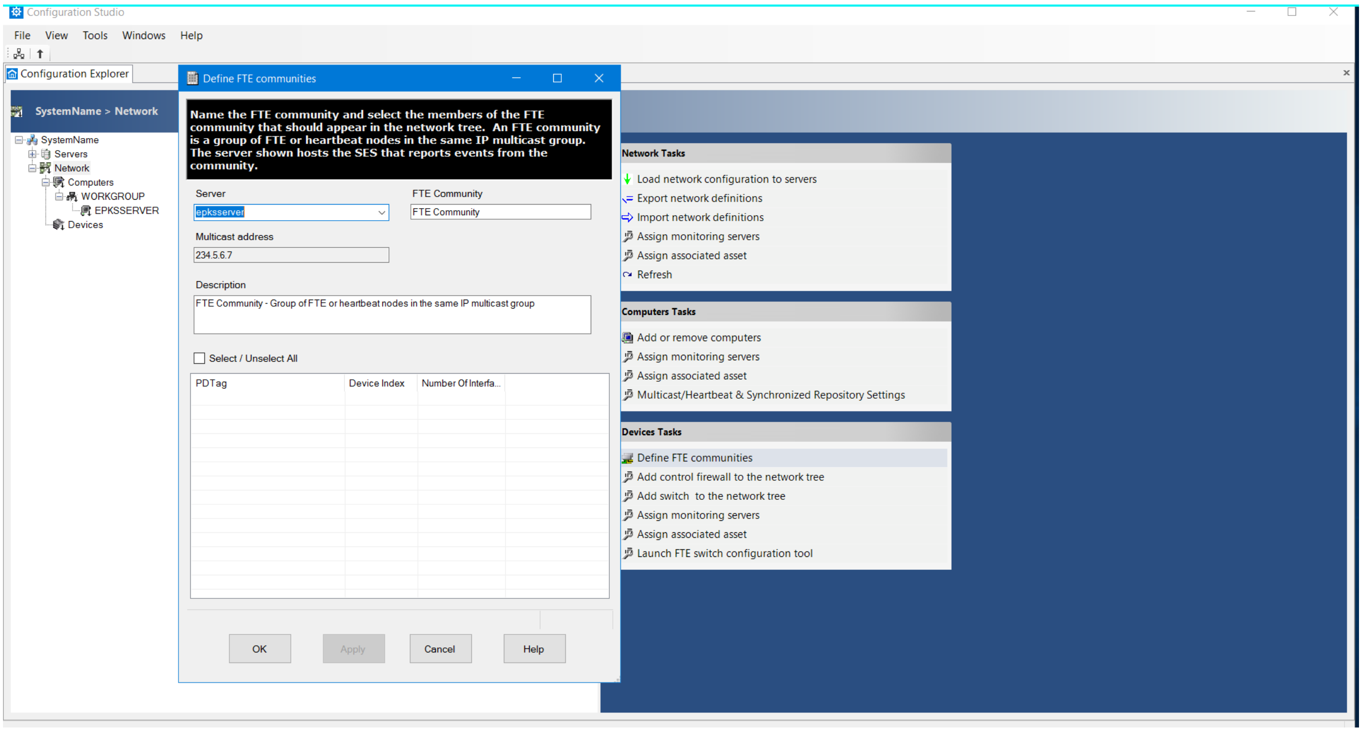
2. Click the Define FTE communities task. The FTE Definition display appears.

3. Type the FTE community name and click Apply. The FTE community appears in the devices tree.

4. Select the server hosting the SES for the FTE community. The System Event Server appears with its multicast address defined.

5. Select the nodes for the FTE community.

6. Click Apply.



Multicast settings

The multicast address is initially configured using Installation Builder.

Adding a switch to the Network tree

For more information about changing switch or router settings, see the Fault Tolerant Ethernet Overview and Implementation Guide.

Prerequisites

- The switch has been configured. To add a switch to the Network tree
    1. In the Configuration Explorer in Configuration Studio, select the system you want to configure.

The Network tree appears underneath the system item.

2 .Expand the Network tree.

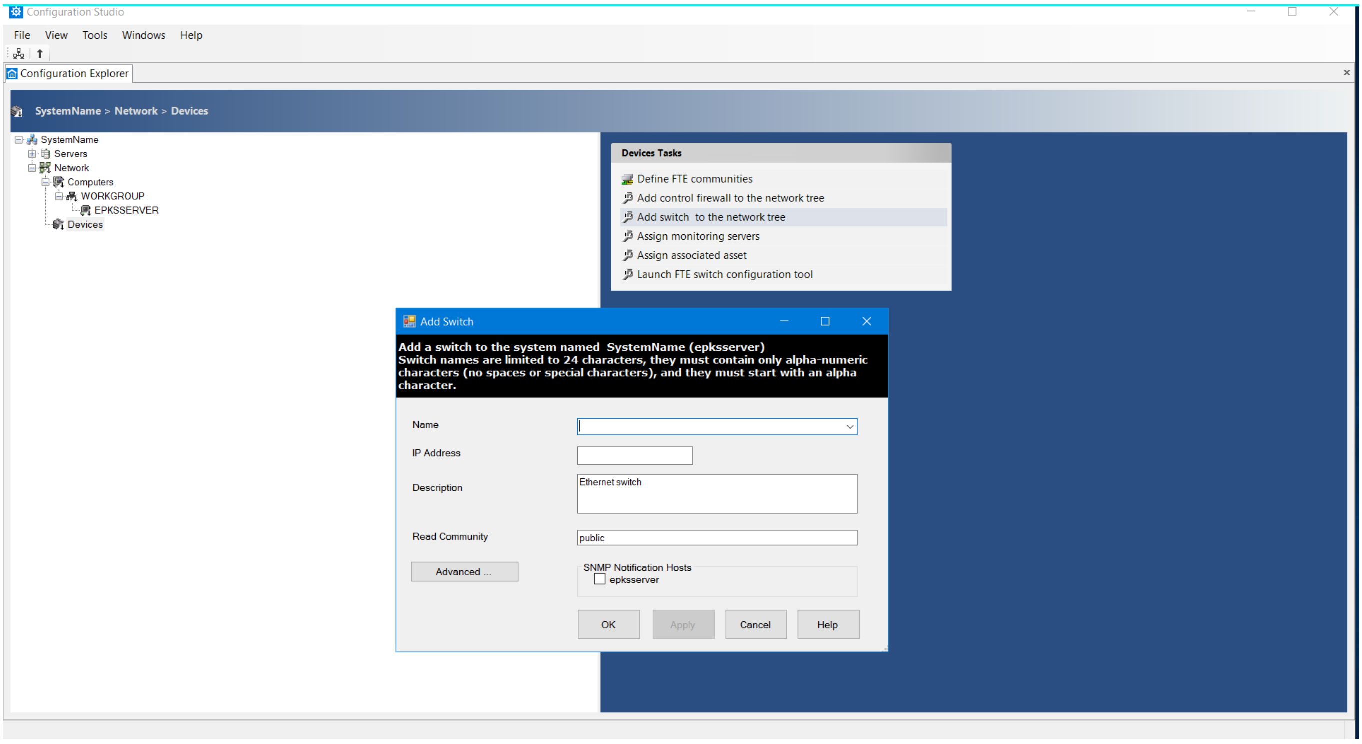
3. Select the Devices item.

4. Click the Add Switch to the Network tree task.

5. Type the switch name, which is to be displayed in the Network tree, and its IP address. The switch name

must contain only standard alphanumeric characters; no spaces are allowed. The switch name is also used to create a WMI namespace used for accessing switch data.

6. Click Apply.



Assigning associated assets to Network tree items

You use Configuration Studio to assign associated assets to Network tree items.

Prerequisites

- You have logged on to Configuration Studio using an account with engineering privileges, and you have connected to a system.

If you use an account that only has operator privileges, you can view configuration details but you cannot assign associated assets.

To assign an associated asset to a Network tree item on the network

1 .In the Configuration Explorer of Configuration Studio, click **Network**.

2 .On the Computer Tasks panel or the Devices Tasks panel, click the Assign associated asset task.

The **Assign associated asset** dialog box appears.

3. In the Item list, select the Network tree item to which you want to assign an associated asset, and then click **Select Asset**. The Asset list dialog box appears.

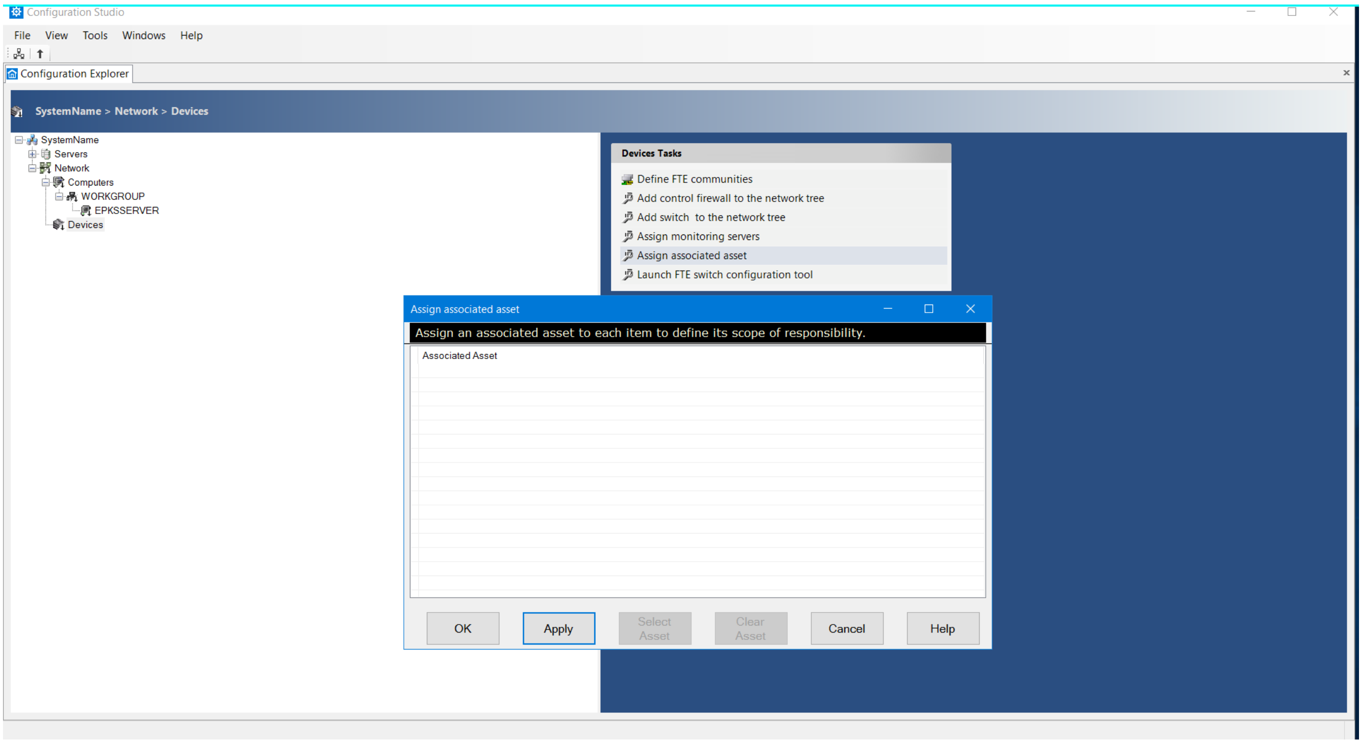
4 .Select an asset from the list, and then click **OK**.

The Asset list dialog box closes and the associated asset is assigned to the selected Network tree item.

5. Repeat as necessary for other Network tree items in the list.

6. When you are finished, click **OK**.

7. When you are ready to load your changes to system servers, click Load network configuration to servers on the Network Tasks panel.



To remove an associated asset from a computer on the network

1. In the Configuration Explorer of Configuration Studio, click Network.

2. On the Computer Tasks panel or the Devices Tasks panel, click the Assign associated asset task.

The Assign associated asset dialog box appears.

3 .In the Item list, select the Network tree item from which you want to remove the associated asset, and then click Clear Asset.

The associated asset is removed from the selected Network tree item.

4. Repeat as necessary for other Network tree items in the list.

5. When you are finished, click OK.

6. When you are ready to load your changes to system servers, click Load network configuration to servers

on the Network Tasks panel.

Loading the Network tree configuration to servers

You load the Network tree configuration after you have initially configured your Network tree or after you have made adjustments to the Network tree, such as deleting items. An icon appears next to the Network tree item in Configuration Studio to indicate that a change has been made and you need to load the configuration. To load the Network tree configuration

1. In the Configuration Explorer in Configuration Studio, select the system you want to configure.

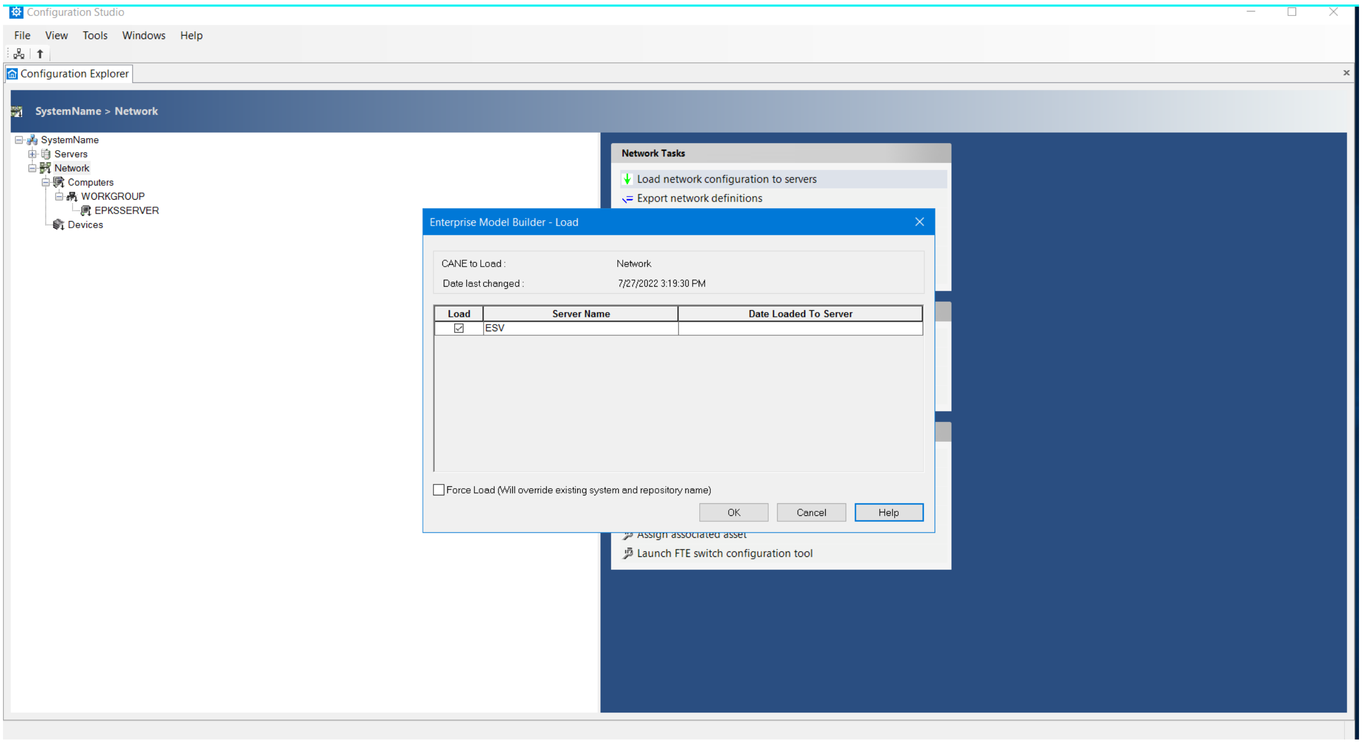
The Network tree appears underneath the system item.

2. Click on the Network item in the tree.

3. Under Network Tasks, click the Load Network Configuration to servers.

4. In the Load column, select the check boxes of the servers that you want to download the Network tree to.

5. Click OK to begin the Network tree load to the selected servers.



Removing items from the Network tree

To remove an item from the Network tree

1. In the Configuration Explorer in Configuration Studio, select the system you want to configure. The Network tree appears underneath the system item.

2. Expand the Network tree and select the item you want to delete.

3. Click the **Remove this item from the Network tree** task.

4. Click **Yes** to confirm you want to remove the item.

5. Click the **Load Network Configuration to servers task** to download this change.

6. Select the servers you want to download to and click **OK**. **Network tree maintenance** If you have an existing item in the Network tree and you change details about this item, for example, you rename a computer or reassign an item to another Windows domain or Organizational Unit, you need to re-add the item to the Network tree.

Configuring System Event Server and System Management Multicasts

System Management Multicasts allow multiple System Managements to be configured on the same FTE Multicast so they operate exclusive to each other. It is recommended that you configure separate System Management Multicast addresses for each cluster server so that an SES can be installed on each server.

By having this configuration you can select an appropriate SES alarm scope for your system. The supported SES alarm scopes are:

- Cluster-contained SES alarms

SES alarms only circulate within the cluster. The cluster is defined as the single server pair or pair of redundant servers, Console Stations, Flex Station computers and other non-Experion nodes connected to the server(s). Alarms from one cluster will not be visible from another.

- System-wide SES alarms SES alarms will circulate throughout the system.

Once the System Management and SES have been installed and configured you are able to change the alarm scope. To configure System Event Server and System Management Multicasts, you need to complete the tasks listed below.

Adding Computer Nodes to the System Management Display Page

When you open the System Management Display on any server, by default no machines are listed. You need to add all the computers within the system to your tree.

To add computers to the System Management Display

1. Right click on the domain or workgroup node situated below **Node Administration**.

2. Click **Properties**. This should bring up the properties window as shown below.

3. Check the **Display Computers** checkbox situated towards the bottom of the window. Doing this will display a list of computers that you can add to your tree.

4. Select the machines that are in your system.

5. Click OK.

Displaying Multicast/Synchronization Configuration Pages

By default the Multicast Configuration Pages are hidden from view. You need to enable them. To display the Multicast/Synchronization Pages

1. Right click Node Administration.

2. Click Properties.

3. On the Startup Behaviors tab, select the Display Multicast/Synchronization configuration pages checkbox.

4. Click **OK**. Next time you open the properties window, there will be two more tabs available for configuration.

Clearing all SES events and shutting SES down

Before configuring different System Management Multicast addresses you will need to clear all SES alarms and shut down the SES. This will prevent orphaned events from being left from previous scope.

To clear all SES events and shut down SES

1. Open up System Management Display.

2. Right click the domain/workgroup node and select Display Events. This opens an Event Summary display.

3. To clear all SES events, you will need to Acknowledge and/or Clear the selected events.

4. To stop the SES servers, expand each computer node and right click on the SES node.

5. Go to All Tasks and select Stop. This will stop the SES.

6. Repeat for all machines that have an SES installed.

Configuring different System Management Multicasts for each cluster

In order to have an SES properly configured on each cluster server, it is important to ensure that each cluster has its own System Management Multicast. Doing this ensures that each SES cannot communicate with each other, preventing duplicate SES alarms circulating the system. To configure a different System Management Multicast for each cluster :

1. Right click Node Administration.

2. Click Properties.

3. Select the Synchronized Repository Settings tab.

4. In the Settings Source box, select one of the machines.

5. Enable Override FTE / Heartbeat Multicast Address. Enabling this will **make IP Multicast Address**

editable. The default IP Multicast Address is 1 above the FTE Multicast address, thus if the FTE Multicast address is 224.0.0.107 then the default System Management Multicast address would be 224.0.0.108 as shown in the diagram below.

6. In the IP Multicast Address box, enter a unique IP address.

It is recommended that the first System Management Multicast Address be 2 above the FTE Multicast

Address and increment each subsequent System Management Multicast Address by 1. It is not

recommended to use the default System Management Multicast address (1 above the FTE Multicast address) as new servers with SES will use it.

7. Click the **Apply** button. The Save Configuration window will display.

8. By default, all nodes/machines will be selected. Deselect all the nodes and then reselect the machines in the cluster that you want within the System Management scope; thus you should select the servers, all connected console stations and any other non-Experion nodes that have direct connection with the server/console station. Failing to deselect other nodes will download the configured multicast address to those machines and overwrite their existing multicast addresses.

9. As each cluster should have their own System Management Multicast address, repeat the instructions above for every cluster.

Configuring the system for cluster-contained alarms

Once you have configured the system so that each cluster has its own System Management Multicast address, it is very easy to set your system up for Cluster Contained SES Alarms so that SES alarms only circulate within the cluster

To configure the system for cluster-contained alarms

1. Click Configure > System Hardware > Distributed Servers.

2. Click on the first distributed server (system number 2 onwards).

3. In the Configuration tab, in the Alarms and Data Subscription box select either Enable Data Only or Disable Both Alarm and Data.

4. Repeat the above steps for all distributed servers.

Configuring the system for system-wide SES alarms

To enable System Wide Alarms you will need to configure distributed servers to send and receive both alarms and process data.

To configure the system for system-wide alarms

1. Click on **Configure** > **System Hardware** > **Distributed Servers**.

2. Click on the first distributed server (system number 2 onwards).

3. In the Configuration tab, in the **Alarms and Data Subscription** combo box select **Enable Both Alarm and Data**.

4. Check that the Network tree has been enabled in the **Asset Enable/Disable** display from **Configure** > **Alarm Event Management** > **Alarm Enable/Disable**.

5. Do this for all distributed servers.

Configuring Redirection Manager for different Server Management Multicast addresses

When a Redirection Connection is configured through System Management Display, HCI Name Service Provider adds values to the local registry. It also replicates the values added to the registry to the other clusters so the Redirection Connection will exist on external clusters.

When each cluster has different System Management Multicast addresses, HCI Name Service Provider is unable to replicate the registry values outside of the multicast. This means that when you configure a Redirection Connection the values will only be added to the local registry and not remote ones, meaning it will only work within the cluster on which you configured it.

To configure Redirection Manager for different Server Management Multicast addresses

- Configure the same Redirection connection on each of the external clusters. Therefore, if you have three clusters within your system and you configure a Redirection Connection on cluster A, you will need to configure the same Redirection Connection on clusters B and C.

Configuring GUS Displays for different System Management Multicast addresses

Another area that does not get replicated properly is Aliases. Each machine in the system has a set of aliases to represent different components on that machine. When a new component is added to the machine, it may add more aliases to the existing alias list however different System Management Multicast addresses prevents the new aliases from being replicated across the multicasts.

As GUS displays relies on aliases this issue may affect the behavior of GUS display operation. Alias Generator enables you to select a machine on the network and add any of the aliases from that machine to your local machine.

To configure GUS displays for different System Management Multicast addresses

- Use Alias Generator to add the aliases from remote machines onto your local machine. For instructions on Alias Generator, see the topic titled "Generating a File using Alias Generator" in the System Management Configuration Guide.

Effects of different System Management Multicast addresses on HCI manager

HCI Name Service Provider scope works on the System Management Multicast address. Therefore when two Name Service Providers are on the same multicast address, they can communicate with each other, when they are on different multicast addresses they cannot.

Configuring the system so that each cluster has its own System Management Multicast address limits the scope of HCI Name Service Provider to cluster only. This affects Redirection Manager configurations from being replicated across clusters and will also affect GUS displays from operating properly.

About notifications and event filters

When you install the System Event Server, a default filter is created that exposes only FTE events. You can create filters for other Windows events or share an existing filter. Event filters define a subset of Windows events that are collected by the SES. These events may be generated by the Windows system, Honeywell applications, or third-party applications.

So that the notifications collected by the SES are consistent with notifications in Experion, you need to specify more information than is initially generated for each Windows event. When you create an event filter, you specify extra information such as:

- Event type
- Event source
- Severity
- Category
- Condition and sub condition
- Active/acknowledge state
- Attributes

Changing the System Performance Server scope

This is an optional task. No user configuration is required on the System Performance Server itself unless you need to change default settings described in this section. The scope of the System Performance Server is created automatically when you configure the Network Tree in configuration studio.

In general the default scope of domain is sufficient. If errors are being generated in the event logs because of access problems to specific nodes, change the scope to exclude the problematic nodes. Changing scope limits OPC browsing of third-party OPC clients, however, it does affect access to data items.

To change the scope of the System Performance Server

1. In the Configuration Explorer of Configuration Studio, click **Network** to expand the Network tree.

2. Click the computer hosting the System Performance Server.

3. Click the Configure HCI Component tasks.

The HCI Component display appears.

4. Select the component named SPS computer name where computer name is the name of the computer where the System Performance Server is installed.

5. Click Enter/Edit Server Specific configuration.

The SPS Server Configuration display appears.

6. Click Add Static Computer nodes to list all nodes.

7. Select or clear the Check boxes as appropriate to limit the SPS scope to any combination of OUs and computers.

8. Select the nodes you want to monitor.

9. Click OK.

Adding system performance data to displays

After you have checked the status of the System Performance Server, you can add system data to your custom displays.

To create a custom display and to display system data

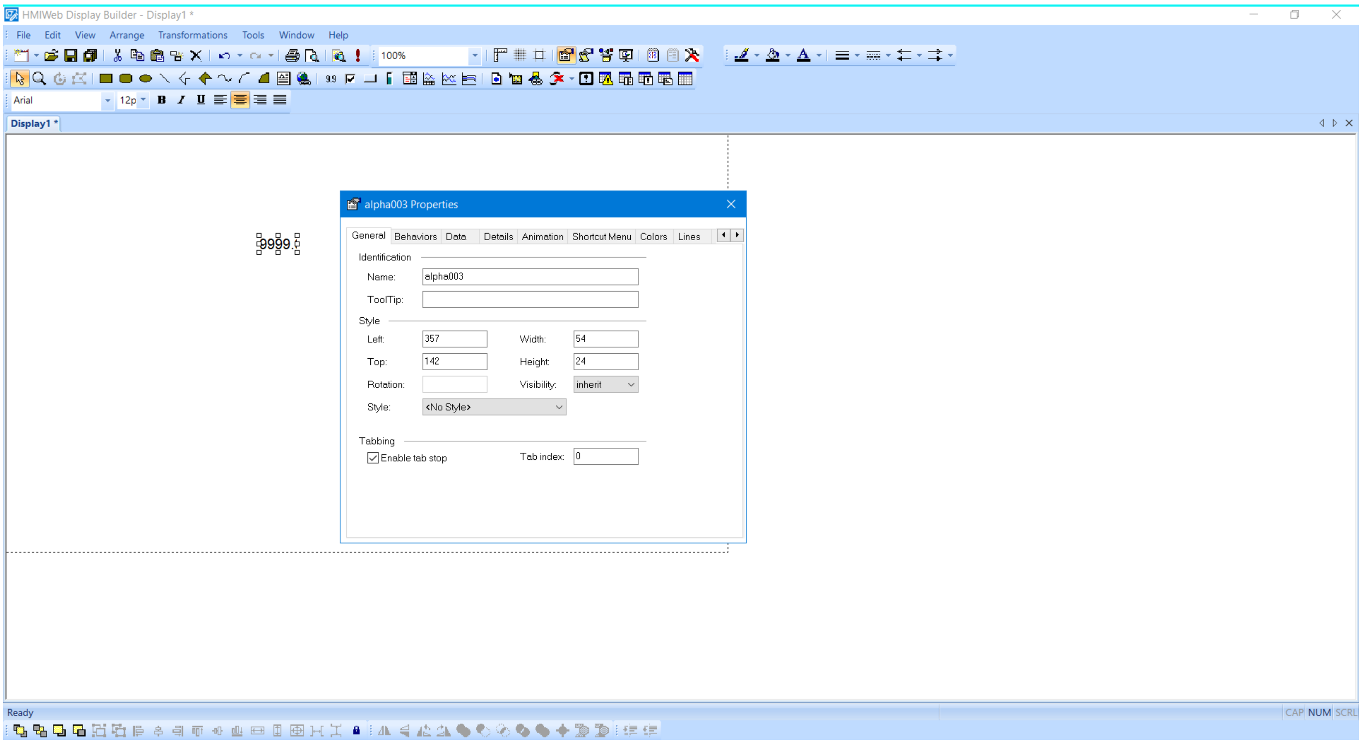
1. Start HMIWeb Display Builder.

2. Choose File > New > Display.

A blank display appears.

3. Click the (Alphanumeric) icon on the Toolbox toolbar and then drag the pointer on the display to define the size, shape and position of an alphanumeric.

4. Double-click the alphanumeric to open the Properties Window, which shows the alphanumeric's currentproperties.



5. Click the Data tab and make sure that Point/Parameter is selected in **Type of database link**.

6. Select SYSMGT from the point list.

7. In the **Parameter** box, type the alias for the data item you want to display in the format computer name alias. For example, localhost.BiosVersion.

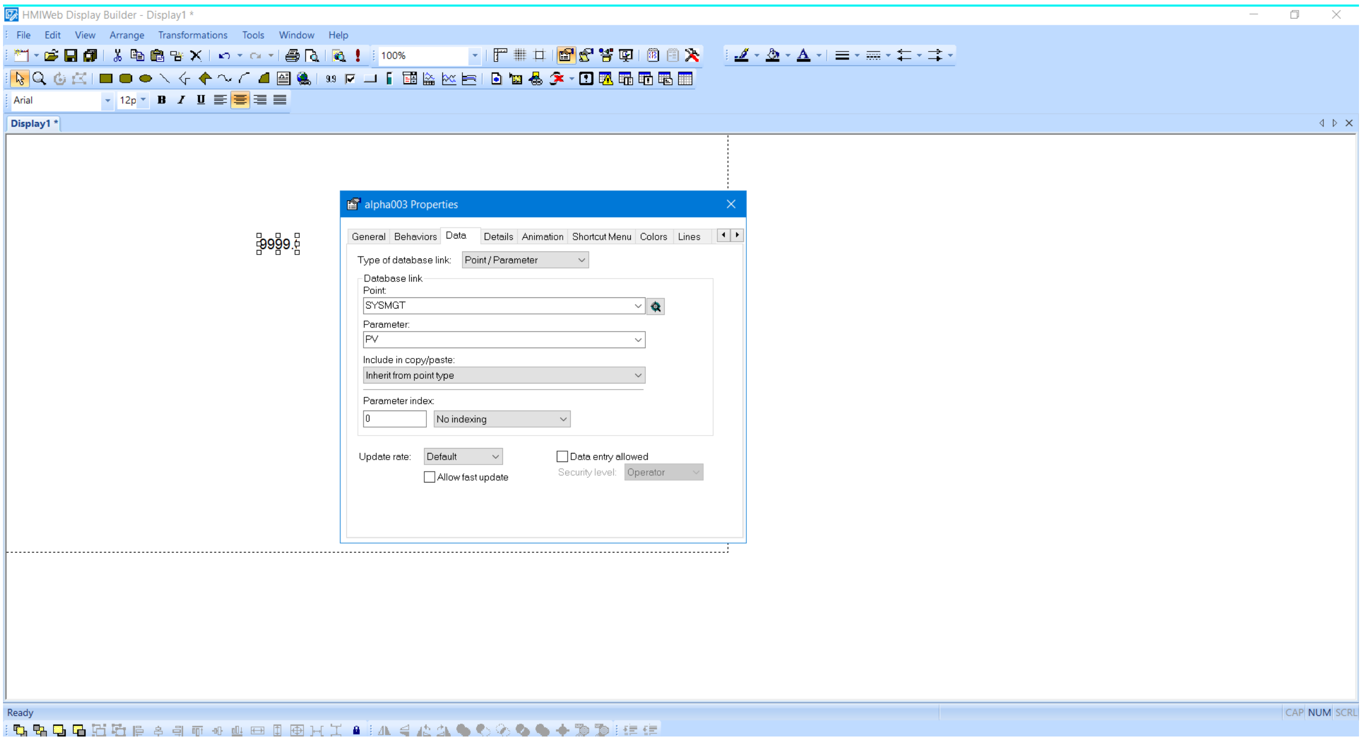
8. To find the alias names defined on the System Performance Server (SPS), do the following:

a. Open the Honeywell System Management Display tool.

b. Locate the SPS in the domain and organizational unit tree.

c. Right-click on the SPS name and choose **Configure** from the shortcut menu.

d. Click the **Aliases** tab.



Setting up the server as a trusted site on the Station computer

If your Station computer does not reside in the same domain as the Experion server to which you are connecting, you need to define the Experion server as a trusted site in Internet Explorer.

To make the server a trusted site

1. Open Internet Explorer.

2. Choose Tools > Internet Options.

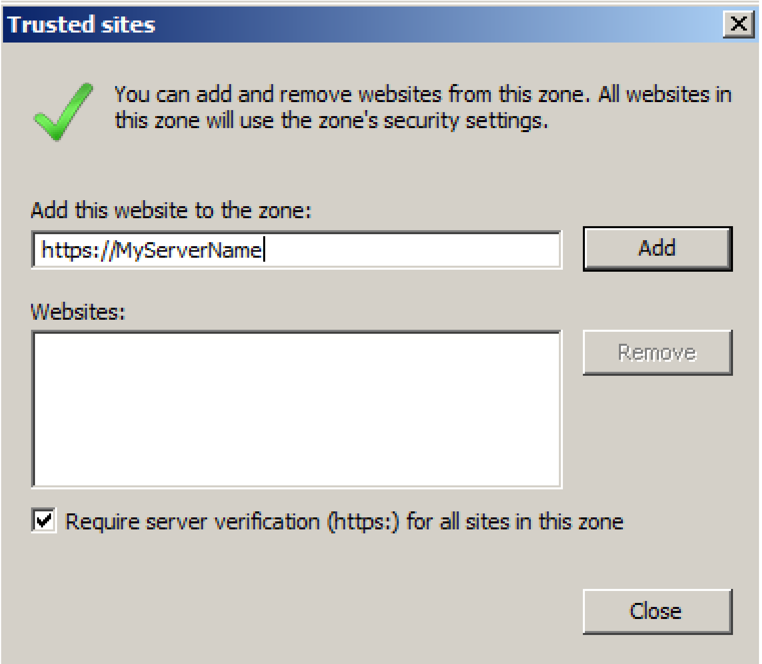
The Internet Options dialog box opens.

3. Click the Security tab.

4. Click the Trusted Sites icon.

5. Click Sites.

The Trusted Sites dialog box opens.



Internet Explorer Trusted Sites dialog box

6. In the Add this Web site to the zone text box, type in http://servername (where servername is the name of the Experion Server) and click Add. This adds the Experion servername to the Web sites list.

If you have redundant servers, type http://servernamea, where servernamea is the name of the primary

Experion server and type http://servernameb, where servernameb is the name of the backup Experion

server.

7. Clear the Require server verification check box.

8. In the Add this Web site to the zone text box, type in the IP address of the Experion Server and click Add.

This adds the IP address to the Web sites list.

9. Click **OK** to close the second dialog box and click OK again to save your changes.

Connecting to the server for the first time

Prerequisites

- For the new Flex Station to integrate successfully with the Alarm Tracker you need to synchronize the date and time on the Flex Station with the date and time on the server, as documented in the section, 'Setting up date and time synchronization' in the Supplementary Installation Tasks Guide.

To connect to the server

1. From Station, choose Station > Connect. The Connect dialog box appears.

2. Click Edit Connection. The Connection Properties dialog box appears.

3. Click either Static Station or Rotary Station, depending on the connection type you require.

4. From the Server list, choose the name of the server you are connecting to.

If the server does not appear in the list, type the server name.

5. If you chose a Static Station in step 3, in the Station number box, type the Station number for this Station. The Station number is the Item Number that was assigned to the Station in Quick Builder.

6. Click Save.

The connection details you have specified are saved in the Station setup file (.stn file).

7. Click Connect.

2.12 About Station

Station is Experion's user interface. You use Station to monitor and control your system. Station displays are used to notify operators of alarm conditions. Station works in conjunction with Experion server, which:

- Collects data from your system and displays it in a manner that you can easily understand
- Allows you to control your system by sending appropriate commands
- Automatically performs scheduled tasks
- Notifies you of system activities, including alarms and system events
- Generates comprehensive reports

What type of Station are you configuring?

There are several different types of Station and several different environments in which Station operates. The two main types of Station are Flex Station and Console Station.

A Flex Station is connected to an Experion server and is the interface an operator uses to interact with your plant, as described above. A Flex Station is generally installed on a computer other than the Experion server and can use either the rotary or static connection type.

A Console Station provides all of the functionality available on a Flex Station, but also has direct access to process controllers and ACE nodes. This enables an operator to monitor and control the plant regardless of the state of the Experion server or redundant servers. A Console Station also differs from a Flex Station in the following ways:

- It has its own connection type
- It is not built using Configuration Studio
- It is installed differently from a Flex Station

In this chapter, the generic term Station is used. The terms Flex Station and Console Station are used to indicate if a particular function is applicable only to that type of Station.

Console station is directly connect to CPU and flex station is connect to SERVER , so in case of losing SERVER , flex station will be lose but console station will be live and useable.

The console station has a license. It is not related to the type of configuration. Actually, when installing Honeywell software for STATION, its determine whether STATION is CONSOLE or FLEX.

In addition to a single instance of Station appearing in a single monitor, there are two potential configurations of Station for use with multiple monitors on a single computer:

- Multi-window Station. Using a single instance of Station, you can view multiple displays at one time and manage where and how displays are placed on a desktop that spans up to four monitors.

- Multiple static Station. Allows you to have up to four instances of Station running simultaneously. With this configuration, you can position an instance of Station in each monitor.

The multi-window Station configuration can be used with both a Flex Station and a Console Station. The

multiple static Station can only be used with a Flex Station. Both of these configurations can be used in the Icon Series Console environment.

About multiple static Station

With multiple static Station, you can have up to four instances of Station running simultaneously on the one computer. From a licensing point of view, all instances of Station on the same computer are treated as one Station connection. For example, if you have a license for two static Stations and two client computers, you can run up to eight instances of Station (four on each computer).

The master Station is the instance of Station that was started first. If you close the instance of Station that was started first, the instance of Station that was started second then becomes the master Station, and so on. When configuring the stn files on a computer, you need to ensure that each file defines a separate Station number for each instance—for example, STN01, STN02, STN03 and STN04. You use the Flex Station Configuration Display to configure the way in which the instances of Station interact.

For example, when operators call up trends and groups from STN01 (the master Station), you can direct trends to STN02 and groups to STN03. If you have several monitors, you can use the station command with the -d[n] option to direct each instance of Station to a specific monitor. For example, you could create a batch file in which each station command directs each instance of Station to a separate monitor. (You could then include this batch file in the Startup folder so that each instance of Station starts when the computer is powered on.)

Connection types

Stations can be connected to a server as either:

Type	Description
Static	Provides a permanent, dedicated link to a specific Station. This is the recommended connection type for operators.
Rotary	Provides an 'as required' link to a Station. This is the recommended connection type for staff who do not use Station full-time. Rotary connections are advantageous from a licensing point of view because your license only specifies the number of simultaneously connected Stations.

Regardless of whether you define a Station as static or rotary, the maximum number of Stations that you can have connected to the server (and running the Station software) at any one time is determined by your license. If you define some of your Stations as rotary, you can install and configure a larger number of Stations in your system, but they cannot all be connected to the server simultaneously.

Connection methods

You can set up a Station connection using one of the following methods:

Method	Description
Network	Connects the Station to the server via a network.
Remote Access Service (RAS)	Connects the Station to the server remotely. RAS can support connection via a modem or a serial link.

About Console Station

The Console Station provides direct access to process data and alarms and messages from Control Data Access (CDA) sources such as Process Controllers, FIM, IOLIM, and ACE nodes. Each Console Station contains an Experion CDA server to communicate directly with the CDA devices. This direct access provides a continuous view of your process, even if the Experion server is unavailable.

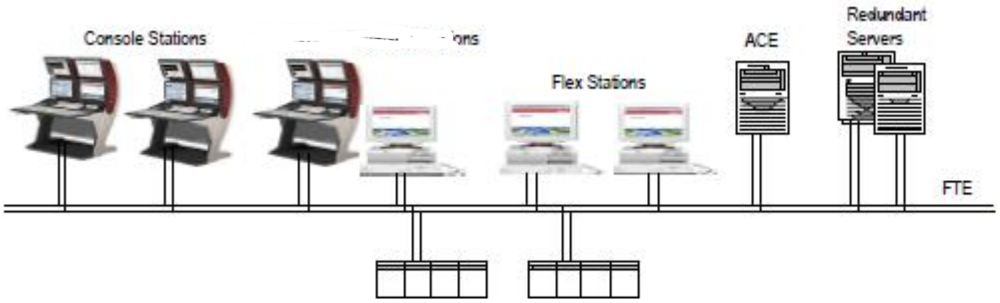
After you configure the connection to the Experion server, the server database files are replicated to the Console Station. This means that configuration of items such as process points is only done once. However, some functionality such as reporting, history and events collection, and flexible point data are still provided by the Experion server. Therefore whenever the Experion server is unavailable, this functionality is not available on the Console Station.

A Console Station can also have clients connected to it. These clients are called Console Extension Stations. A Console Extension Station connects to a Console Station in the same way a Flex Station connects to an Experion server. A Console Extension Station has the same functionality as a Console Station.

A Console Station and Console Extension Station can operate in the following environments:

- Distributed system architecture
- Icon Series Console environment
- Multi-window environment

The following figure shows an example architecture including Console Stations.



Process Controllers

Functionality available on Console Stations

The following functionality is always available on Console Stations regardless of the availability of the Experion server:

- Data access from direct data sources such as Process Controllers and ACE nodes
- Notifications from direct sources
- Security restrictions, for example security levels and asset assignments
- Real-time trending of data from direct data sources such as Process Controllers
- Event journaling of Console Station events such as operator actions, for example, acknowledging alarms and logging on
- Display printing
- Backup and restore of the Console Station database
- 'Server-less' restart, whereby the Console Station can restart and begin communicating with process
- controllers, without needing to connect to the Experion server.

Engineering functionality not available on Console Stations

The following engineering functionality is not available on Console Stations and needs to be accessed via a Flex Station.

- Controller configuration displays for the following controllers:
  - Allen-Bradley PLC-5
  - Bristol Babcock
  - FSC
  - Series 9000
  - Safety Manager
  - TDC 3000
  - UDC
- HC900 and UMC800 set point program and recipe configuration for the following types of controllers:
  - HC900/UMC800
  - Series 9000

What happens when the Experion server is unavailable

When the Experion server becomes unavailable, for example during a server failover, an alarm is raised on the Console Station to indicate that the Console Station is operating in a server unavailable mode. During this time, the following functions on the Console Station are unavailable:

- History collection, assignment, retrieval and display
- Trend configuration
- Group configuration
- Chart visualization (Embedded CB Charts/Forms)
- Event and SOE Summary display
- Alarm and event printing
- Adding comments to alarms and events
- Report requests or report printing
- View or control of points and alarms from server connected controllers or other servers in a DSA configuration
- External alarm notification from server connected controllers
- Server configuration changes, for example server wide settings and asset assignment
- Run-time scope of responsibility change by asset profile or Console membership
- Engineering Tools configuration load/upload and snapshot save/restore

Limitations when performing Engineering operations from a Console Station and the server is shut down

The following table shows the limitations when the server is shut down and you perform Engineering operations from a Console Station.

Engineering operations performed when servers were running and the Console Station was shut down	Limitations on starting up console after servers are shut down
Load new points to the controller	Checkpoint files on the Console Station will not be latest. Restoring from this checkpoint file will not include the newly loaded point.
Rename the point	Alarms will be reported with the old point name instead of newly renamed point.
Change Foundation Fieldbus Advanced Alarm configuration	Foundation Fieldbus Advanced Alarm will be reported with either old or empty conditions.
Delete controller	Any active alarms on the deleted controller that appeared on the console before shutting down will still appear to be in alarm. Once the server comes back online and the Console Station synchronizes with the server, these active alarms from the deleted controller will be removed from the console.

How server wide settings relate to Console Station

The server wide settings that you can configure using the Server Wide Settings Station displays are applicable to Flex Stations, Console Stations, and Console Extension Stations. When you add a Console Station to your system, the server wide settings that have been defined on the server to which the Console Station is connected are automatically applied to the Console Station and any connected Console Extension Station clients. Server wide settings include:

- Startup displays
- Timeouts
- Alarm and event options

How global cache settings relate to Console Station

Console Station Subscription check time and Cache age time are fixed at 5 and 15 seconds respectively, and cannot be changed.

Security

As with a Flex Station, you can use either Station-based or operator-based security with a Console Station and Console Extension Station. You also use assignable assets to further limit access to your Experion system. The type of security you choose for the Console Station applies to any connected Console Extension Stations.

Likewise, the asset assignment you apply to the Console Station also applies to any connected Console Extension Stations.

If you are using operator-based security:

- You need to define operators on the server as well as selecting the Operator sign on required option when you configure the Console Station.
- You assign assets to operators to define an operator’s scope of responsibility.
- You can assign assets to operators as well as Console Stations and consoles and configure your system so that the asset assignment for both the operator and the Console Station or the console are checked before access is granted to an asset.
- You can also use integrated security. Integrated security allows you to combine Windows user accounts with Experion operator accounts.

If you are using Station-based security, you assign assets to the Console Station to define its scope of

responsibility. This can be changed at run-time by the operator by either changing the Asset Profile assigned to the Console Station, or by changing the Console membership of the Console Station.

If you have domain controllers you can use high security policy. High security policy enables you to use Windows group policy to apply restrictions to particular groups of users.

Configuring a Console Station

To configure a Console Station you need to complete tasks on both the Experion server and the Console Station.

Prerequisites

- You have completed all of the tasks for installing Experion Console Station. See the Getting Started with Experion Software Guide.

- You can verify hosts files are consistent among Stations and server when you use the hosts file verification tool as documented in the section 'Verifying the hosts file'.
- You have synchronized the date and time on the Console Station with the date and time on the server, as documented in the section, 'Setting up date and time synchronization' in the Supplementary Installation Tasks Guide.

Complete the following tasks on the server

Task
Configure Console Station details.
Configure Console Station options
Assign scope of responsibility to the Console Station
Configure the minimum system alarm priority, if required. This setting is located on the **Assignment** tab of the Console Station display. It controls which system alarms will be shown. Only alarms with the specified priority or higher will be displayed.
Configure File Replication so that any custom displays, summary display views and mapping files (used for notification and category attribute mapping) are replicated to the Console Station.
If you have an OEP or IKB keyboard, configure LEDs.
If you are using operator-based security, define operators on the Experion server.
Optionally add Console Extension Stations to the Console Station
Optionally configure consoles and add Console Stations to consoles
If you have redundant servers, synchronize the servers so that the Console Station configuration is replicated to the backup server.

Complete the following tasks on the Console Station

Task
Configure the Station connection properties.
Create a file replication share.

Results

- Configuration of the Console Station is successful when:

– The server name appears in the Status Bar and there is no red LED. (A yellow LED indicates that the Console Station is synchronizing with the server.)

– The Node Status and the Station Status are set to OK on the Console Station Status Summary display.

Configuring Console Station details

This task can be done using any Station that is already connected to the same server to which the Console Station is to be connected.

If you try to configure more Console Stations than you are licensed for, you receive an error message.

To configure Console Station details

1. In the Configuration Explorer of Configuration Studio, click Stations and Consoles.

2. Click the Configure Console Stations task.

The Console Station Configuration Summary display appears in the System Display tab in Configuration Studio.

3. Click a blank row to call up the Console Station Configuration display.

4. In the Network Name box type the computer name of the Console Station.

5. If you want to specify operator-based security for this Console Station or for the Console Extension Stations

that connect to this Console Station, select Operator sign-on required.

6. Type the remaining required information as appropriate.

7. Click the Options tab and configure the options as appropriate.

Console Station configuration properties

Property	Description
Network Name	The computer name of the Console Station computer.
Console Extension Stations	The number of Console Extension Stations that you want connected to this Console Station.
Operator sign-on required	Specifies whether you want to use operator-based security for the Console Station or Console Extension Stations that connect to this Console Station.
Console	For more information about Console selection see "Adding Console Stations to a Console"
Normal Update rate	The rate, in seconds, at which dynamic values on displays at this Console Station are updated from the server database.
Fast update rate	An optional fast update rate, in seconds, that can be set by function key for displays at this Console Station.
Station Alias	Applicable only if you are adding this Console Station to a console. An alternative name for this Console Station that can be used to specify the destination of a display when invoked from another Station within the console. For more information about aliases see 'Sending displays using aliases'.

3 Control Builder

Experion control strategies are built using Control Builder**,** a graphical, object-oriented tool that supports Experion C300 Controller system design, documentation, and monitoring. It provides comprehensive handling of I/O and covers continuous, logic, motor, sequential, batch and advanced control functions through a library of Function Blocks **(**FBs**)**.

Function Blocks are basic objects to execute different control functions. Each block supports parameters that provide an external view of what the block is doing. FBs easily interconnect via “soft wires” to construct control applications or strategies.

Function Blocks are grouped together and contained in Control Modules **(**CMs**)** and, in the case of sequential FBs, Sequential Control Modules **(**SCMs**)**. SCMs greatly simplify batch logic implementation by sequencing a group of process equipment through a series of distinct steps to accomplish one or more process tasks. CMs and SCMs act as **“**containers**”** for Function Blocks.

3.1 Physical Equipment Block

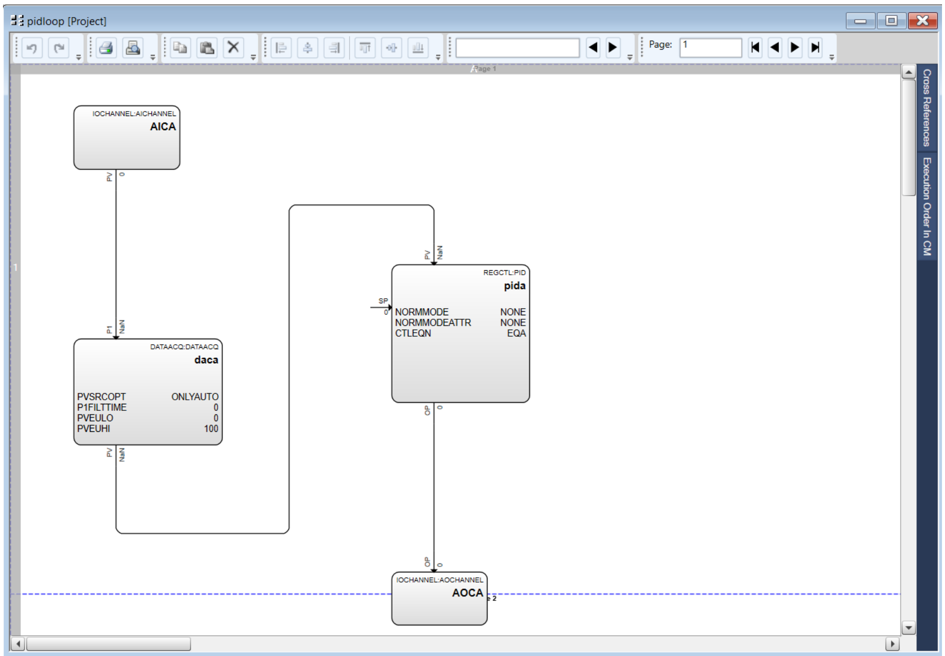


Figure below An example Control Module in Control Builder

Figure below illustrates a simple Control Module -- in this case a PID loop -- made up of basic FBs.

Note that in this example several FBs, i.e. AI, DACA, PIDA and AO, are “contained” within the CM named PIDLOOP.

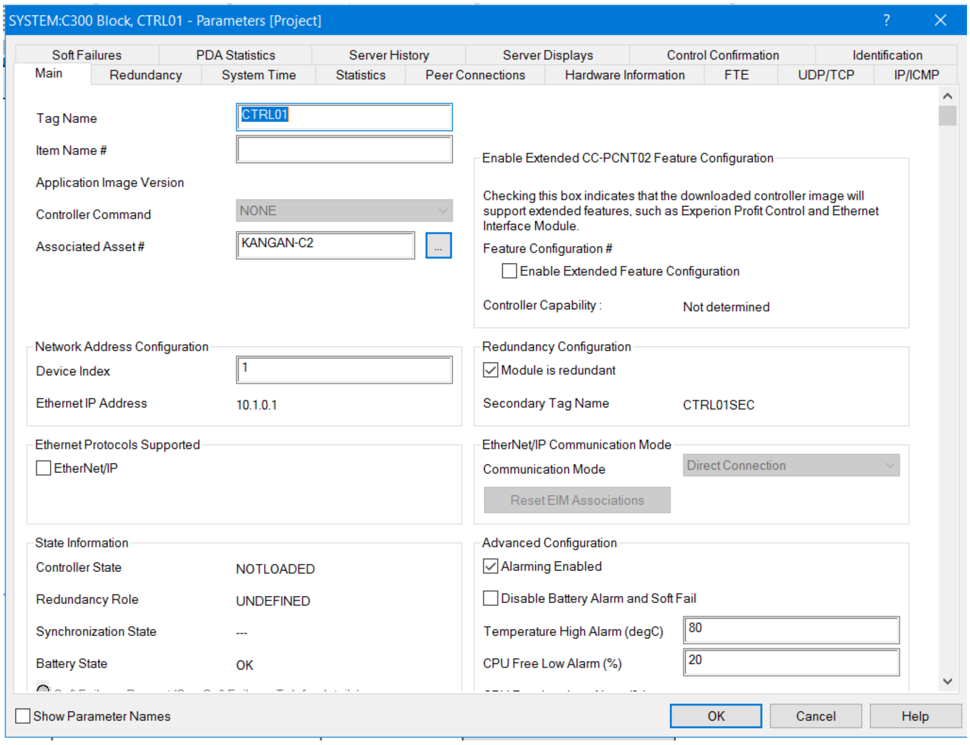
Each Control Module maybe scheduled at its own execution rate from 50 msec to 2 sec. The default execution rate for this project is 1 sec.

3.1.1 Control Processor Module (CPM) Block

The CPM block represents the physical primary and secondary CPM and their associated Control Execution Environment (CEE) to implement the control strategy built in the Control Builder application. It is redundancy compliant.

Each of CPM block shall be assigned a unique name that indicates its location on the network and which area it controls.

The following figure illustrates the configuration of CPM Block of this Project.



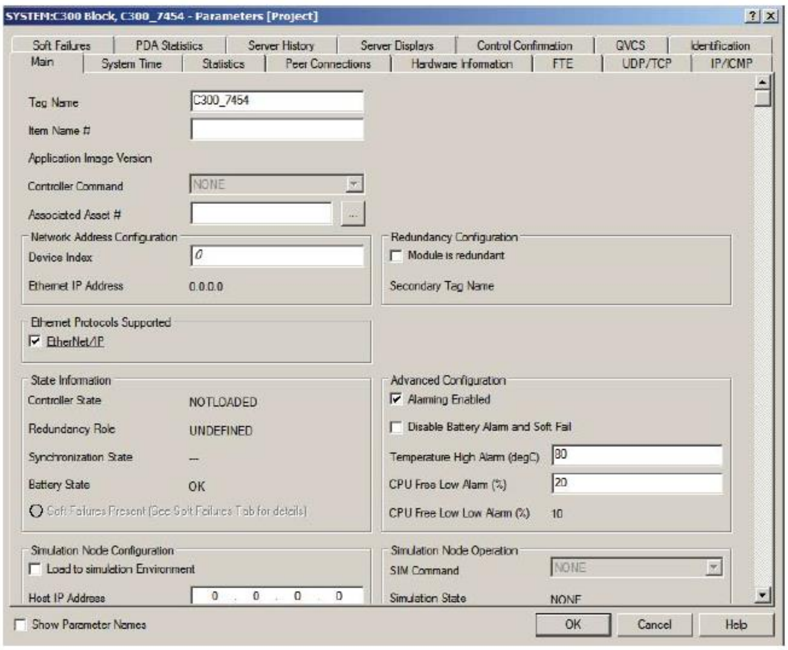
CPM Block Configuration

01

<u>*Configuration CMP*</u>

1. <u>***Click file > new >controllers > C300 controller (2 I/O Links)***</u>

<u>***Calls up the C300 block configuration from with a default tag name field highlighted***</u>



2. Key in desired name of up to 40 characters or accept the default. Press <Tab>

Note Tag Name is limited to 16 characters and when creating redundant controllers, the last four characters of the secondary controller tag name will be _SEC. Therefore, you should create a tag name no longer than twelve characters to allow for the suffix of the secondary controller tag name.

Moves cursor to the Item Name field.

3. Enter Item Name. Press <Tab>.

Moves cursor to Associated Asset field.

4. Click the button to the right of the Associated Asset box.

The Point Selection dialog box appears.

5. Select an asset from those configured in the Enterprise Model Database to set the Scope of Responsibility (SOR) for the point.

Note: No validation is done at the configuration time. If you enter an asset that does not exist in the points database, the associated asset for the point reverts to the server point. If the asset does exist but is not an area-enabled asset, then the first area-enabled asset up the tree is used for the SOR of that device. A subsequent upload of that device point to Control Builder returns the area-enabled asset and not the original non-assignable asset entered.

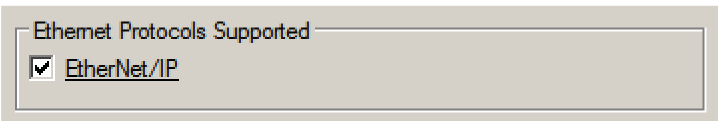
Moves cursor to Device Index field.

6. Click on the value and enter a valid Device Index number. Click o odd numbers for primary controllers. Press <Tab>.

   Moves cursor to Module is redundant check box.

7. Check the box if the controller will be one of a redundant pair. If you check the box, the Secondary Tag Name will appear in the Redundancy Configuration box. C300 Controller is configured as either non-redundant or redundant with a Secondary C300 Controller block added when the Primary controller FB configuration form is closed.

8. If you want to enable EtherNet/IP supportability, select EtherNet/IP. This is a licensed feature.



9. *The Alarming Enabled check box contains a check (default). Press <Tab>.*

To disable alarming for the parameters in the Advanced Configuration box, clear the Alarming Enabled check box.

Moves cursor to the Disable Battery Alarm and Soft Fail field.

10. *If you have the C300 battery backup, clear the box to allow the battery-related soft failures and events to be reported.*

If you do not have the C300 battery backup, check the box to suppress battery-related soft failures and events.

Moves cursor to the Temperature High Alarm (deg. C) field.

11. *Click on the Temperature High Alarm (deg. C) field.*

Accept the default or enter a value at which an alarm is generated for controller hardware high temperature. Press <Tab>.

Note setting this value at 0 disables the temperature high alarm. Moves cursor to the CPU Free Low Alarm (%) field.

12. *Accept the default or key in desired value. Press <Tab>.*

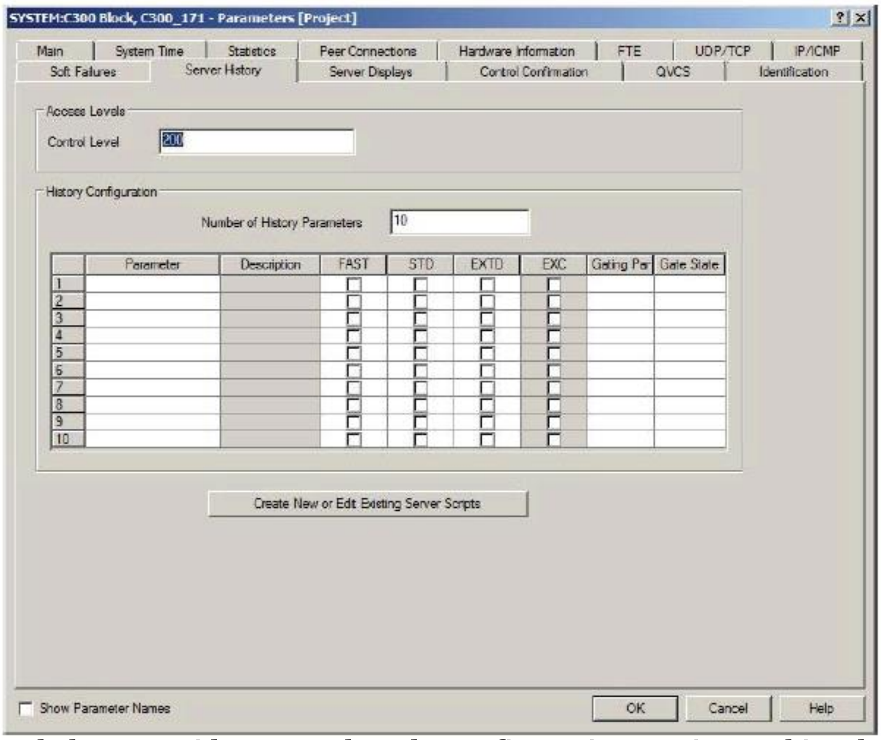
Note: Do not set lower than the default value of 20%. Moves cursor to Simulation Node Configuration.

13. *If this block is to be loaded to a simulation environment, check this box. Press <Tab>.*

Moves cursor to Host IP Address field

14. *If the controller is to be a SIMC300 controller you must enter the IP Address of the SIMC300 controller. Cursor moves to the Host Name field.*
15. *Enter the host name of the SIMC300 controller if it does not appear in the field.*
16. *Click Server History tab.*

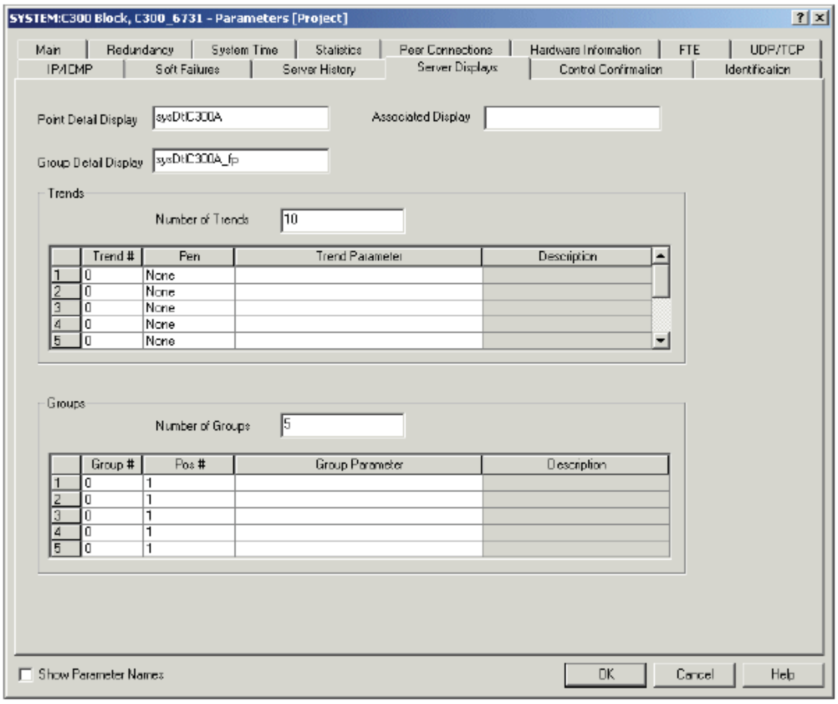
Calls up the Server History configuration form.



17. *__Use the on-line help as a guide to complete the configuration entries on this tab.__*

__Click the Server Displays tab.__

__Calls up the Server Displays configuration form.__

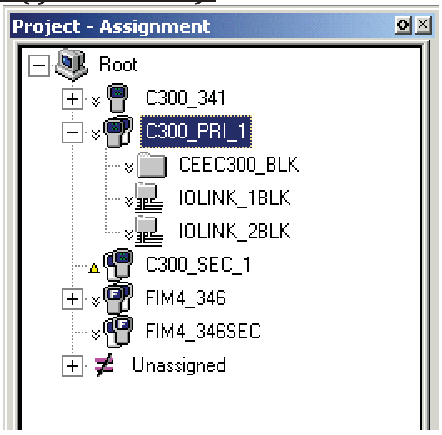


18. *__Use the on-line help as a guide to complete the configuration entries on this tab. Click the OK button.__*

__Closes the form.__

__The following block icons now appear in the Project window:__

- *C300 Controller block, and*
- *its associated CEEC300 block*
- *2 IOLINK blocks and*
- *a C300 Secondary Controller block (if redundant).*



3.1.2 Control Execution Environment (CEE) Block

The CEE block provides control functionality for associated CPM block. Although the block is a functional type instead of a physical one, it is regarded as physical one because of its relationship with the Control Processor.

Each of CEE block as an integral part of CPM block shall also be assigned a unique name that indicates its location on the network and which area it controls. The following figure illustrates the configuration of CEE Block of this Project.

Configuration CEE

01

A CEE block is created automatically when a C300 Controller function block is added to the Project tab.

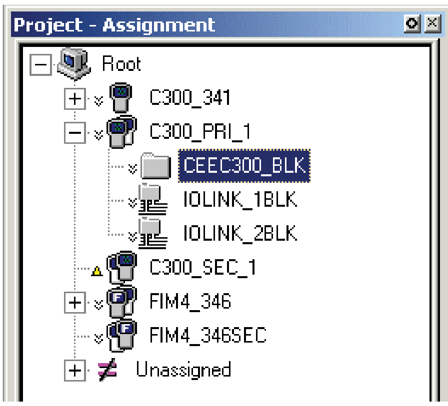
All illustrations used in the procedure are for example purposes only.

Prerequisites

- *Control Builder is running*
- *Tree windows are open*

To configure a CEEC300 function block

1. *In the Project window, right-click on the CEEC300_BLK block icon.*



Calls up shortcut menu.

2. *Click Module Properties.*

   Calls up CEEC300 Block configuration form. The Tag name field is highlighted.

3. *Key in desired name of up to 16 characters or accept the default. Press <Tab>. Moves cursor to Item Name field*

4. *Enter Item Name. Press <Tab>.*

   Moves cursor to Base Execution Period field.

5. *Select the Base Execution Period from the drop down list. Press <Tab>.*

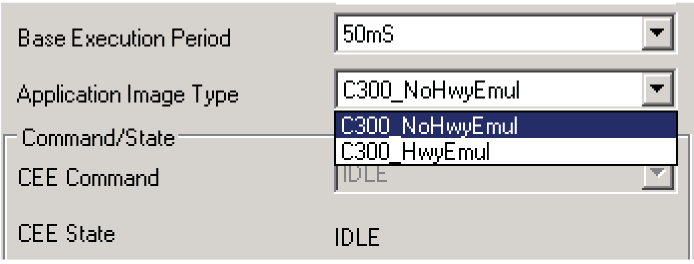
   Note: Prior to selecting the IOLINK configuration for the controller, the base execution period must be selected. Otherwise, an error message appears as follows: "CEE BASEPERIOD can not be changed because IOLINK TYPE of at least one IOLINK of this C300 has been configured."

   For Honeywell Turbine Control Solutions, select the Base Execution Period as 20 ms. Moves cursor to User Lock for CEE Run field.

Note that CEE Command field is available only in Monitor mode.

6. *Select the Application Image Type from the drop-down list. Press <Tab>.*

The Application Image Type distinguishes the type of application image running in the C300 and indicates whether the image supports C300 functions with EHB emulation e ul) or C300 functions without EHB emulation enablers (C300_NoHwyEmul).



7. *Accept the default or click*

down-arrow button and select desired user. Press <Tab>.

Moves cursor to User Lock for CEE Idle field.

8. *Accept the default or click*

down-arrow button and select desired user. Press <Tab>.

Moves cursor to Program Access may command CEE from Idle to Run check box.

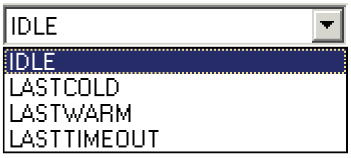
9. *Add a check to the box if you want to enable access. Press <Tab>.*

Moves cursor to Program Access may command CEE from Run to Idle check box.

10. *Add a check to the box if you want to enable access. Press <Tab>.*
    Moves cursor to the Alarming Enabled check box.
11. *Accept the default (Alarming Enabled), or uncheck the box to disable alarming function. Press <Tab>. Moves cursor to the Enable Memory Limit Exceeded Alarm check box.*
12. *Accept the default (Memory Limit Exceeded Alarm enabled), or uncheck the box to disable alarming function. Press <Tab>.*

Moves cursor to the Powerup Restart Settings for CEE State.

13. *Accept the default or click down-arrow button and select desired state.*



Note that if you select LASTTIMEOUT state, then the Warm Timeout field is active.

14. *If the Warm Timeout field is active, accept the default or key in desired timeout value. Press <Tab>.*
    Moves cursor to the Time Zone field.
15. *Enter the appropriate value for the time zone of the controller location. Press <Tab>.*
    Moves cursor to the Daylight Savings Time check box.
16. *Add a check to the box if Daylight Savings Time is in effect for this location. Press <Tab>.*

Note: The Daylight Savings Time check box is selected/cleared by default at the start/end of DST respectively if you have enabled the Automatically apply DST option.

Moves cursor to the Year Format field.

17. *Accept the default or click down-arrow button and select desired format. Press <Tab>.*

Moves cursor to the Weekday Format.

18. *Accept the default or click down-arrow button and select desired format.*

Click on the Peer Configuration tab.

The Store Expiration Time Field is highlighted in the Peer Defaults box.

19. *Accept the default or key in desired value. Press <Tab>. Moves cursor to Subscription Period field.*

20. *Accept the default or click down-arrow button and select desired period. Press <Tab>.*

Moves cursor to Number of Peer Environments field.

21. *Key in number of peer environments for this controller. Press <Tab>.*

Or, skip this field, if no peer environments will be used. Click Server History tab and go to Step 25.

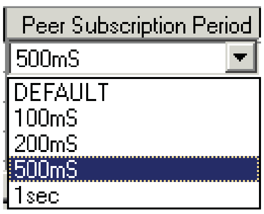
If peer environments will be used, cursor moves to the Peer Environment Name of column in the Peer Environment Table.

22. *Key in valid name for existing peer environment.*

Press <Tab>.

Moves cursor to Peer Subscription Period column.

23. *Accept default or click down-arrow button to select another value specific to the given environment.*



Press <Tab>.

Moves cursor to Store Expiration Time column.

24. ***Accept the default or key in desired value. Click Batch tab. Moves cursor to Batch Events Memory field.***

25. ***Accept the default or click down-arrow button and select desired state. Click Server History tab.***

Calls up the Server History configuration form.

26. ***Use the on-line help as a guide to complete the configuration entries on this tab. Click on the Server Displays tab.***

Calls up the Server Displays configuration form.

27. ***Use the on-line help as a guide to complete the configuration entries on this tab. Click the OK button. Closes the form.***

This completes the configuration procedure for CEEC300 block.

3.1.3 *IOM block:*

01

Configuring the Main tab - IOM block:

The following configuration information pertains to the Main tab for all Series C module types.

- ***Redundancy is determined by checking the, This IOM is redundant check box.***
- ***The QVCS tab becomes visible when this option is obtained and enabled.***
- ***All illustrations used in the procedure are for example purposes only***

The following is an example of a DO-24b Block, Configuration form - Main tab.

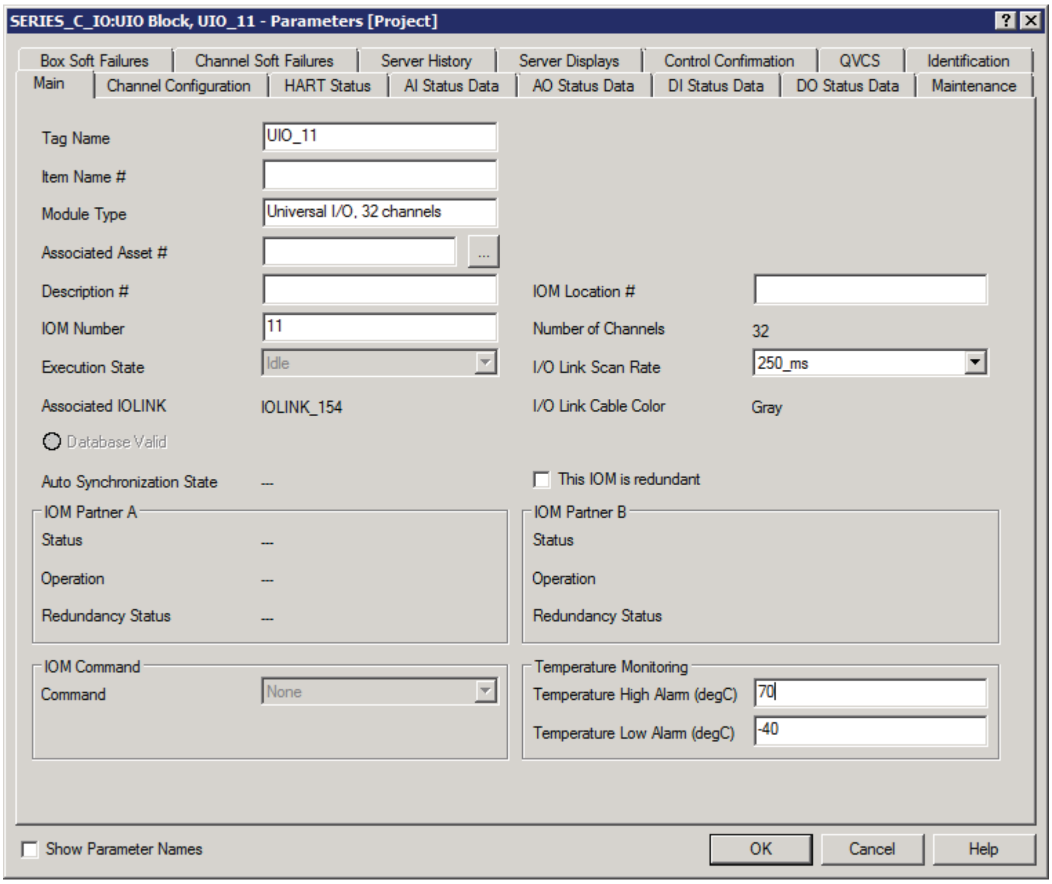


Figure 31: Main tab

The parameters of the Main tab is listed in the following table.

Table 34: Main tab parameters

Plain text	***Parameter name***	***User configurable***	***Notes***
Tag Name	***TAGNAME***	***Project Only***	***System assigned or user configured unique name. Consisting of up to 16 characters and at lease one charactermust be a letter (A-Z).***
Item Name	***ITEMNAME***	***Project Only***	***A non-unique name by which an entity is known within the context ofthe enterprise model.***
Module Type	***IOMTYPE***	***No***	***This non-configurable parameter is adescription of the respective I/O module***

Plain text	Parameter name	User configurable	Notes
Associated Asset	ASSOCASSET	Yes	Allows user to select an asset fromthose configured in the Enterprise Model Database to set the Scope of Responsibility (SOR) for the point.
Description	DESC	Yes	Used to specify descriptive text for the function block. Appears on bothdetail and group displays.
IOM Number	IOMNUM	Yes	This parameter value must be unique on a specific I/O Link, and its range must be between 1 and 40. If IOMNUM=0 and the IOP block is assigned to an I/O Link block, ControlBuilder automatically defaults the IOMNUM to the next available value on the I/O Link.
Execution State	IOMSTATE	No	Defines execution state
Associated IOLINK	IOLINK	No	Defines associated IOLINK
Database Valid		No	Defines if database is valid
IOM Location	IOPLOCATION	Yes	Identifies the user-entered location (within the plant) where this IOM canbe found.
Number of Channels	NUMCHANS	No	The number of channels available inthe IOM
I/O Link Scan Rate	SCANRATE	Yes	Defines scan rate
I/O Link Cable Color	IOLINKCOLOR	No	Defines cable color. See “I/O Link Address Jumpers” onpage 37 for color listing.
This IOM is redundant	IOREDOPT	Yes	Module is part of redundant pair.
Status	IOMSTSA	No	Defines status of IOM: Idle, OK, No Response
Operation	IOMOPER	No	Defines operation status of IOM: Primary, Secondary
Redundancy Status	REDDATA	No	Defines if redundancy is enabled
Frequency 60/50Hz(AI LLMUX only)	FREQ6050	Yes	User-configured power supplydetermination: 50Hz or 60Hz (AI-LLMUX, DIHV)
Command	IOMCOMMAND	No	Swap primary, reset errors, etc.

DI Mode	DIMODE	Yes	Defines the mode of the DI-24V and DI-SOE modules to be Normal, SOE,or Low Latency
Configured Prover Signal	CONFIGPROVERSIGNAL	Yes	Provides the user the ability to specifyon any given Pulse Input Module the signal stream to output to the prover signal screw pair. Only dual streams can be selected.
Actual Prover Signal	ACTUALPROVERSIGNAL	No	Provides the user or a program the actual signal stream being output tothe prover signal screw pair.

Plain text	Parameter name	User configurable	Notes
Maintain Selection on Fault	MAINTAINONFAULT	Yes	Provides the user the ability to specify the action to be taken by the system in case of a special condition or faults in the module or its ability to output the selected stream to the prover signal.
Temperature High Alarm	TEMPHILM	Yes	Enables the user to configure the temperature value in the range from low limit to +70.0.
Temperature Low Alarm	TEMPLOLM	Yes	Enables the user to configure the temperature value in the range from -40.0 to high limit.

To configure the Main tab

1. *Enter a Tag Name that is more meaningful to you than its default pre-assigned number (ex. Test Strategy DI Module).*
2. *Item name is based on relationship established in Enterprise Builder*
3. *Enter the Module Type.*
4. *Click the button to the right of the Associated Asset box.*

   The Point Selection dialog box appears.

Select an asset from those configured in the Enterprise Model Database to set the Scope of Responsibility (SOR) for the point.

Note: No validation is done at the configuration time. If you enter an asset that does not exist in the points database, the associated asset for the point reverts to the server point. If the asset does exist but is not an area-enabled asset, then the first area-enabled asset up the tree is used for the SOR of that device. A subsequent upload of that device point to Control Builder returns the area-enabled asset and not the original non-assignable asset entered.

Enter an optional Module Description to explain the I/O Module's function

Enter appropriate values for IOM Number to match the plug-in IOM jumper number. If necessary, press Fl to access on-line help for assistance during this step.

If redundancy is required, check the This IOM is redundant check box.

If you have configured a Pulse Input Module, select the Maintain On Fault check box, if required. This parameter provides you the ability to specify the action to be taken by the system in case of a special condition or faults in the module or its ability to output the selected stream to the prover signal.

Note: You must perform the remainder of the pulse proving configuration at runtime.

If you have configured a Universal Input/Output Module, configure the Temperature High Alarm value and Temperature Low Alarm value in degree Celsius.

Note: By default, the values of Temperature High Alarm and Temperature Low Alarm are +70 and -40, respectively.

Configuring Server History tab:

The following configuration information pertains to the Server History tab for the following modules:

AI-HART

AO-HART

DI-24

DO-248

PIM

All illustrations used in the procedure are for example purposes only

The following is an example of a DO-248 8lock, Configuration form - Server History tab.

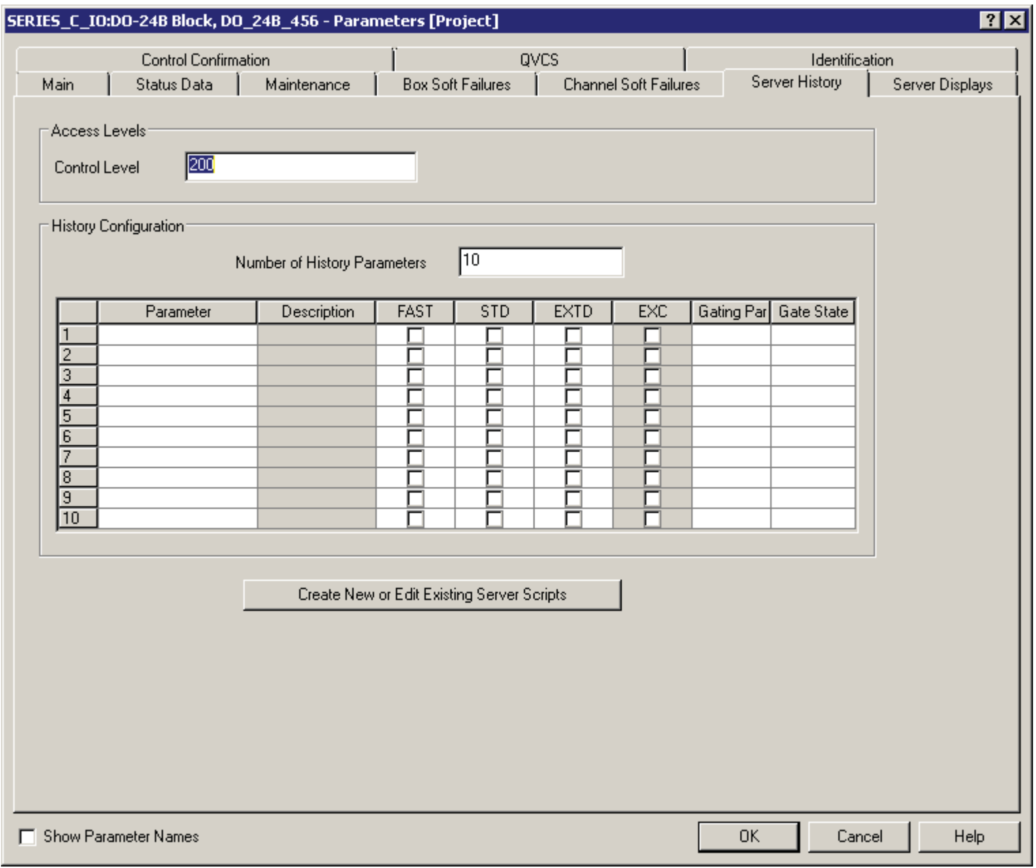


Figure 32: Server History tab

The parameters of the Server History tab are listed in the following table.

Table 35: Server History tab parameters

Plain text	***Parameter name***	***User***	***Notes***
Control Level	***SCANCTRLLVL***	***Yes***	***Indicates Server control level to be associated with this function.***
Number of History	***HIST.NUMPARAMS***	***Yes***	***Defines number of history parameters to be included in***
Parameter	***HIST.PARAM***	***Yes***	***Valid parameter name for a parameter associated with the given point that is to be collected and stored as historical data at***
Description		***No***	***Provides a brief description of the entered parameter.***
FAST	***HIST.FAST***	***Yes***	***Select the Fast type of history***
STD	***HIST.STD***	***Yes***	***Select the Standard type of history***
EXTD	***HIST.EXTD***	***Yes***	***Select the Extended type of history***
EXC	***HIST.EXC***	***Yes (Station only)***	***Select the Exception type of history collection.***

Plain text	Parameter name	User	Notes
Gating Parameter		Yes	Optional gating parameter to define conditions under which data for this parameter should be collected.
Gate State		Yes	Defines gate state for configured gating parameter.
Create New or Edit Existing Server Scripts (Button)		NIA	Launch the Server scripting configuration utility.

To configure the Server History tab:

1. Under the Server History tab, enter the appropriate information for Control Level along with values for appropriate parameters related to history collection and archiving. 1f necessary, press Fl to access on-line help for assistance during this step.
2. Proceed to the following procedures to configure parameters on the remaining tabs for I/O Module, or click OK to accept only the changes made so far and return to the Project tree.

Configuring Server Displays tab:

The following configuration information pertains to the Server Displays Failure tab for the following modules:

- AI-HART
- AO-HART
- DI-24
- DO-24B
- PIM

All illustrations used in the procedure are for example purposes only

The following is an example of a DO-24B Block, Configuration form - Server Displays tab.

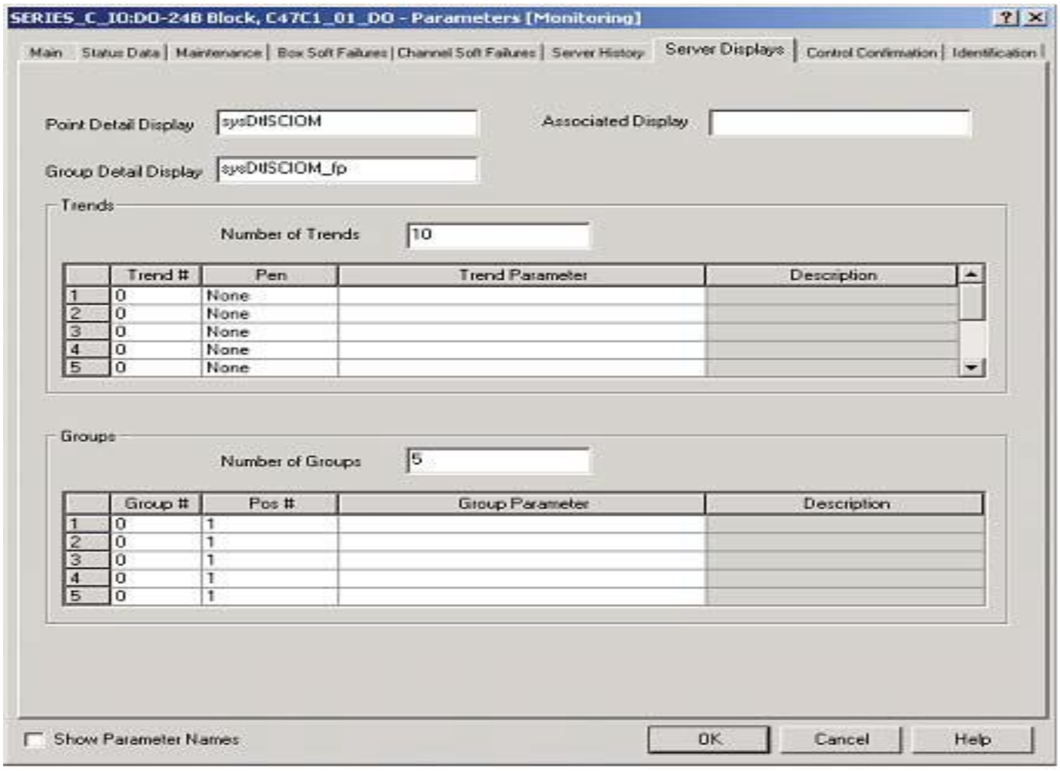


Figure 33: Server Display tab

The parameters of the Server Displays tab are listed in the following table.

Table 36: Server Display tab parameters

Plain text	*Parameter*	*User cofigurable*	*Notes*
Point Detai1 Disp1ay	*SCANPNTDTL*	*Yes*	*By default, a Display template is already entered into Point Detail Display box (for example, sysDtlFTEB.dsp). This template can be used for creating your own display or it can be used as is, provided that your function block name matches name built into play that is supplied as a template.*
Point Detai1 Disp1ay	*SCANPNTDTL*	*Yes*	*By default, a Display template is already entered into Point Detail Display box (for example, sysDtlFTEB.dsp). This template can be used for creating your own display or it can be used as is, provided that your function block name matches name built into play that is supplied as a template.*

Associated Display	***SCANASSOCDSP***	***Yes***	***Name of the erver display to be associated with this function block.***
Number of Trends	***TREND.NUMPARAMS***	***Yes***	***Defines the number of trend parameters to be included in the Trends Configuration table.***
Trend #	_	***Yes***	***Defines Trend numb to be associated with this trend parameter***
Trend Position	_	***Yes***	***Defines position of the pen that will be used to trace assigned parameter on Station Trend display.***
Trend Parameter	_	***Yes***	***Va1id parameter name for a parameter associated with given configured forhistory col1ection.***
Description	_	***NO***	***Provides a brief description of the entered parameter.***
Number of Groups	***GROUP.NUMPARAMS***	***YES***	***Defines the number of group parameters to beincluded in Groups Configuration table.***
Group #	_	***YES***	***Defines Group number to be associated withthis group parameter.***
Pos #	_	***YES***	***Defines number of position configured meter will occupy in the Station Group display.***
Group Parameter	_	***YES***	***Valid parameter name for a parameter associated with the given at is configured in the system.***
Description	_	***NO***	***Provides a brief description of the entered parameter.***

To configure the Server Displays tab.

1 ***Under the Server Displays tab, enter the appropriate information to specify related Point Detail Display, Group Detail Display, and Associated Display along with values for appropriate parameters to define Trends and Groups for display. 1f necessary, press Fl to access on-line help for assistance during this step***

2 ***Click OK on the configuration form to accept all configuration selections made on each configuration tab and to return to the Project tree.***

Configuring Control Confirmation tab - IOM block:

The Control Confirmation tab is common to all configuration forms for tagged blocks in Control Builder. If you have an optional Electronic Signature license, you can configure electronic signature information for the tagged block through this tab on the block's configuration form in Control Builder.

The Electronic Signature function aligns with the identical Electronic Signatures function that is initiated through Quick Builder and Station for Server points.

When this block is loaded to a controller,

- ***its control confirmation configuration (electronic signatures) is also loaded to the Server***
- ***you can view the control confirmation configuration for this tagged object in Station and also make changes to it.***

- If you make changes through Station, you must initiate an Upload or Upload with Contents function through the Controller menu in Control Builder for the object in the Monitoring tab to synchronize changes in the Engineering Repository Database (ERDB).

The following configuration information pertains to the Control Confirmation tab for all Series C I/O modules. Control Confirmation is enabled by checking the Control Confirmation check box.
All illustrations used in the procedure are for example purposes only

The following is an example of a DO-24B Block, Configuration form - Control Confirmation tab.

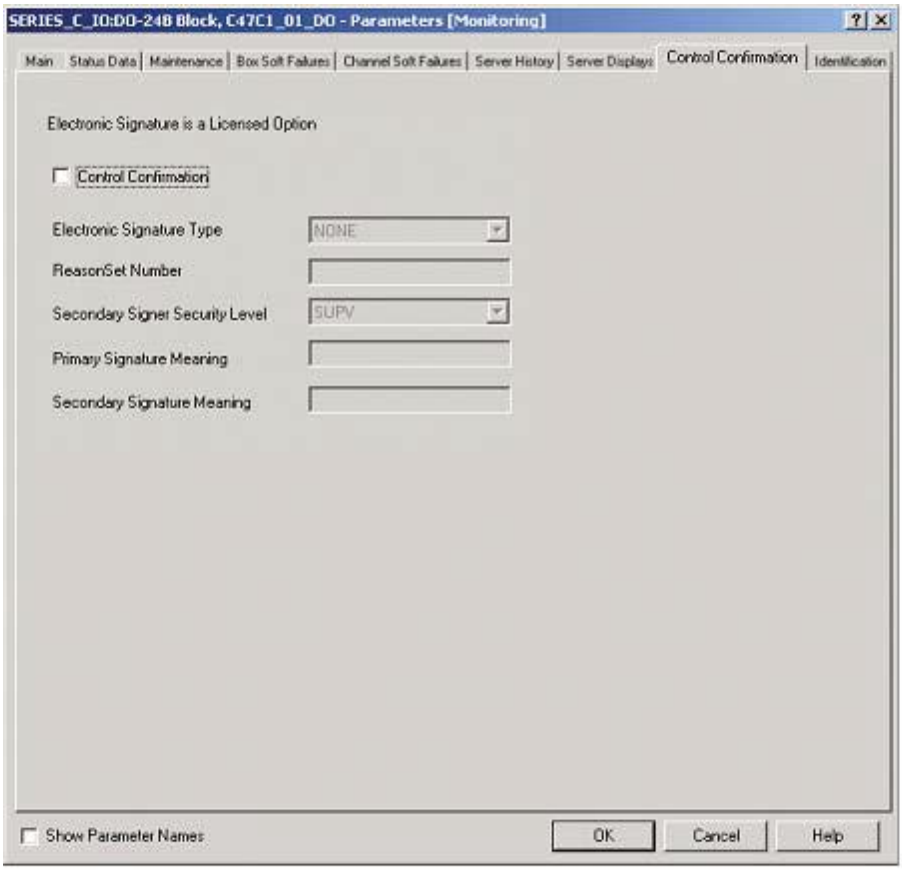


Figure 34: Control Confirmation tab

Prerequisites

- ***Control Builder is running***
- ***A Series C I/O control module was created***

To configure the Control Confirmation tab.

1.Under the Control Confirmation tab, check or clear the Control Confirmation check box. If necessary, press Fl to access on-line help for assistance during this step.

2. If the Control Confirmation check box is checked, the Electronic Signature Type drop-down list is enabled, with the options to select:

- ***NONE***
- ***SINGLE***
- ***DOUBLE***

3. Proceed to the following procedures to configure parameters on the remaining tabs for I/O Module, or click
OK to accept only the changes made so far and return to the Project tree.

Configuring Identification tab - IOM block:

The following configuration information pertains to the Identification tab for all Series C I/O

modules.All illustrations used in the procedure are for example purposes only

The following is an example of a DO-24B Block, Configuration form - Identification tab.

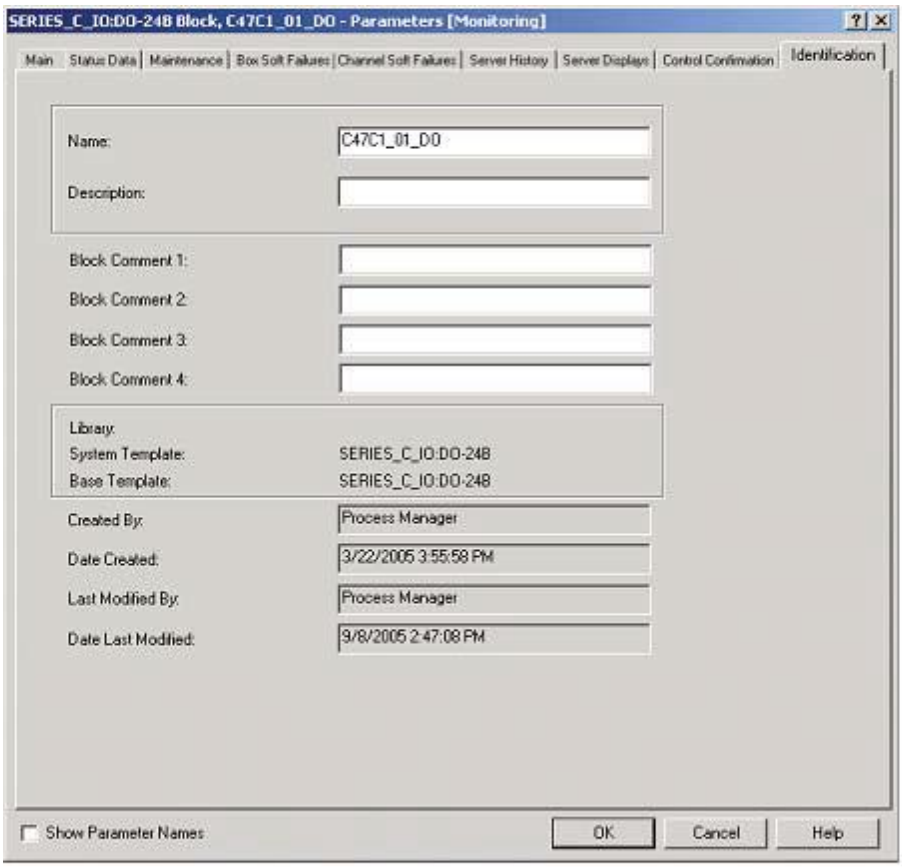


Figure 35: Identification tab

The parameters of the Identification tab are listed in the following table.

Table 37: Identification tab parameters

Plain text	Parameter name	User configurable	Notes
Name	NAME	Yes	Unique block name consisting of up to 16 characters to identify the block. At least onecharacter in the name must be a letter (A-Z).
Description	DESC	Yes	Descriptive text appears on detail and groupdisplays to uniquely describe this particular function block
Block Comment 1 Block Comment 2 Block Comment 3 Block Comment 4	BLCKCOMMENT1 BLCKCOMMENT2 BLCKCOMMENT3 BLCKCOMMENT4	Yes	Comment to be associated with this blockconsisting of up to 40 characters.

Plain text	Parameter name	User configurable	Notes
Library	-	No	Identifies Control Builder Library that is sourceof template.
System Template	-	No	Identifies System Template that is source forthis block.
Base Template	-	No	Identifies Base Template that is used for thisblock.
Created By	CREATEDBY	No	Identifies user who created block, if operatorsecurity is implemented. Otherwise, may justshow Default login.
Date Created	DATECREATED	No	Shows date and time template was created. If this block is in Version Control System, shows date and time initial version of template was created.
Last Modified By	MODIFIEDBY	No	Identifies user who made last modifications toblock, if operator security is implemented. Otherwise, may just show default login. If thisblock is in Version Control System, modifications apply to last version of block.
Date Last Modified	VERSIONDATE	No	Shows date and time last modification was made to block's configuration. If this block is in Version Control System, modification date and time applies to last version of block.

Configuring QVCS tab - IOM block:

Qualification and Version Control System (QVCS) provides version management for all tagged objects and acustomer defined lifecycle management.

The following configuration information pertains to the QVCS tab for the following modules:

- AI-HART
- AO-HART
- PIM

Note: No user-defined configuration setting on the QVCS tab.

All illustrations used in the procedure are for example purposes only

The following is an example of a DI-24 Block, Configuration form - QVCS tab.

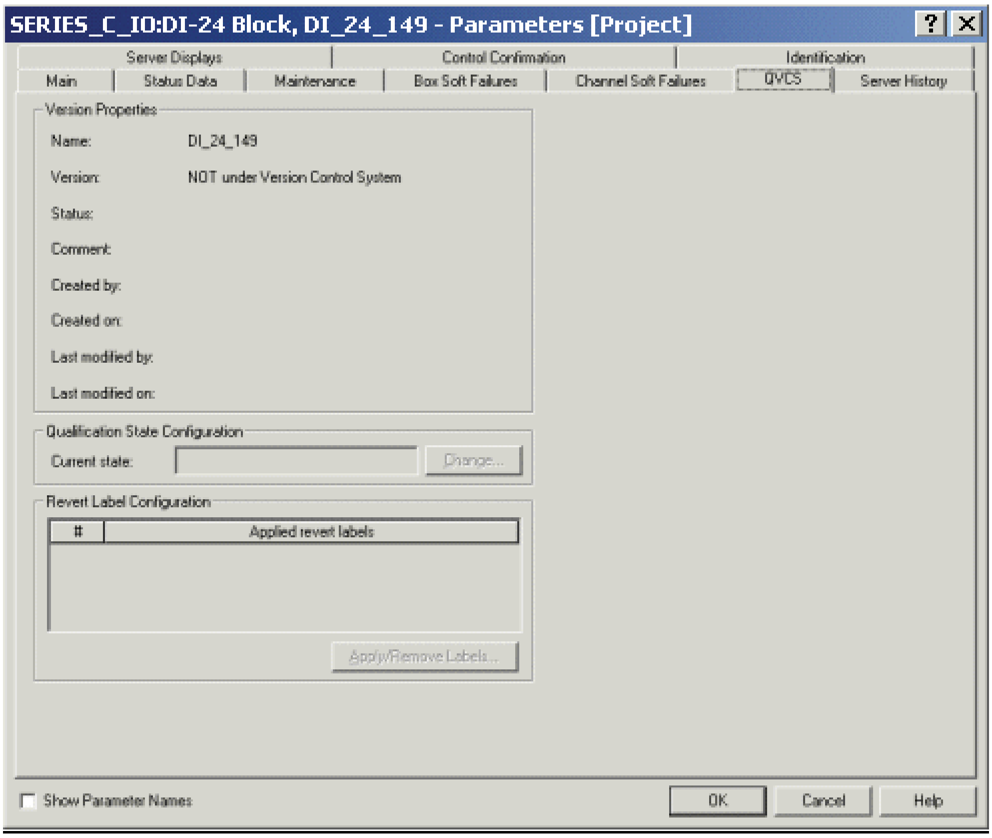


Figure 36: QVCS tab

Configuring the Calibration tab - IOM block:

The following configuration information pertains to the Calibration tab for the following modules:

- *AI-HART*
- *AO*
- *AO-HART*

Calibration can only be done from Monitoring.

The Execution State must be set to IDLE before enabling and modifying calibration status.All illustrations used in this procedure are for illustration purposes only

The following is an example of an AI-HART Block, Configuration form - Calibration tab.

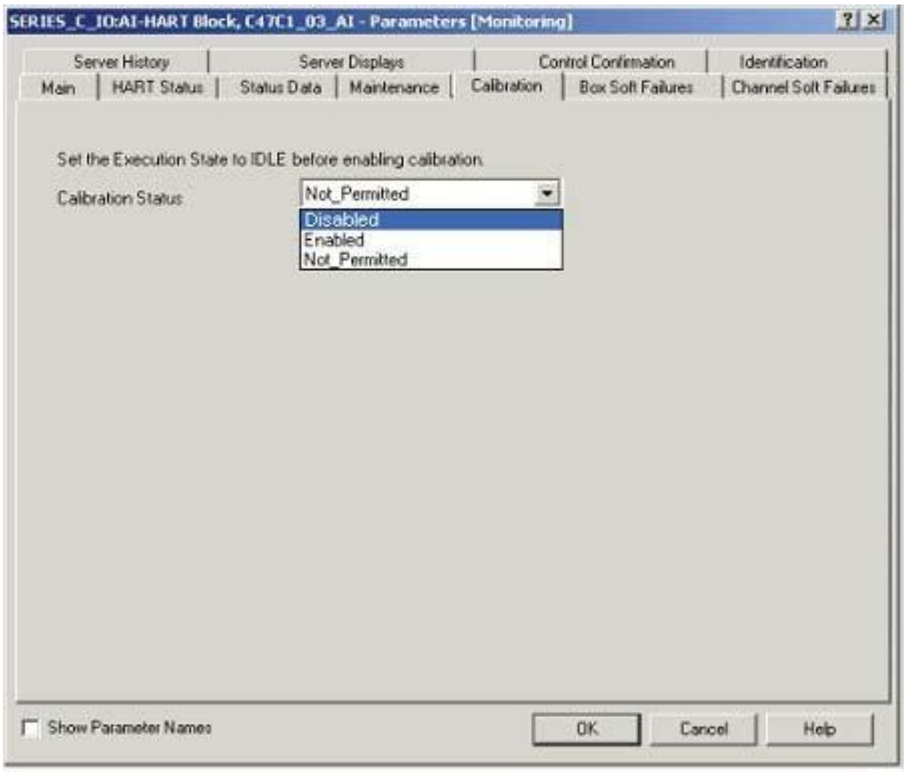


Figure 37: Calibration tab

To configure the Calibration Status

1 ***Select either:***

- ***Disabled indicates disabling Calibration***

- ***Enabled indicates enabling Calibration***

Not Permitted indicates the Execution State of the IOM is still in the RUN state, which does not allow youto select Disable or Enable. Changing the Execution State of the IOM to IDLE allow you to modify the Calibration Status.

2 ***Proceed to the following procedures to configure parameters on the remaining tabs for the module, or clickOK to accept only the changes made so far and return to the Project tree.***

Configuring HART Status tab - IOM block:

The following configuration information pertains to the HART Status tab for the following:

- ***AI-HART***
- ***AO-HART***

Note: No user-defined configuration setting on the HART Status tab and there are no configuration items in HART status tab.

All illustrations used in the procedure are for example purposes only

The following is an example of an AI-HART Block, Configuration form - HART Status tab.

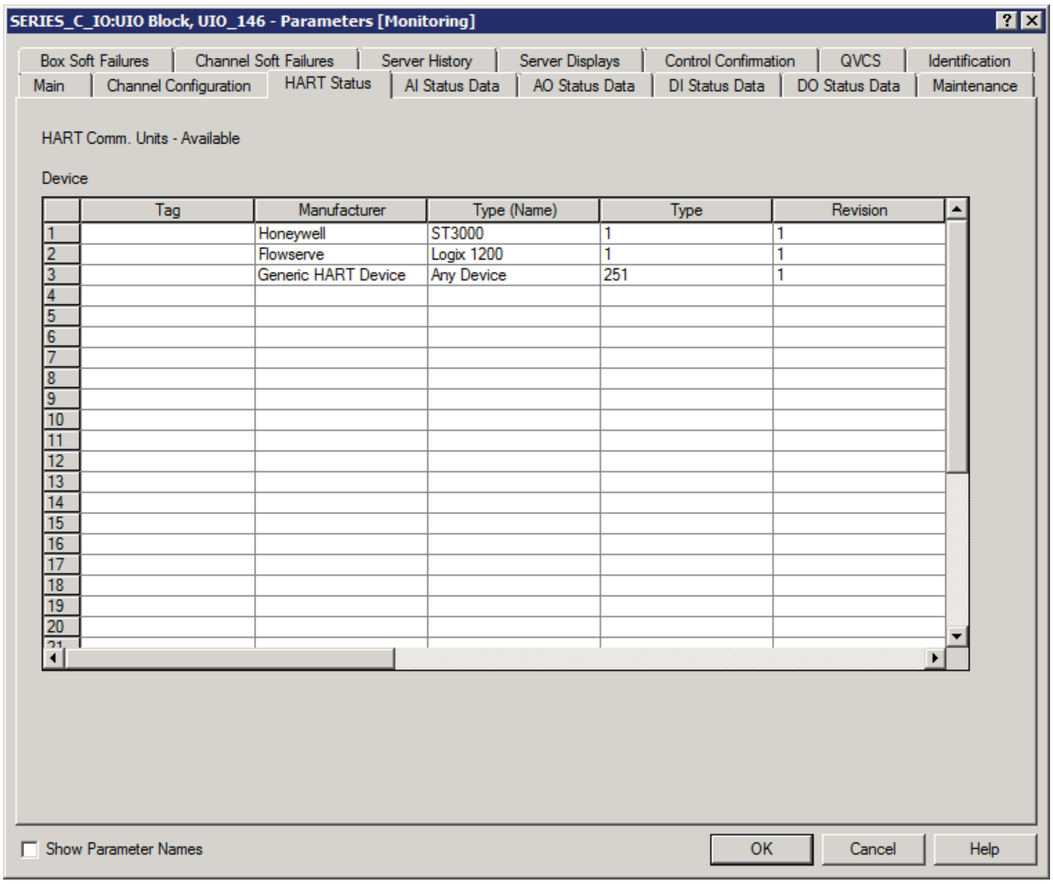


Figure 38: HART Status tab

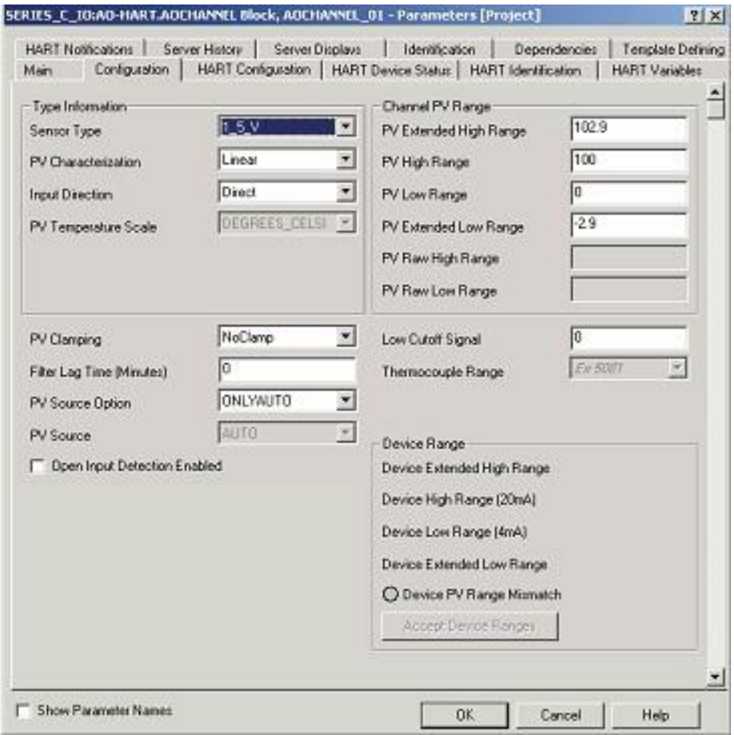
3.1.4 Channel block:

Configuring the Configuration tab -

<u>The following configuration information pertains to the Configuration tab for Series C I/O Analog Input orAnalog Output modules.</u>

<u>All illustrations used in the procedure are for example purposes only</u>

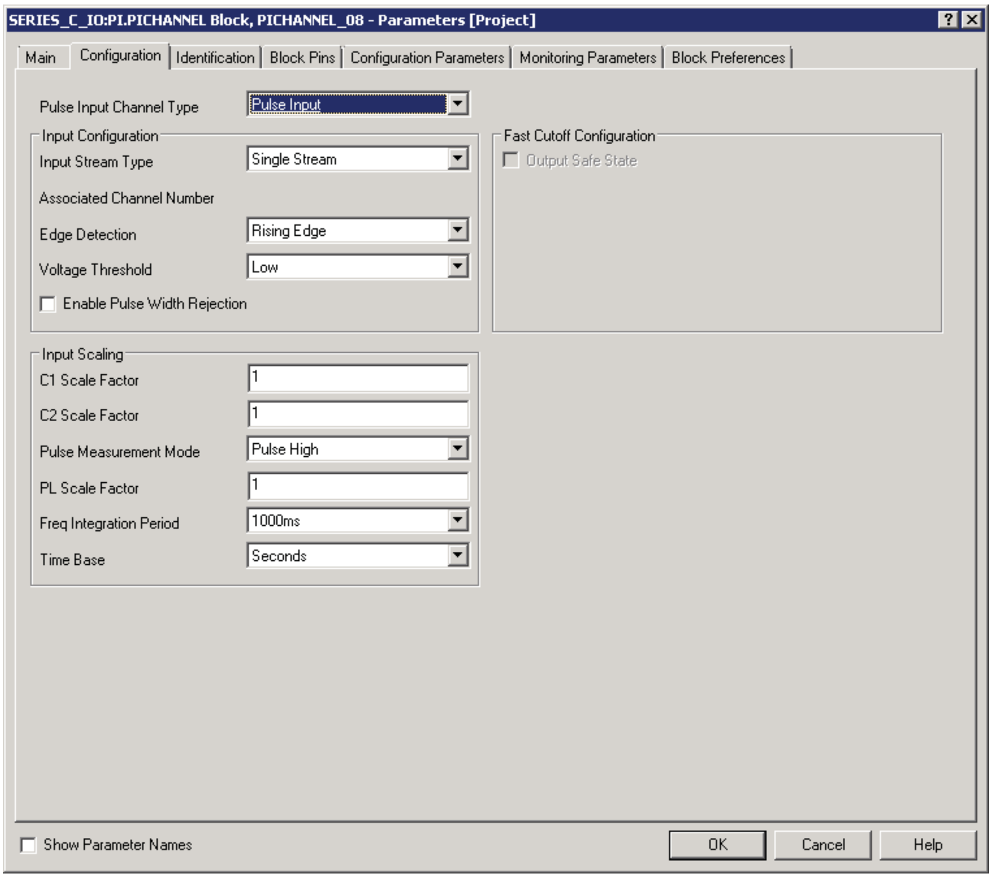
<u>The following is an example of an AO Channel Block, Configuration form - Configuration tab.</u>



<u>Figure 39: Configuration tab</u>

Configuring the Configuration tab - PI channel block:

The following figure is an example of the Configuration tab of the PI channel block.



To configure the Configuration tab

1 *Click the Configuration tab of the channel block.*

The Configuration tab configuration form appears. The value Pulse Input is selected by default in the

Pulse Input Channel Type box.

2 *Select the pulse input type in the Pulse Input Channel Type box. The available options are as follows:*

- *Pulse Input- You can configure all eight channels as pulse input channel types.*
- *Pulse Input with Fast Cutoff- You can only configure channel 7 and channel 8 for fast cutoffapplications.*

The following parameters in the Main tab of the PI channel block are enabled after the module is loadedwhen the input type is Pulse Input with Fast Cutoff.

Output Safe State (SAFEOUTPUT)

- *Target Value (TV)*

TV Processing Run Flag (TVPROC)

- *Output State (SO)*

Bad Output State Flag (BADSO)

- *SO Command OFF (SOCMDOFF)*

SO Command ON (SOCMDON)

Note: These parameters are disabled when the input type is selected as Pulse Input.

1 *Select the input stream type in the Input Stream Type box. You can select one of the following:*

- *Single Stream - You can select this option for all channels.*
- *Dual Stream - You can select this option only for odd-numbered channels.*

When you configure a channel for Dual Stream, its associated even-numbered channel number appears inthe Associated Channel Number box.

For example, if you have configured channel 1 for dual stream, the channel number 2 appears in the

Associated Channel number box.

Attention:

- *When you select the pulse input channel type as Pulse Input with Fast Cutoff for Fast Cutoff channels (channel 7 and channel 8), Single Stream is selected by default and is non-editable.*
- *When you select the pulse input channel type as Pulse Input, you can configure Dual Stream on the FastCutoff channels (channel 7 and channel 8).*
- *Dual Stream option is not applicable for Fast Cutoff channels (channel 7 and channel 8) when the Pulse Input Channel type as Pulse Input with Fast Cutoff. In this scenario, the option Single Stream is selected by default and is non-editable.*

- *If you select the Input Stream Type as Dual Stream for odd-numbered channels, then you cannot configuretheir associated even numbered channels. For example, if you have configured channel 1 as Dual Stream, youcannot configure channel 2. In addition, the associated channel does not appear in the tree view if you have configured the channel for Dual Stream.*
- *If the CM already contains the even-numbered channel and you try to configure its associated odd-numberedchannel as Dual Stream, an error message appears. You need to delete the associated even-numbered channelfrom the CM. For example, if a CM contains channel 3 and channel 4, and you try to configure channel 3 as Dual Stream, an error message appears. You need to delete channel 4 to configure channel 3 as Dual Stream.*
- *When a channel is configured for Dual Stream, the Status Data tab of the PIM does not display any values forthe associated channel in the Monitoring view. For example, if you have configured channel 3 for Dual Stream, the channel 4 row in the Status Data tab does not display any values.*

2. *Select the Enable Pulse Width Rejection check box if you want the pulse widths less than a specified duration to be excluded from counting.*

   Note: Refer to the Series C Spec and Tech data for the specific duration.
3. *Refer to the Control Builder Parameter Reference document for more information on configuring otherparameters.*
4. *Click OK.*
5.

Configuring the DI channel for pulse counting:

DI channel pulse counting functionality is configured for pulse counting.

Prerequisites

For UIO, you can configure up to four channels only on DI channels 15 to 18 for pulse counting.

For UIO-2, you can configure up to four of any of the available 32 channels configured as DI for pulsecounting.

Configuring HART Configuration tab - Channel block:

The following configuration information pertains to the HART Configuration tab for all Series C I/O AnalogInput and Analog Output modules.

The parameter HALARMENABLE is added to HART configuration tab in the channel block and:

- is enabled by default
- in addition, can be changed from monitoring side, irrespective of the channel point

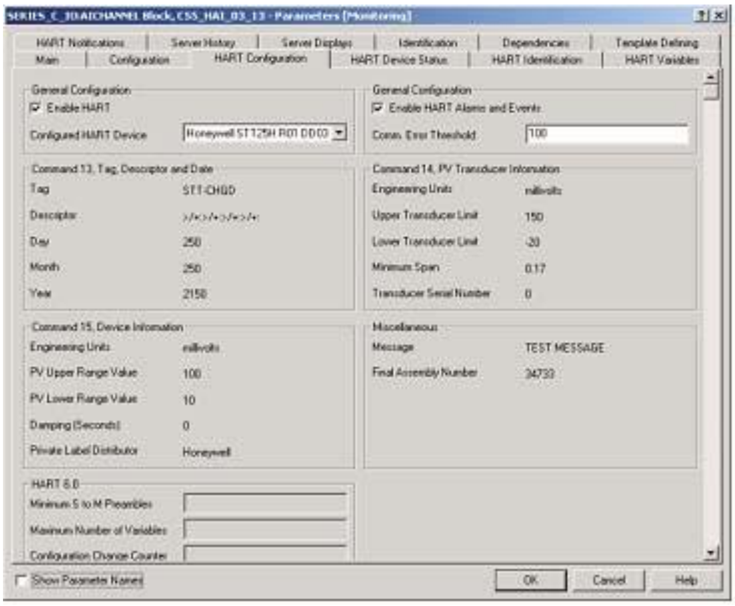


Figure 43: HART Configuration tab

The parameters of the HART Configuration tab are listed in the following table.

Table 38: HART Configuration tab parameters

Plain text	Parameter name	User configurable
General configuration		
Enable HART	HENABLE	Yes
Enable HART Alarms and Events	HALARMENABLE	Yes
Configured HART Device	HCFGDEV	Yes
Comm. Error Threshold	HCOMTHRS	Yes

Plain text	Parameter name	User configurable
Command 13, tag descriptor and date		
Tag	HTAG	No
Descriptor	HDESC	No
Day	HDAY	No
Month	HMONTH	No
Year	HYEAR	No
Command 14, tag descriptor and date		
Engineering Units	HTDEU	No
Upper Transducer Limit	HTDURL	No

Lower Transducer Limit	HTDLRL	No
Minimum Span	HTDMINSPAN	No
Transducer Serial Number	HTDSN	No

Configuring HART Device Status tab - Channel block:

The following configuration information pertains to the Identification tab for all Series C I/O modules.All illustrations used in the procedure are for example purposes only

The following is an example of an AI Channel Block, Configuration form - HART Device Status tab.

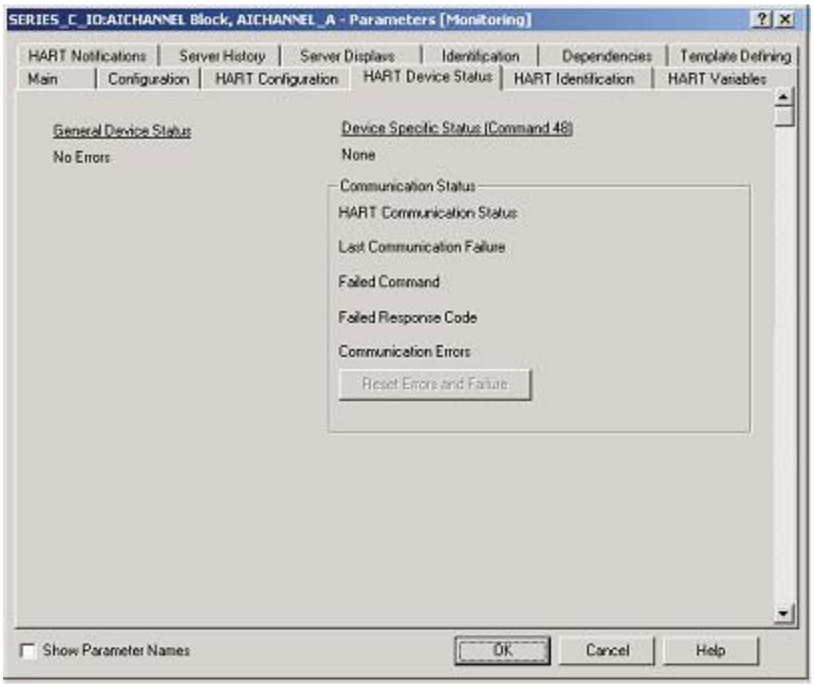


Figure 44: HART Device Status tab

The parameters of the HART Device Status tab are listed in the following table.

Table 39: HART Device Status tab parameters

Plain text	Parameter name	User configurable
Communication Status		
HART Communication Status	HCOMSTS	No
Last Communication Failure	HCOMFAIL	No

Failed Command	HCMDFAIL	No
Failed Response Code	HCMDRESP	No
Communication Errors	HNCOMERR	No

Configuring HART Identification tab - Channel block:

The following configuration information pertains to the Identification tab for all Series C I/O Analog Input andAnalog Output modules.

All illustrations used in the procedure are for example purposes only

The following is an example of an AI Channel Block, Configuration form - HART Identification tab.

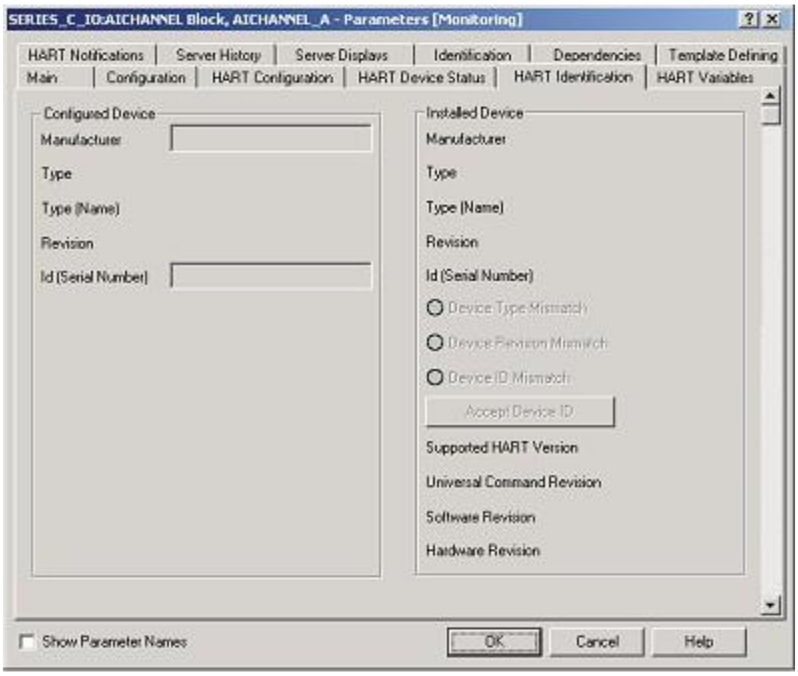


Figure 45: HART Identification tab

The parameters of the HART Identification tab are listed in the following table.

Table 40: HART Identification tab

Plain text	Parameter name	User configurable
Configured device		
Manufacturer	HDVMFGCD	No
Type	HDVTYPCD	No
Type (Name)	HDVTYPCDNAME	No
Revision	HDVRREVCD	No
Id (Serial number)	HDEVIDCD	No
Installed device		

Manufacturer	HDEVMFG	No
Type	HDEVTYP	No
Type (Name)	HDEVTYPNAME	No
Revision	HDEVRREV	No
Id (Serial number)	HDVID	No
Device Type Mismatch	HDEVMISM	No

Plain text	Parameter name	User configurable
Device Revision Mismatch	HREVMISM	No
Device ID Mismatch	HDEVIDFL	No
Accept Device ID	ACCEPTDEV	No
Supported HART Version	HARTVERSION	No
Universal Command Revision	HUCMREV	No
Software Revision	HSWREV	No
Hardware Revision	HHWREV	No

Prerequisites

- Control Builder is running
- A Series C I/O control module was created

Configuring HART Variables tab - Channel block:

The following configuration information pertains to the Identification tab for all Series C I/O Analog Input orAnalog Output modules.

All illustrations used in the procedure are for example purposes only

__The following is an example of an AI Channel Block, Configuration form - HART Variables tab.__

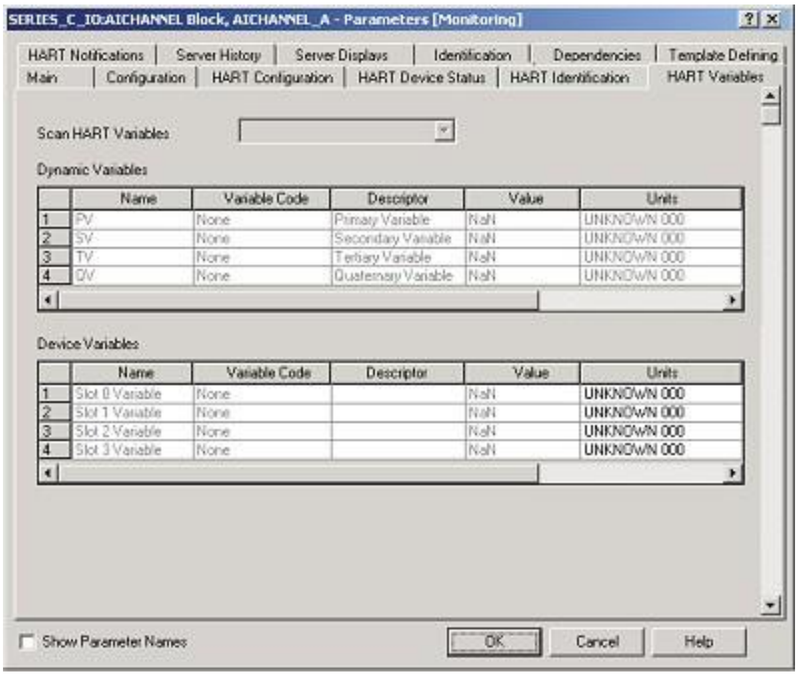


__Figure 46: HART Variables tab__

__The parameters of the HART Variables tab are listed in the following table.__

__Table 41: HART Variables tab parameters__

Plain text	Parameter name	User configurable
Scan HART Variables	HSCANCFG	Yes
Dynamic variables		
Name	HDYNNAME	Yes
Variable Code	HDYNDVC	Yes
Descriptor	HDYNDSC	Yes
Value	HDYNVAL	Yes
Units	HDYNEU	Yes
Device variables		
Name	HSLOTNAME	No
Variable Code	HSLOTDVC	Yes
Descriptor	HSLOTDSC	Yes
Value	HSLOTVAL	No
Units	HSLOTEU	No

Configuring HART Notifications tab - Channel block

After importing the Device Description and assigning the device type to the channel, you can modify the Command 48 string notification types of the channel from the Project view. However, after modifying thenotification types, you must user Load or Load values while Active option.

The following configuration information pertains to the Identification tab for all PM I/O Analog Input andAnalog Output modules.

All illustrations used in the procedure are for example purposes only

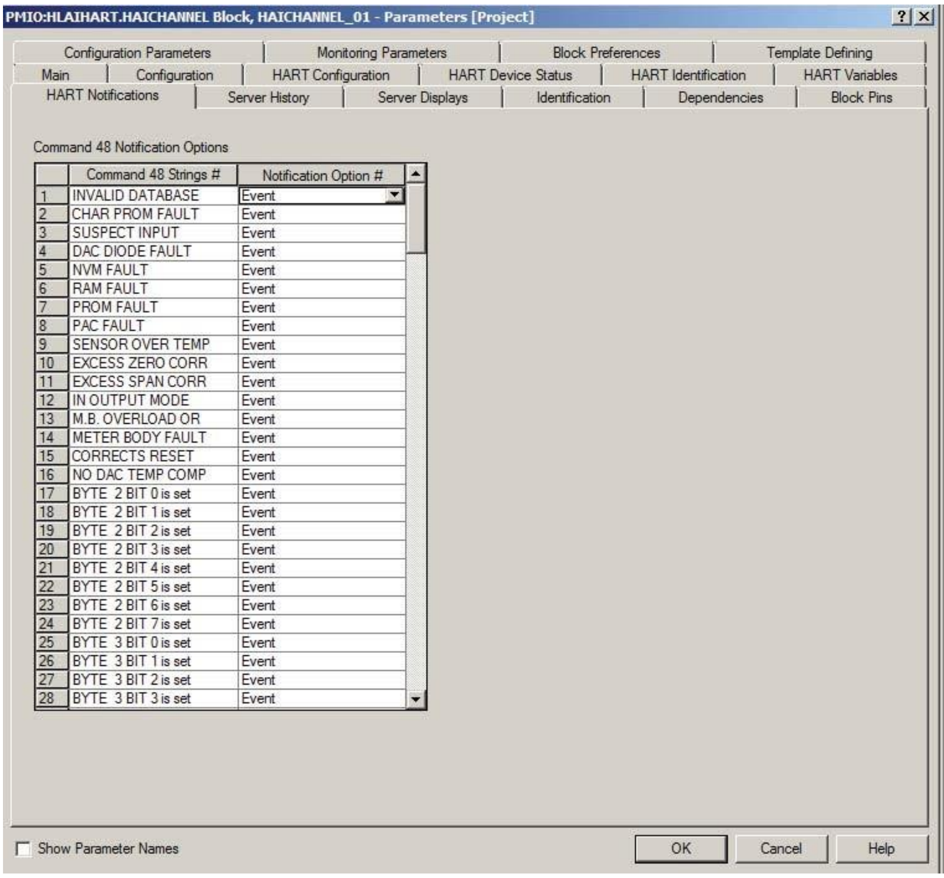


Figure 47: HART Notifications tab

The parameters of the HART Notifications tab are listed in the following table.

Table 42: HART Notifications tab parameters

Plain text	Parameter name	User configurable
Command 48 Strings	HCMD48STRINGS	No
Notification Option	HCMD48NOTIFY	Yes

For more information about modifying notification option, refer to HART I/O Implementation Guide.

Configuring Dependencies tab - Channel block

The following configuration information pertains to the Identification tab for all Series C I/O Analog Input andAnalog Output modules.

All illustrations used in the procedure are for example purposes only

The following is an example of an AO Channel block, Configuration form - Dependencies tab.

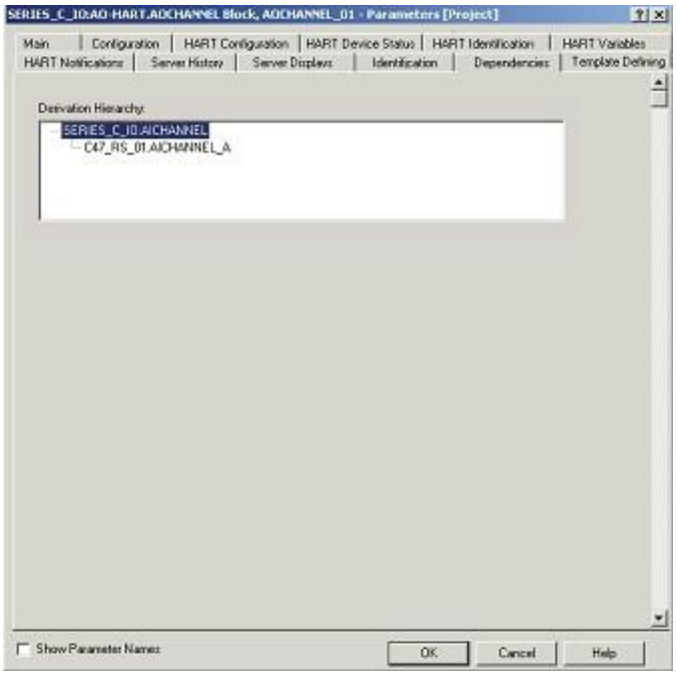


Figure 48: Dependencies tab

Configuring Template Defining tab - Channel block:

The following configuration information pertains to the Identification tab for all Series C I/O Analog Input andAnalog Output modules.

All illustrations used in the procedure are for example purposes only

The following is an example of an AO Channel Block, Configuration form - Template Defining tab.

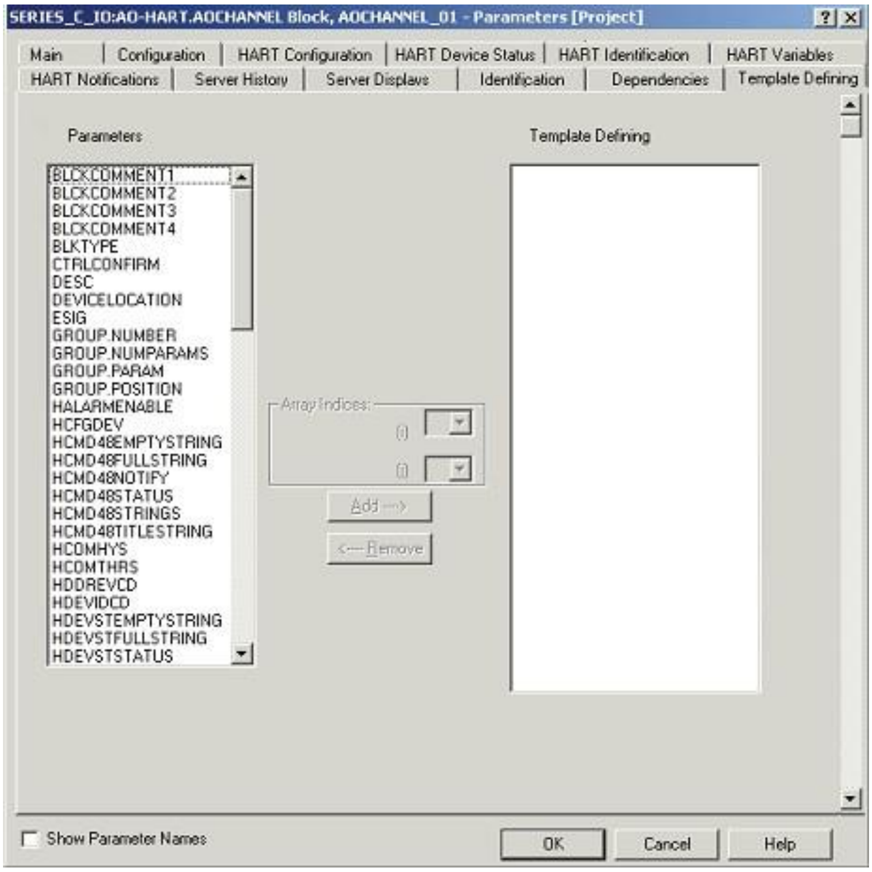


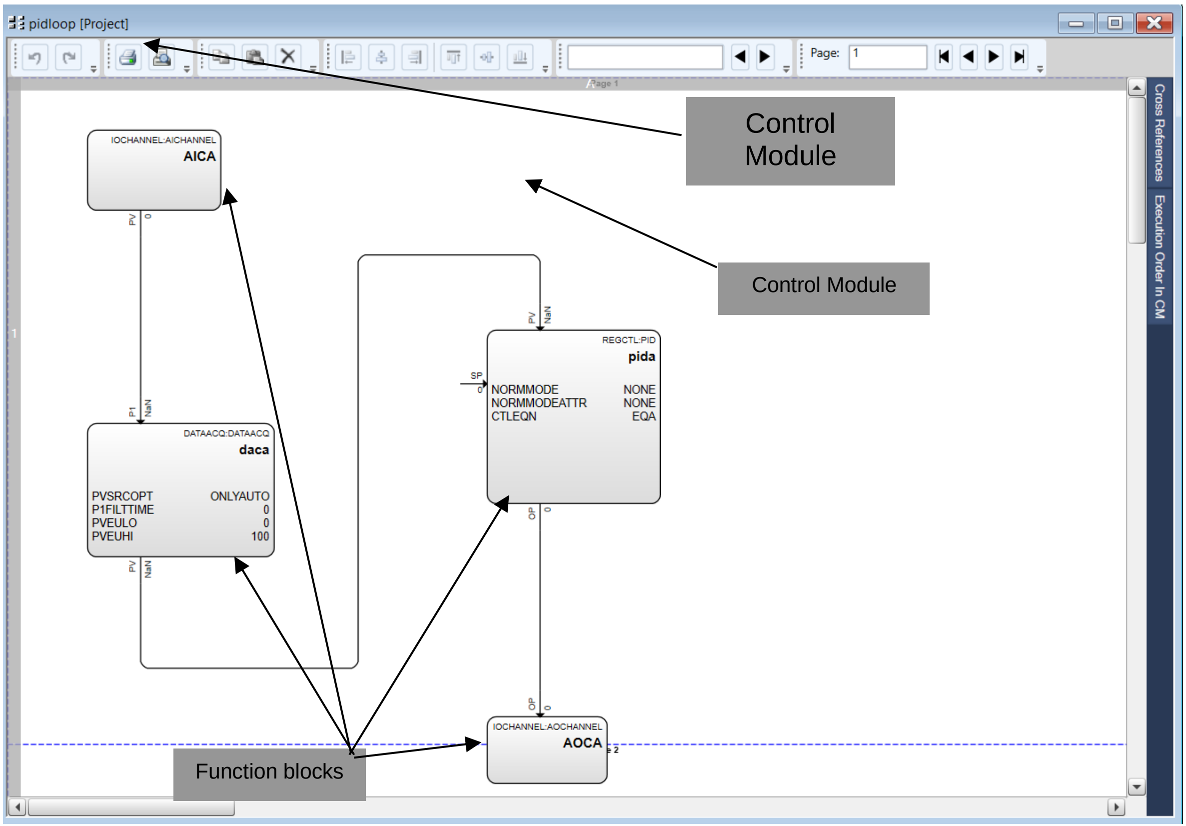
Figure 49: Template Defining tab

3.2 Control Module (CM)

A control module acts as a container for its component blocks referred as Functional Blocks. It holds continuous and discrete function blocks. So it can be used for building continuous and logic control scheme.

It provides these basic services for configured blocks:

- Serves as the unit of load for continuous and discrete control strategies.
- Transfers data between passive parameters that have no associated active connector.



typical Control Module (CM)

- Executes component function blocks in an es tablished order, which is configurable or arbitrarily determined by the CM.
- Provides independent tag names component blocks their parameters.
- Serves the execution master for continuous and discrete control strategies

3.2.1 Starting Control Builder

To properly start Control Builder, you must:

- start Configuration Studio, and
- Successfully login (with the appropriate security privileges) using established user-accounts.

Starting Configuration Studio to access Control Builder

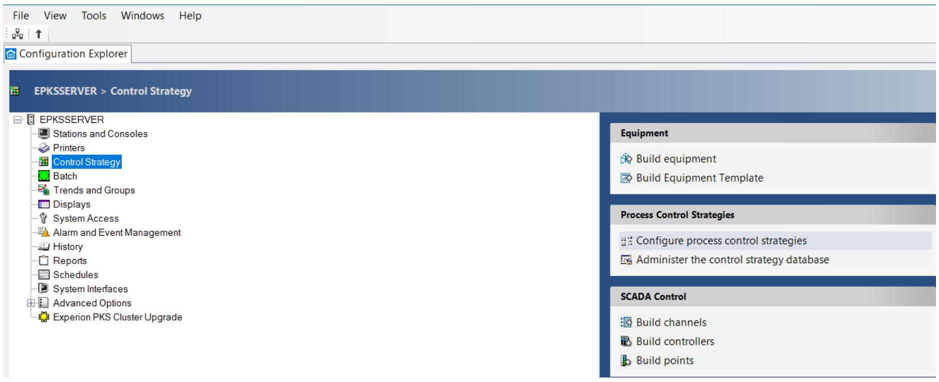
Configuration Studio houses the various server-based applications that are used to configure control strategies.

Control Builder is now accessed through Configuration Studio.

To start Configuration Studio and then start Control Builder

1. Launch Configuration Studio. The Connect dialog box appears.

2.Select a listed server and click the Connect button. Configuration Studio appears.



3. From the Configuration Explorer tab tree view, click Control Strategy. The Control Strategy selections appear.

4. From the Process Control Strategies grouping, click Configure process control strategies. Control Builder appears.

5. Go to the next section Login to Server, if user accounts have not been set. Must login to identify the user, the security level, and the Server that Control Builder is to connect to for this session.

Login to Server

The ability to access Control Builder through Configuration Studio is based on valid login account and security permissions. The security access for login integrates Windows user accounts with Station operator based accounts. Please review the information in the Configuring Security and Access section of the Server and Client Configuration Guide before attempting to login to Control Builder.

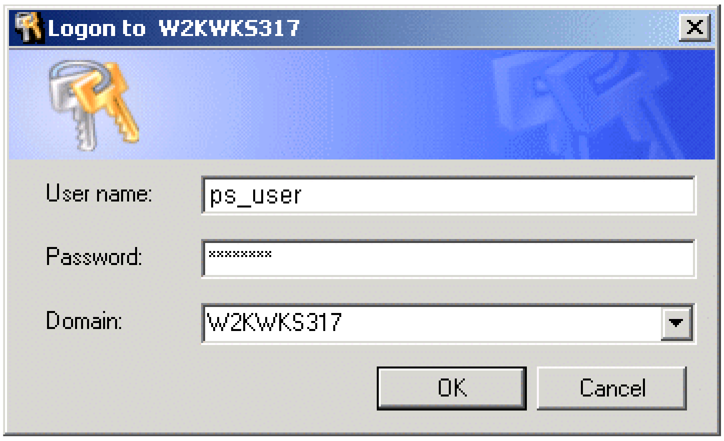
Configuring Windows user account in Station

The following procedure sets up a Windows user account through Station.

1.Do you want on enable the Single Sign On function?

- If the answer is Yes, go to Step 2.
- If the answer is No, go to Step 4.

2. Click the Enable Single Sign On check box. Check mark appears in the check box, single sign on function is enabled, and only the Server Name field is accessible . Login is based on current Windows user account logon to the named Server and the login dialog does not appear for subsequent logins.



3. Click OK. Control Builder appears. Go to Step 8.

4. The default User Name is usually the one used for the Windows account logon. For example, ps_user. The Station operator based default name is mngr and it is case sensitive. Please key in your assigned user name, as applicable. Identifies user and associated security level.

5. Click and key in your login password in the Password field. You need an access level of at least Engineer to create a control strategy. Please check with your system administrator to get your assigned password, if required. The password is tied to your Windows account logon or the operator-based security for the Station application. The default password for operator-based security is mngr1, and it is case sensitive. Confirms the identity of the user and associated security level.

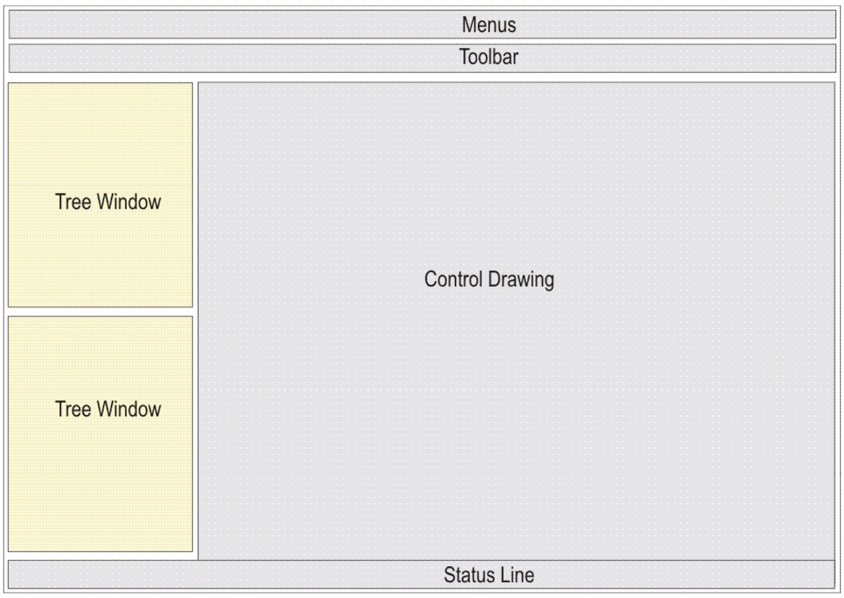
6. The Domain Name field identifies the domain where the Server is located. For Station operator-based security logon, the Domain Name must be Experion Security. Key in desired domain name or click the down-arrow button to select it from the list. Identifies the Domain associated with the Server.

7. Click OK. Control Builder appears.

8. Go to the next section Layout of the main window. Must open tree windows to view database and library contents.

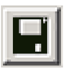
Layout of the main window

The following figure shows the layout of Control Builder's main window.

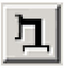
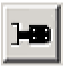
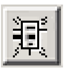


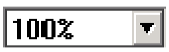
Control Builder Toolbar

The toolbar provides quick access to commonly used commands.

Button	Description
	Open Tree. Open a new tree window in Control Builder.
	Close. Closes the tree window that has focus in Control Builder.
	Open Library
	Left
	Right
	Save. Saves the item.
	Delete. Deletes the item that has focus.
	Copy. Copies the item to the clipboard.
	Paste. Pastes the item from the clipboard.

Button	Description
	Print. Prints the item that has focus.
	About

	Help. Displays the Help for Control Builder.
	Point Selection
	Wire
	Param Connector
	Execution Environment Assignment
	Module Containment
	Load
	UpLoad
	Substitute Names. Activates the dialog box that allows the assigning of substitute names for blocks.
	QVCS Manager. Activates QVCS Manager.
	Toggle State. Toggles the selected item to the opposite state, such as: inactive to active or active to inactive.
	SCM Navigation

Button	Description
	Fieldbus Device Description Input
	Change Scale. Changes the scale of the item that has focus.

Control Drawing

The main work area of Control Builder where:

- Control Modules are created
- Function blocks are inserted and connected
- Control strategies are initiated

Control Builder Status Line

The Status Line indicates a number of status properties of the Control Builder application, or a prompt for action.

At the left side of the status bar are prompts to help you to access features in Control Builder. At the right are four boxes that show various properties on status and connections to servers and their databases, which aredescribed in the following table.

Status Line box (from left to right)	**Description**	
ERDB Synchronization status	Indicates the synchronization status of Control Builder with the ERDB. Valid values are:	
	Indication	**Color**
	<blank>	Grey
	SYNC	Green
	NOSYNC	Reverse
	SYNCERR	video Red
Connection to Server ERDB	Indicates whether Control Builder is connected to the **PRIMARY** (ServerB) or **BACKUP** (ServerA) ERDB. You will not be able to build control strategies and perform most downloads when connected to the BACKUP server.	

Server	Shows the Server to which Control Builder is 'logged in.' (Derived from Configuration Studio.) By default, it also indicates the CDA Server which Control Builder is using.
Security Level Access	Indicates the user level access to which Control Builder is 'logged in.' For example, **ENGR** or **MNGR**.

Opening and navigating a tree window

This procedure illustrates how If this is the first time Control Builder has been launched, there are no tree windows open. Otherwise, Control Builder opens with the same tree windows that were displayed when it was last closed.

There are three available trees or views within each tree window:

- Project tree
- Monitoring tree
- Library tree to initialize the tree windows within Control Builder.

Prerequisites

- Control Builder is running

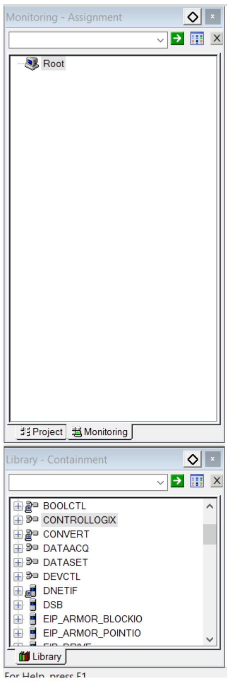
To open a tree window

1. Click View > Project/Monitor Tree, OR Library Tree OR click the Open Tree button in the toolbar.

(This option will not be available, if two tree windows are already open.)

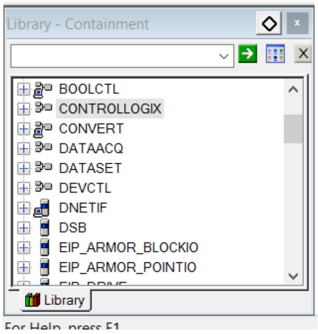
Project/Monitor tree OR Library Tree window opens on the main Control Builder window.

2. Repeat Step 1 to open another tree window, if required. (This option will not be available, if two tree windows are already open.) A second tree window opens, as shown in the following figure.



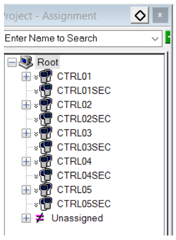
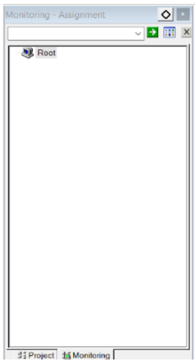
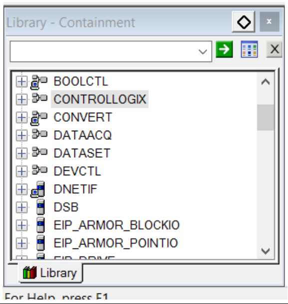
3. Click the appropriate tab (Project, Monitoring or Library) on the bottom of the tree window to select the desired tree view. Selected tree is displayed.

4. Click on the ⊞ sign of the desired Library to expand and view the contents.



About Tree Window tabs

The Tree Window provides quick access to three tabs: Project, Monitoring, and Library.

Tab	Description
	Project tab. The Project Tree View allows instances or strategies to be viewed as trees, showing the assignment relationships in a window in the Control Builder.
	Monitoring tab. The Monitoring Tree View displays instances or strategies that have been downloaded in the system and are live.
	Library tab The Library Tree View includes a variety of specific libraries that contains unique functions specific to that library

Moving a docked tree window

- Click and drag the title bar of the appropriate tree window to another border of the main Control Builder window. Tree window assumes the desired position. For example, if you drag the tree window near the right edge, it will "do

Closing a tree window

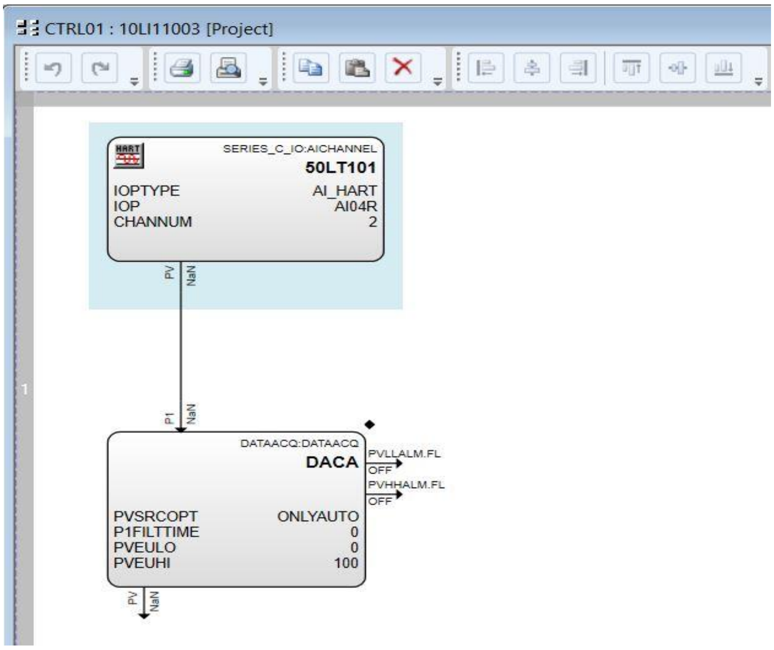
1. Select either tree window by clicking on one of the tree window tabs or the title bar. Window is highlighted.

2. Click File > Close, or you can click the < Close tree button in the toolbar.

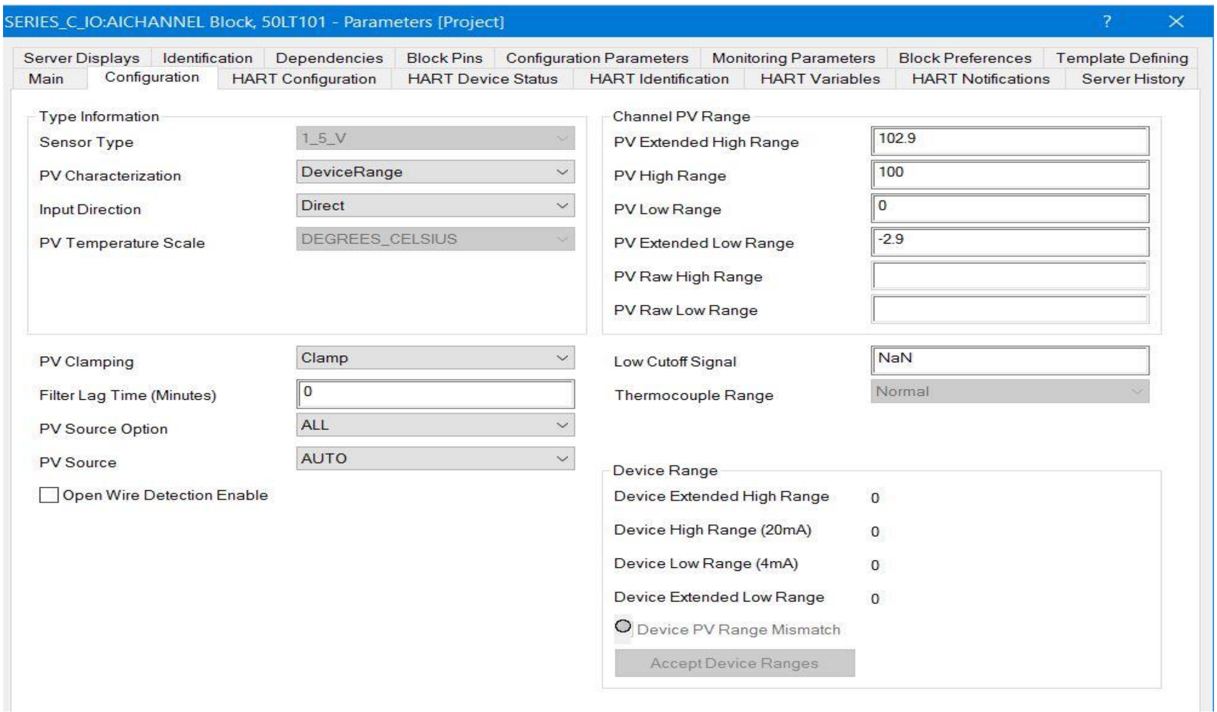
The selected tree window closes.ck" on the right-hand side.

3.2.2 I/O forcing environment

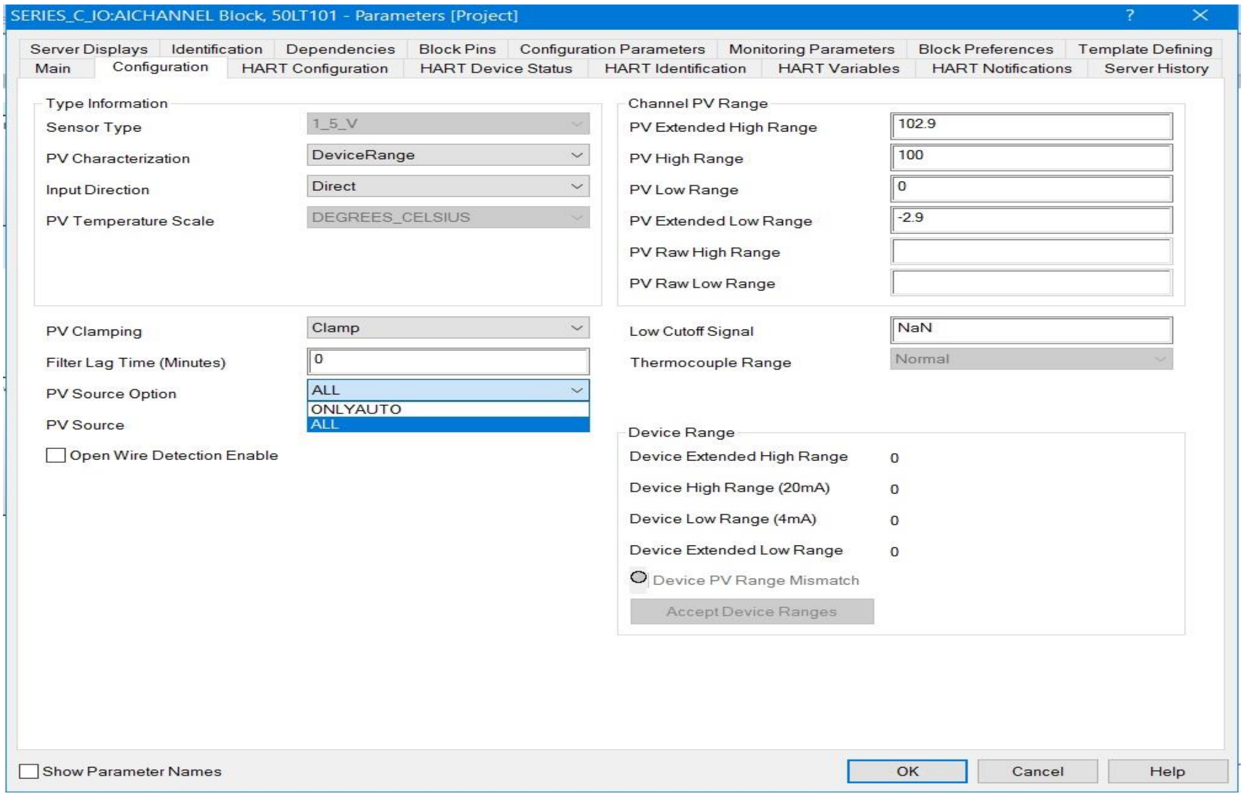
In the control builder environment, after opening the corresponding module control,



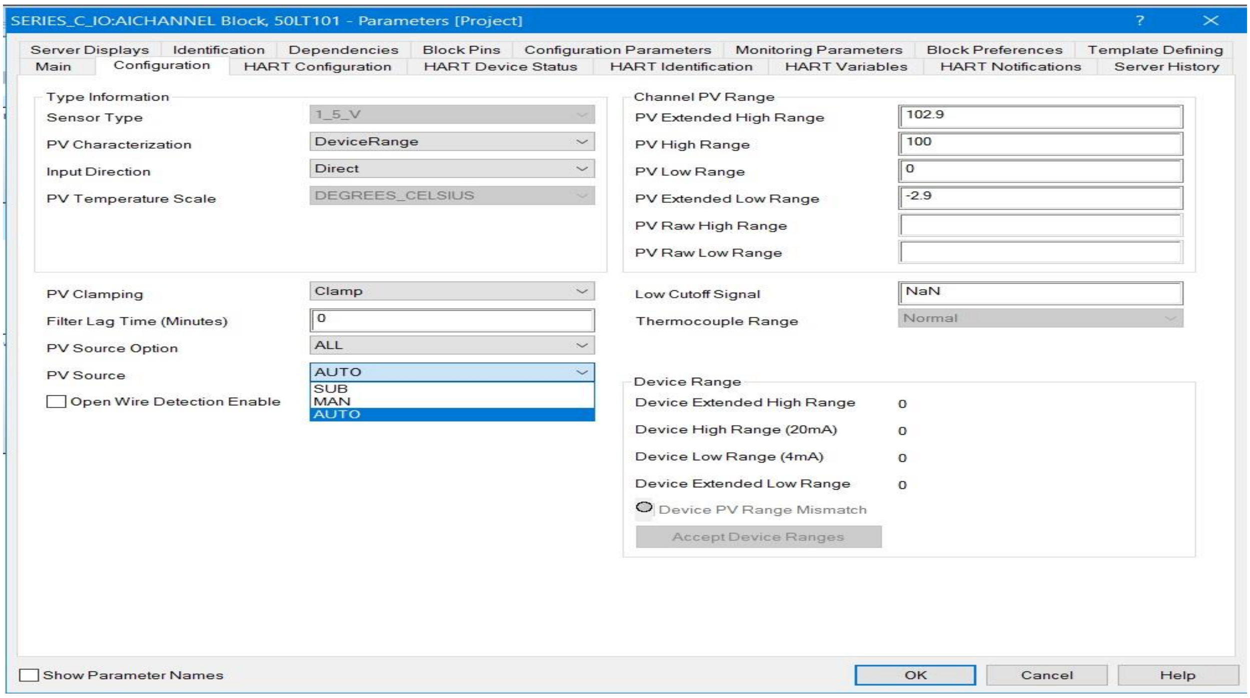
we will first click twice on the corresponding channel that needs to be forced.



Next, in the configuration section, we first set pv source option to ALL mode.



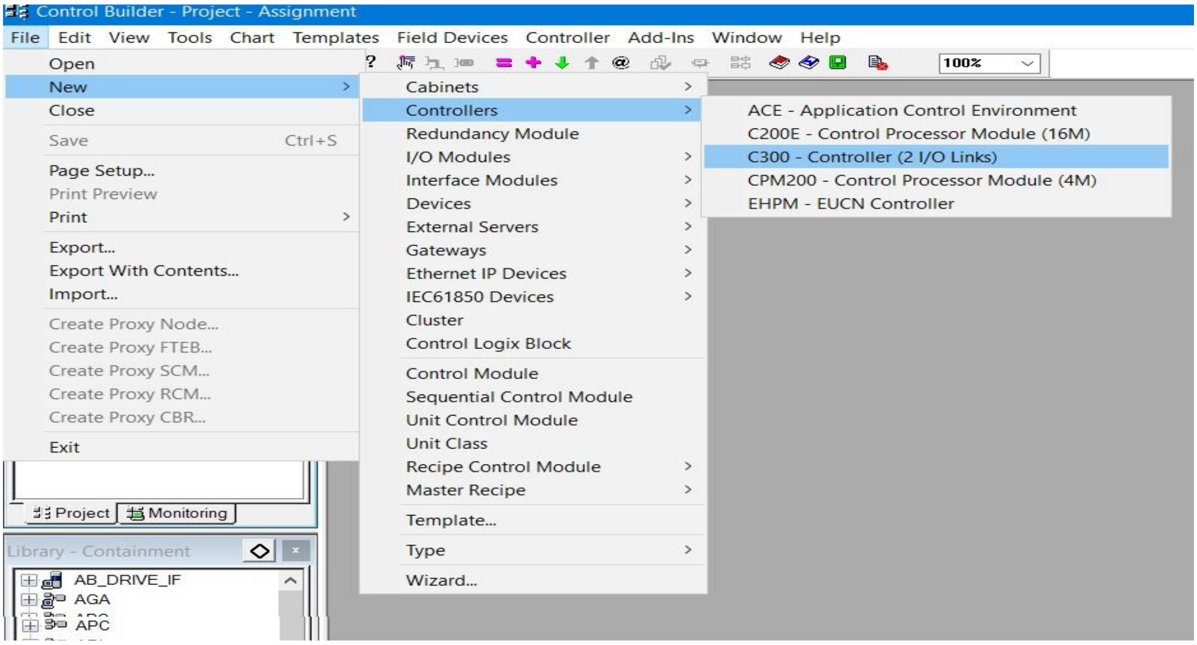
Then we will change PV source from AUTO mode to MAN mode



Now you can change its value by clicking on the PV or PVFL pin of this channel.

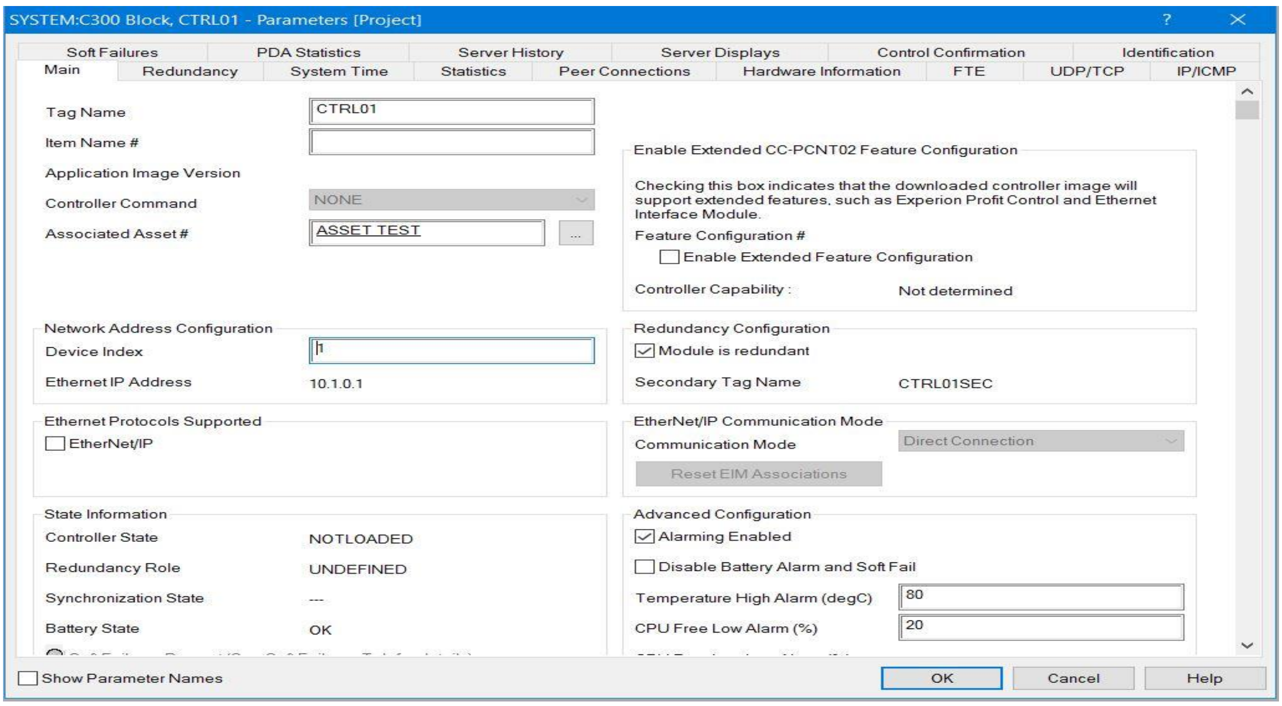
3.2.3 CPU, I/O module, etc. assignment

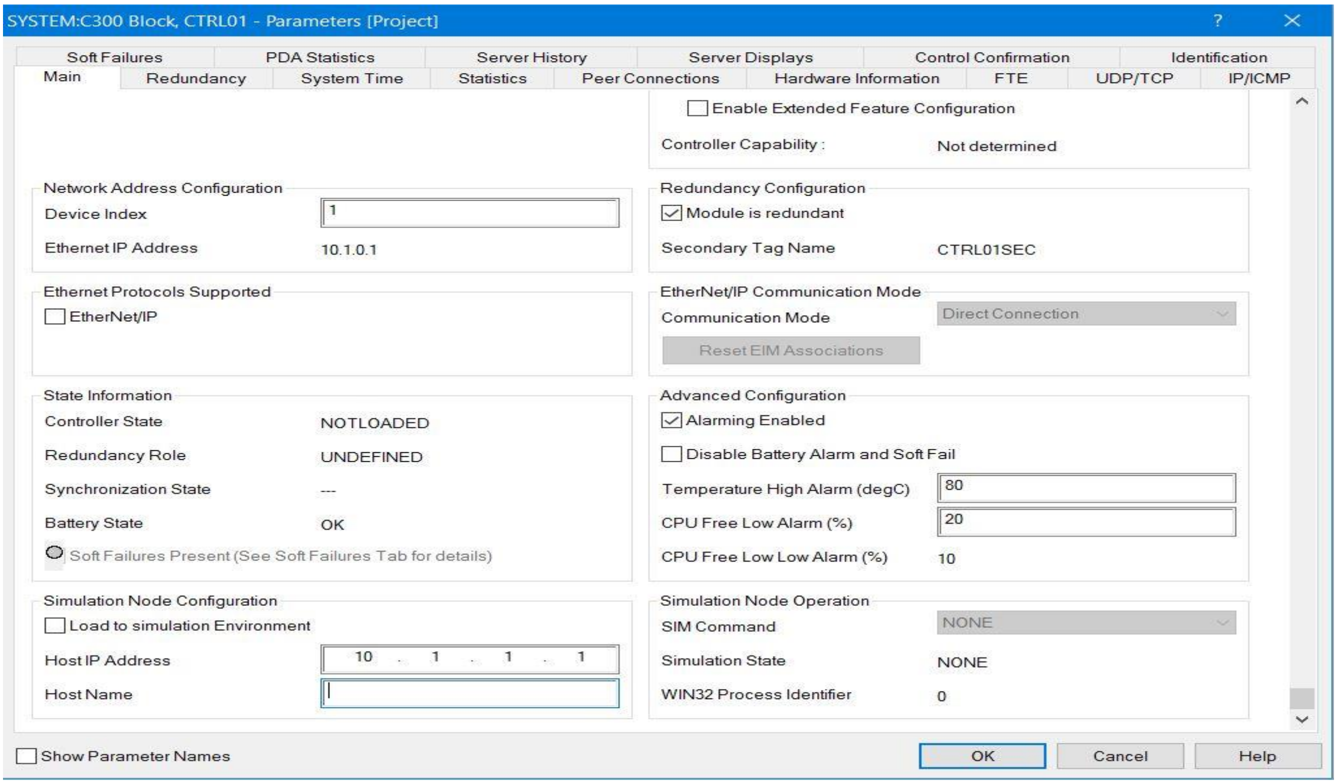
As shown in the picture, after opening the CONTROL BUILDER, by clicking on the fly, first select NEW, then we will consider the controller and C300.



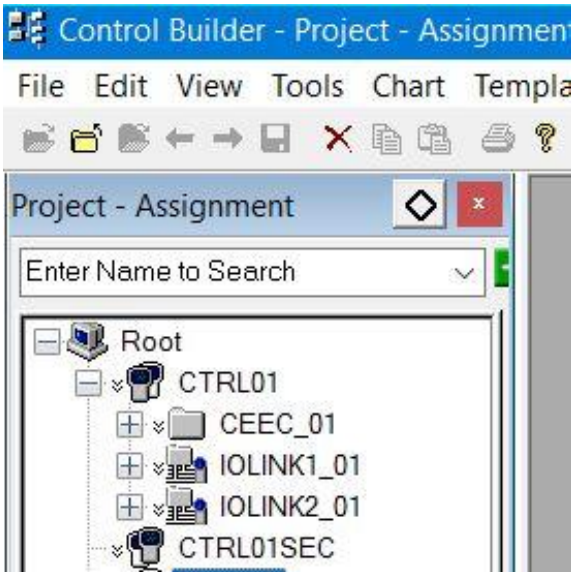
By clicking on C300, the following page will open.

Next, we can enter the controller's specifications, including the name, whether it is single or redundant, the IP used, etc., in the opened page.

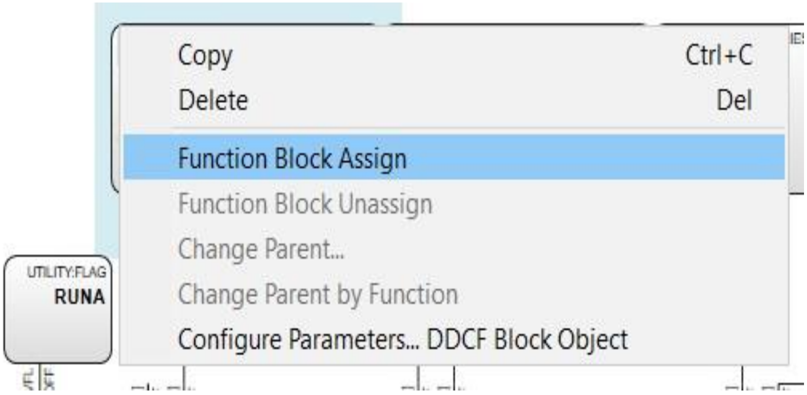




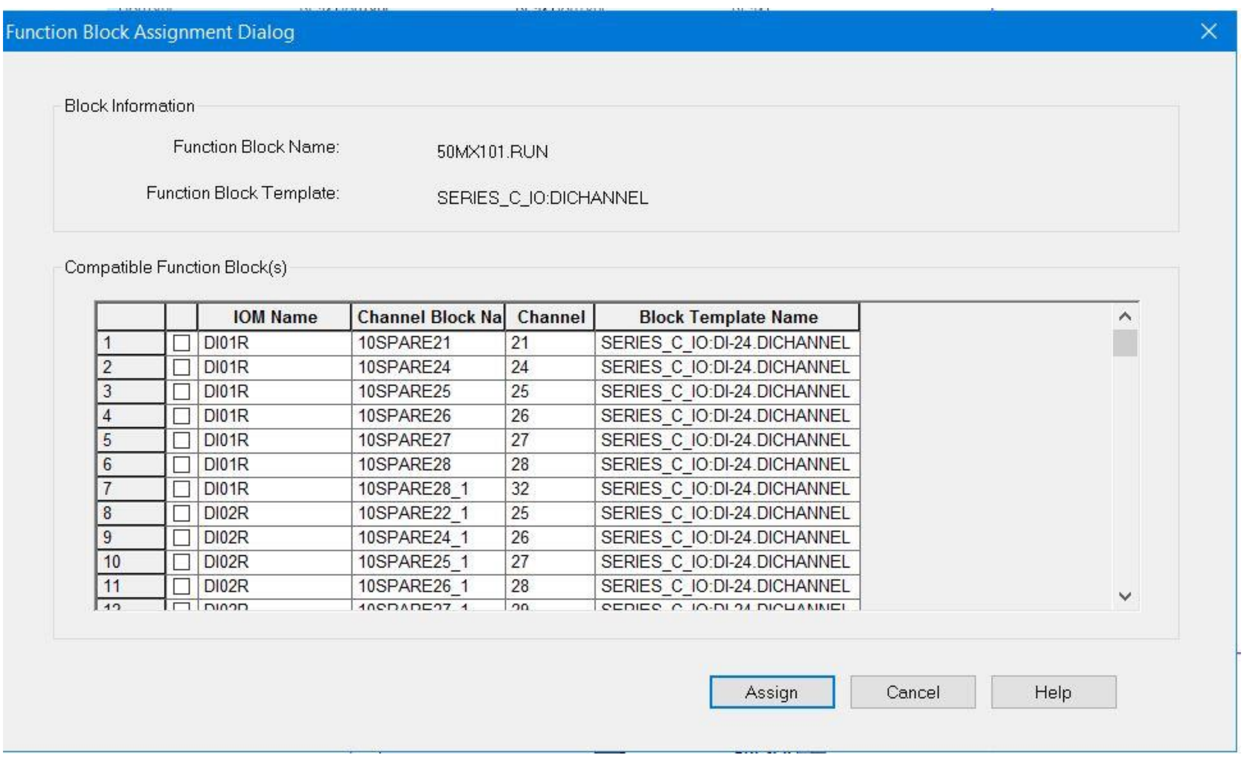
By clicking on the OK option, the screen will be closed and the controller will be shown in CONTROL BUILDER as shown below.



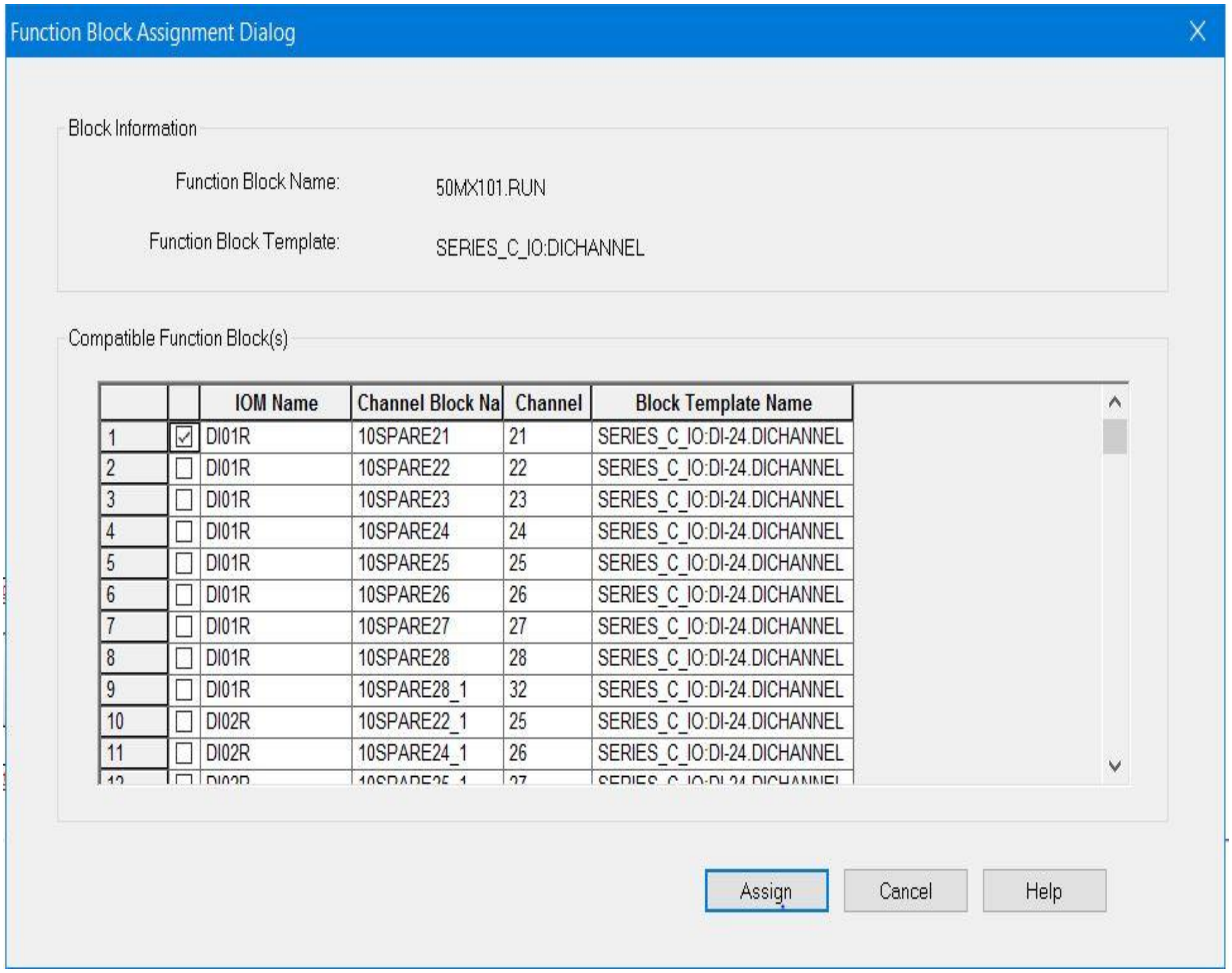
We will do the same for assigning signals.
By opening the desired CM, we will right-click on the channel that needs Assign.



By selecting Function Block Assign, the following image will appear.



Select the card to which this signal should be assigned and by selecting the channel of this signal, we will click the Assign option. Now the assigned desired signal will be considered.



4 C300 Controller requirement

4.1 start up

The C300 Controller module executes a series of tasks automatically when power is applied to the controller module. These tasks include:

- Power-On Self Test (POST) to verify the presence and integrity of the controller module hardware.
- Initialization of the hardware and software environment
- Determination of whether to transition to the application image, if present.

C300 Controller Startup and Power On Self Test routine

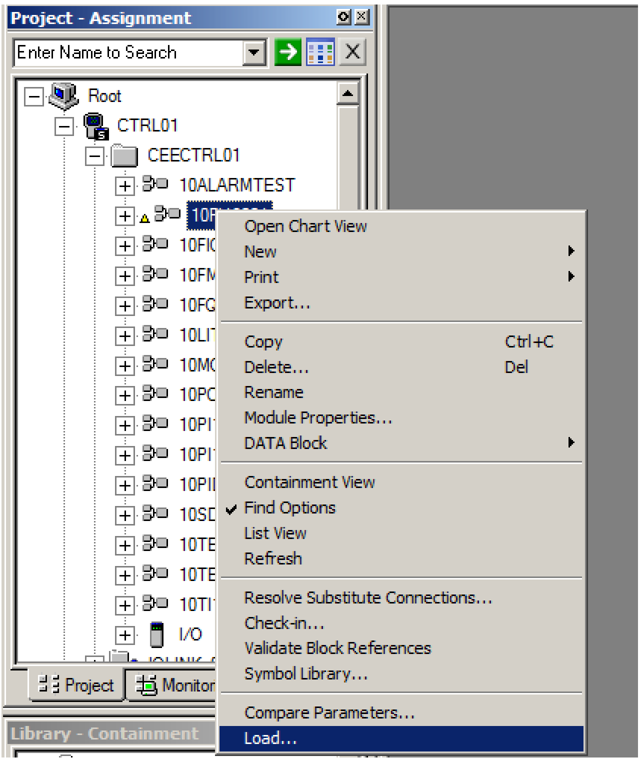
Stage	Description
1	The C300 Controller Power LED lights upon insertion of the module onto a powered IOTA board.
2	The C300 boots up using firmware installed in the flash memory. (Initial firmware boot image is installed in the factory.)
3	A Power-On Self Test (POST) is performed as part of the C300 Controller start up. Controller display shows Tnnn, where nnn indicates the number of the test currently being executed during POST.)
4	The Status LED briefly shows all of its colors (Red, Green, Orange), and then remains solid Red until POST completes.
5	The four-character display walks through a series of horizontal and vertical bars, as well as several brightness levels. Once completed successfully, the display indicates the progress of the POST. See C300 Faceplate indicators/displays for a complete listing and description of the controller's status indications and displays.
6	Progress of the POST is shown on the C300's faceplate display. If a fault occurs during POST, the test halts and the display shows the test number (Tnnn) associated with the detected fault. Corrective Action: If a fault occurs and halts the POST, then reset the controller and allow the POST to execute once more. If the fault persists, then replace the failed controller module.
7	If no faults occur, POST continues to execute. Upon completion of POST, the C300 determines whether to transition to boot mode or application mode. If a valid application image is present in the C300, then it is started in application mode, unless the controller has been commanded to shutdown, or is rebooted into the FAIL state. The Status LED indicates the state of the controller's associated hardware.
8	The controller indicates '-bp-' on its display while it waits for a response from the BootP server service. The BootP server supplies the C300 its IP Address assignment. If no response is received from the BootP server in 2 minutes, the C300 will timeout.

	Once the C300 receives a response from the BootP server, '-TS-' is indicated on the display while it waits for a response from the configured time source. If the configured time source is not available, the controller will attempt to connect with an alternate time source. See Time management in the C300 Controller for more information.
9	Action: Verify the correct Device Index is shown on the display (#nnn) and that there are no address errors. Address errors - The C300 determines if any other module in the FTE network is using the same Device Index. If another module is discovered with the same Device Index, the C300 will not join the FTE network but instead will enter the "no address" state and wait for a new Device Index to be set. If another node is discovered with the same IP Address, the controller will not join the FTE network but instead will enter the "dup address" state and wait for a new Device Index to be set. Action: If the Device Index is invalid, or any address error is displayed, see Reset Device Index and IP address of a controller and follow the procedure to reassign the Device Index and IP addresses Once a valid IP Address is assigned, a redundant C300 negotiates its redundancy role (primary/secondary) with its partner module. See Stage 10.
10	The C300 alternately displays its Device Index, redundancy role and execution state on the controller's faceplate display. Redundancy role - Question marks '????' may appear on the display to indicate that the C300's redundancy role has not been determined. For non-redundant controllers, the redundancy role is shown as 'nrdn'. Execution state - If no EE databases have been retained in a primary or non-redundant controller, the controller enters the No Database 'NODB' execution state. If the controller's database has been retained, the execution state is 'IDLE.' Secondary controller first displays '????' until its redundancy role is determined. Once determined the secondary controller indicates it is an unsynchronized secondary or synchronized secondary. The secondary controller also shows the controller's execution state, which is 'BKUP.'

4.2 Download

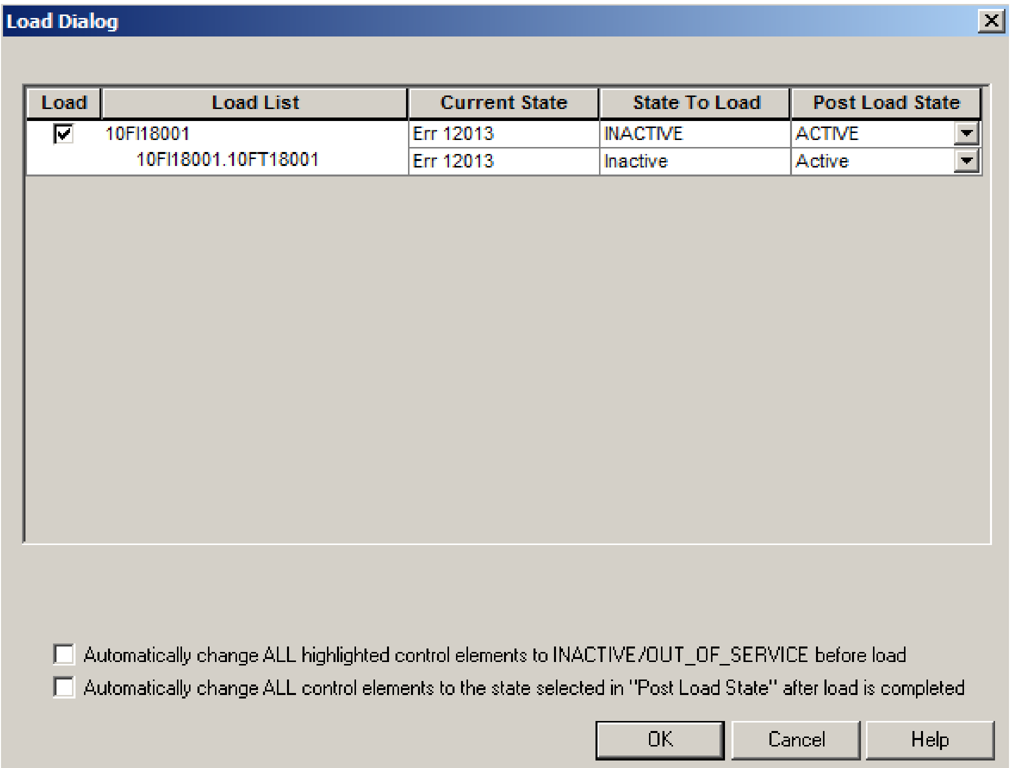
After create a Control Module (such as: pump, motor, indicator, etc.) can be downloaded when the controller and cards are in monitoring mode and on. In Control Module, each hardware channel must be assigned to the desired card. After creating the control module, it should be placed under the desired controller (CEE).

After making sure of these things, in project mode, right click on the control of the module we want to download and click on the load option.

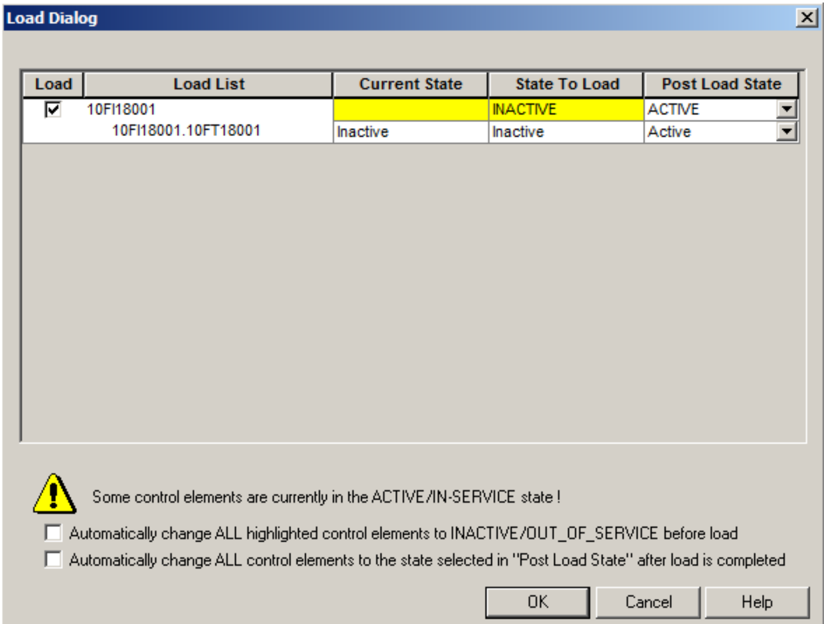


At the same time, make sure that the desired module is assigned to the desired HMI screen, otherwise the module control will not be correctly placed in the monitoring mode after downloading.

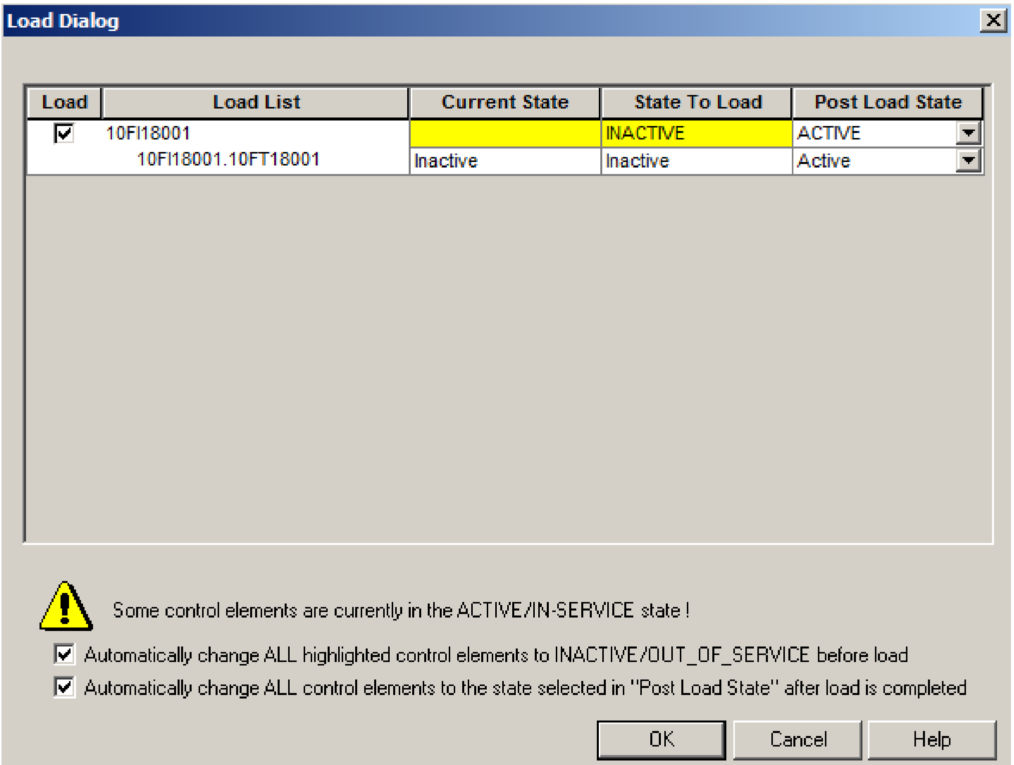
After selecting the load option, if the control module has already been downloaded and needs to be downloaded again, we will see the following figure.

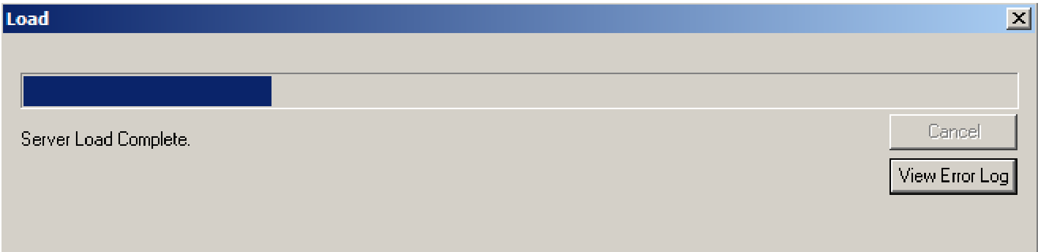


Wait for the errors in the current state column to be resolved,

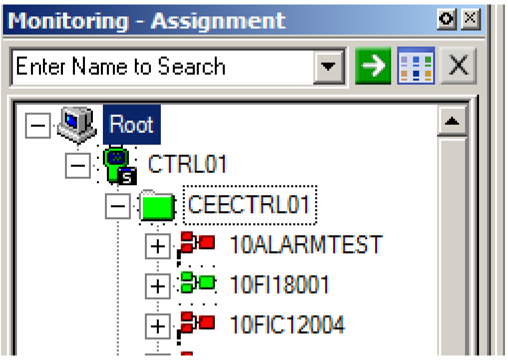


then activate the two check boxes after that click OK ,





now the control module is fully downloaded.

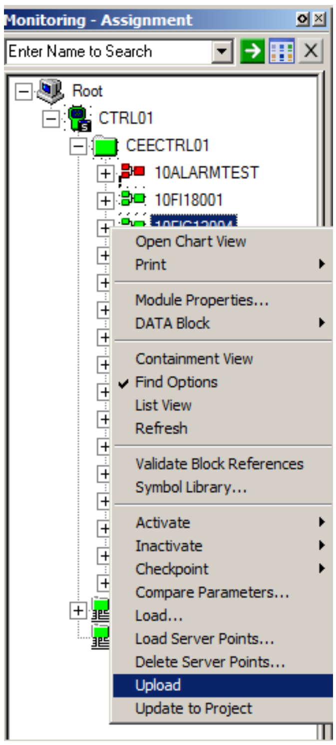


If a control module has been downloaded correctly, it will be greencolor ,  otherwise it will be red color . 

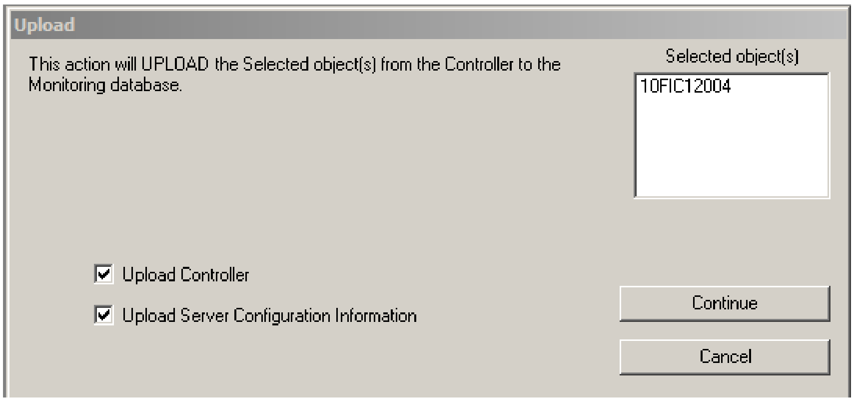
4.3 On line

To make any changes (like changing PID values) in the logic of the system in the monitoring mode or online (who is the operator , engineer, manager, etc.), there is a need to consider those changes in the project mode or offline.

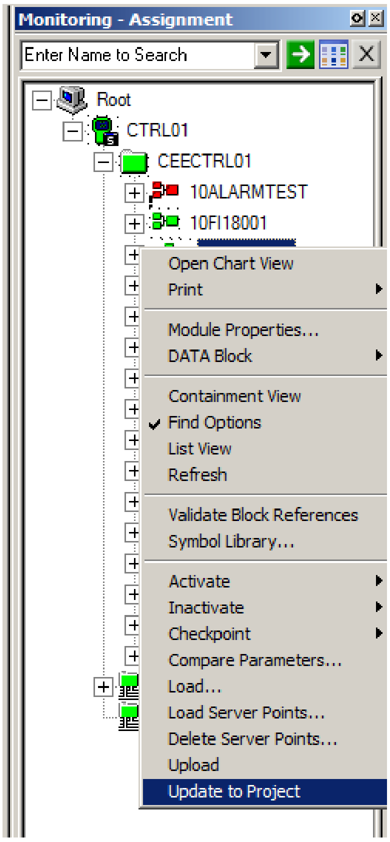
To do this, first right click on the desired control module's in monitoring mode, select the upload option,



after the window opens, select the continue option.



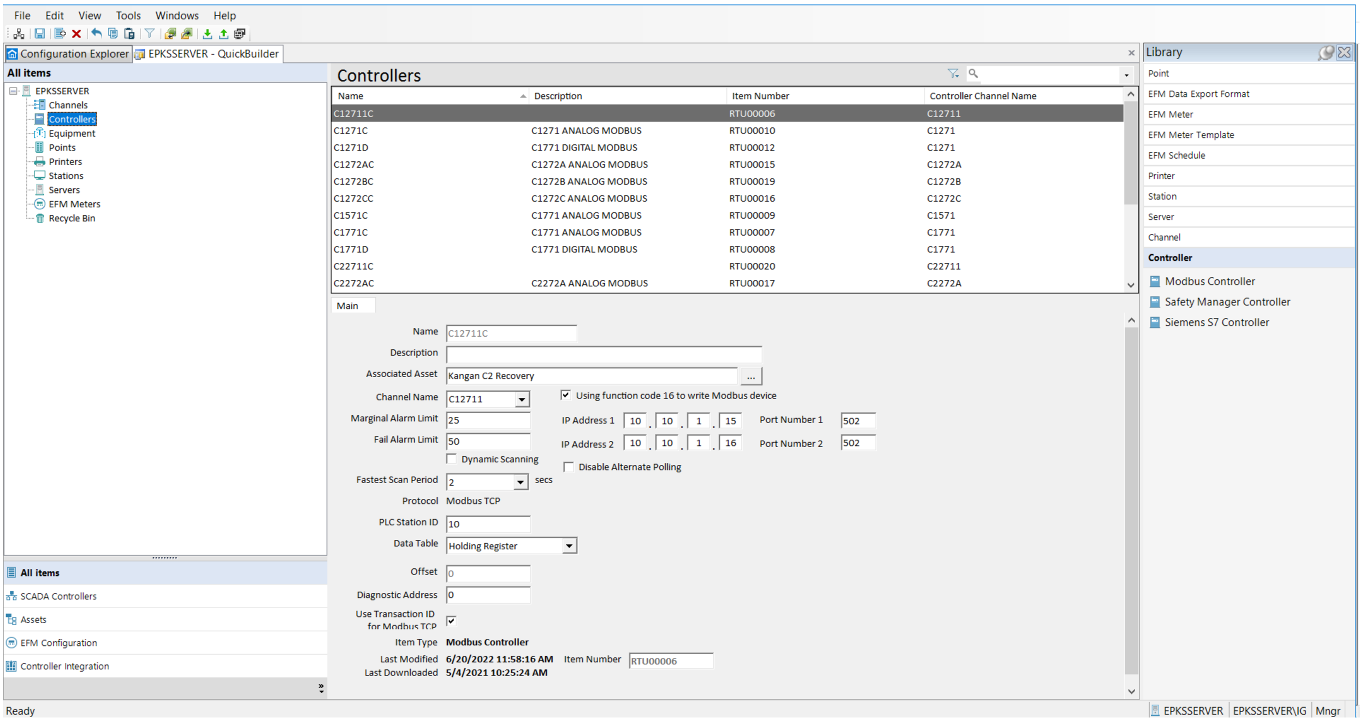
Then right click again on the mentioned control module and select the update to project option,



after completing the step we will have everything we had in the online mode also in the offline mode.

5 Quick builder

Quick Builder is a graphical tool for configuring Flex Stations, printers, controllers (other than Process Controllers) and standard points. (If you have Process Controllers, you configure them with Control Builder. For details, see the Control Building Users' Guide.)
After building hardware and points with Quick Builder, you download these items to the server database. When you download the data in Quick Builder-or part of it-to the server, it becomes part of the configuration database. (The configuration database defines how each component in your system is configured.) If necessary, you can update the data and repeat the download process. Alternatively, you can upload data from the configuration database into Quick Builder, edit it and then download it back to the server.
The procedures you use to build items with Quick Builder are documented in the Quick Builder Guide.



5.1 Station displays

Station displays for points

As part of points configuration you can:

- Associate a display with each point
- Group related points to form:
- Groups
- Trends

For any point in your system, you can also specify:

- A user-defined Point Detail display, instead of the standard Point Detail display
- A user-defined Group Faceplate Template display, instead of the standard Group Faceplate Template display

Associated displays

If you define an associated display for a point, operators can select a point (for example, on a custom display or in the Alarm Summary) and call up the Associated display (by pressing F2 or clicking the associated display button on the toolbar); the display that has been defined as the associated display for that point is then called up.
You can define an associated display for a point by using the following methods:

- Using Quick Builder or Control Builder to specify the associated display

- Using the Point Detail display on Station to specify the associated display

Groups and trends

You can configure up to 16,000 groups and up to 3,000 trends.

Groups enable users to view the current operating data (or configuration data) for up to eight points of any type on a single group display.

Trends enable users to view trend information for up to eight point parameters.

You can use Quick Builder or Control Builder to configure points to form groups or trends or you can assign a point to a group or trend in Station. If you assign a point to a group or trend in Station, this is not reflected in the point definitions in Quick Builder or Control Builder unless the point definition is uploaded to Quick Builder or uploaded to Control Builder. Consequently if you download points from Quick Builder or Control Builder, changes to trends and groups that have not been uploaded are overwritten.

User-defined point detail displays

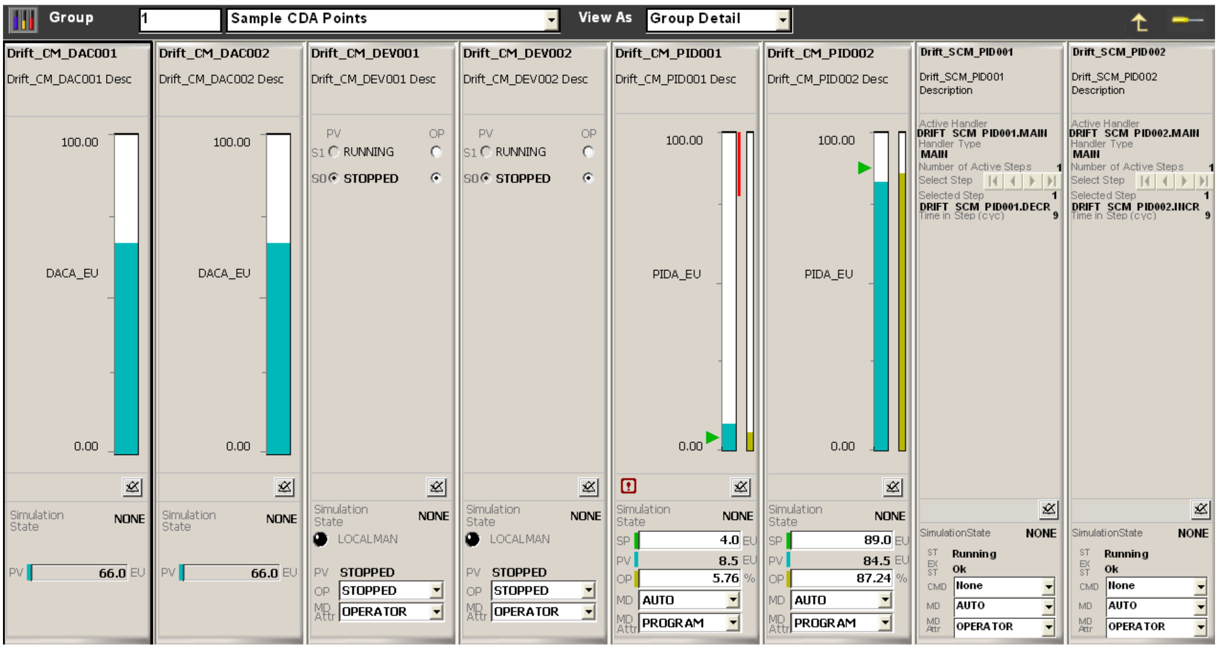
You are supplied with a set of predefined displays that are used to show point detail data.

You can create custom Point Detail displays using Display Builder and then use Quick Builder to specify the custom Point Detail display for particular points.

Group faceplate templates

You are supplied with a set of predefined 'faceplates' or templates that are used for status, analog, or accumulator points in a Group Detail display.

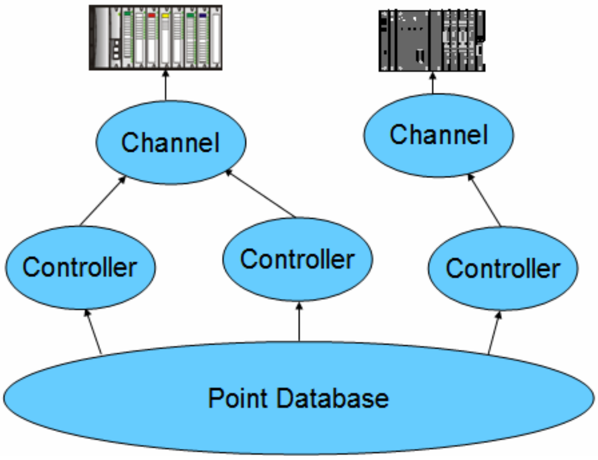
For example, the following figure shows a Group Detail display for a group consisting of various types of points using the standard faceplates for the point type.



You can create custom Group Faceplate Template displays using Display Builder and then specify the template for particular points when you build the point.

5.2 SCADA Points

Points its naming shall always refer to P & ID and IO schedule.



SCADA Control

Build channels

You build SCADA channels using Quick Builder in Configuration Studio.

Channels are the communications link between controllers and the Experion server. There is a specific reference guide for each controller type that describes the settings applicable to your controller type.

For more information about building channels, see the Quick Builder Guide and the controller reference for your controller type.

Build controllers

You build SCADA controllers using Quick Builder in Configuration Studio.

Controllers are devices used to monitor and control processes and items of field equipment. A controller is linked to the Experion server by a channel. There is a specific reference guide for each controller type that describes the settings applicable to your controller type.

For more information about building controllers, see the Quick Builder Guide and the controller reference for your controller type.

Build points

You build standard (SCADA) points using Quick Builder in Configuration Studio.

Experion uses points to store information about field values or devices, for example, the state of a pump, a temperature sensor, or the process variable of a control loop.

For more information about points and point parameters, see the 'Understanding and configuring points' section in the Server and Client Configuration Guide. For more information about building standard points in Quick Builder, see the Quick Builder Guide.

Defining points that reference signals

The **bbpnt** utility converts a Bristol Babcock ACO file to a server point build file. This utility generates points that address signals in the load.

Before assigning a point reference number (that is, ANAnnn, STAnnn), bbpnt checks to see if the point name already exists in the server database. If it does exist, the same point reference number is used. If it does not exist, the next unused point reference number is used.

It is recommended that you maintain a strict naming convention for ACO and point source files.

The syntax for this utility is:

bbpnt acofile [options]

where acofile The ACCOL ACO file from which the points will be extracted. The ACO extension is not required.

Options :

Option	Description
-o *pntfile*	Output point source file. The default is **acofile.pnt**. The .pnt extension does not need to be specified.
-l	Ignore local/global flag and extract all ACCOL signals form the ACO file.
-r *rtunum*	Controller number to which the points are assigned. The default is 1.
-m *string*	Mask string. String containing up to twenty 0s and 1s indicating which positions in the signal name should be copied to the point name. Positions not specified in the mask string will be included in point names. The default will be all characters in signal name to be included up to the maximum point name length.
-s *hrnum*	Starting character position in the signal name as stored in the ACO file for the point name. The 16 character point name is extracted from the signal name starting at this position. Defaults to the first character position, 1.
-a *code*	Area code. A 2-character code specifying the area to which the points will be added. The default is no area code.
-p *pntnum*	Point number. Starting point number for all types of points. The default is 1.
-v	Verbose. Print out extra information for debugging.

Applicable analog signals in the ACO file will produce output for the point definition file. What is produced in the point definition file depends on the contents of the ACO file. For example, A1SOURCE and A1DESTIN entries are only produced for analog alarm signals that have one of their alarm limits assigned to another signal. As much relevant information as can be extracted form the ACO file will be put into the point definition file.

Applicable logical signals in the ACO file will produce the following output for the point definition file.

5.3 Customizing point source files

Point source files generated by **bbpnt** can be customized using a text editor if required. The next step is to import the point build files into Quick Builder. See Quick Builder's help for detail on importing point files. After the points have been imported into Quick Builder, they can be downloaded to the server.

Importing the point build files into Quick Builder

Point build files generated by **bbpnt** can be imported into Quick Builder and amended. For example, you might want to change the Scan Period entries and perhaps add comments for each point or attach algorithms.

Configuring alarms for signal data

Alarms are used to update point parameters and are the quickest way of transmitting data over the network. Alarms can only be configured for points built on signal data.

Use alarms to:

- Update point parameters.
- Quickly transmit data over the network.
- Reduce periodic scanning.
- Store as events in order to retain Bristol Babcock alarm timestamp information. (The Sequence of Events (SOE) file is used for this purpose.)

- **Reducing periodic scanning with alarms**

Place a status point parameter on a slow scan period (300 seconds) and configure the signal in the node to alarm on a change of state.

- **Retaining timestamp information**

An event report entry includes the timestamp received in the alarm report to a resolution of one millisecond. If an alarm value is used to update a process variable, and this causes an alarm to occur, the alarm will be time stamped with the system time not the time in the alarm report. The difference between these timestamps could be significant depending on the network architecture, hence the need to retain the "real" alarm time in the SOE file.

Defining points that address Bristol Babcock

The steps for defining points with parameters that address Bristol Babcock controllers are:

1. For points that reference signals:
   a) Using the bbpnt utility, create a point source file from the ACO file. The ACO file is produced by the Bristol Babcock aic5 program and is a complete definition of the load that resides in a particular controller.
   b) Import the bbpnt output file into Quick Builder.
   c) Configure alarms for these points as required.
2. For points that reference arrays, you define these points wholly within Quick Builder. (The **bbpnt** utility does not generate a point definition file for points that reference arrays.).
3. When you have finished defining/editing both types of points in Quick Builder and are ready to download them, disable the Bristol Babcock channel on the server.
4. Download the point definitions to the server.
5. Enable the Bristol Babcock channel.

5.4 Defining a UDC channel

You must define each physical communication link to the UDC network as a UDC channel.

To define a channel:

1. Click to open the Add Items dialog box.
2. Select Channel from Add Items.
3. Select UDC from Type.
4. Set the property values on the Main tab-see
5. Click the Port tab and for Port Type select either:

- Serial
- Terminal Server

Defining a DPR channel

You need to define a channel for every connection the server uses to communicate with the DPR controllers.

To define a channel using Quick Builder:

1. Click to add a channel.

2. In the Add Items dialog box, select Channel as the item and DPR as the channel type.

3. Enter the channel details on the Main property page for the channel. For help with the channel definitions.

4. To complete the channel definition, click the Port tab and define either a serial or terminal server port.

5.5 Defining a UDC controller

The server requires a controller definition for each loop in a UDC. You need to define one controller for each UDC 3000, and up to two controllers for a UDC 3300, UDC 5000 or a UDC 6000 (these controllers can have two control loops).

When defining a UDC 3300, UDC 5000 or UDC 6000 controller, ensure that you use an appropriate numbering scheme.

To define a controller:

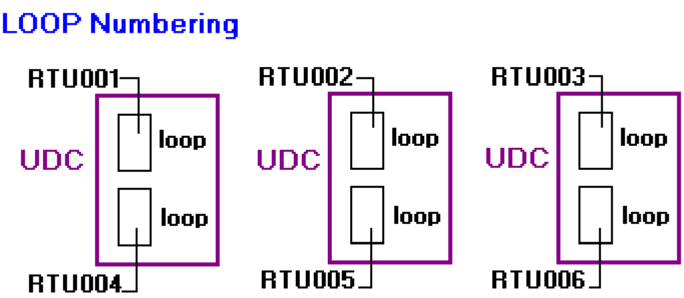
1. Click to open the Add Items dialog box.
2. Select Controller from Add Items.
3. Select UDC from Type.
4. Set the property values on the Main tab.

Loop address numbering

The server uses a separate controller to represent each of the 2 control loops in a single UDC 3300, UDC 5000 or UDC 6000.

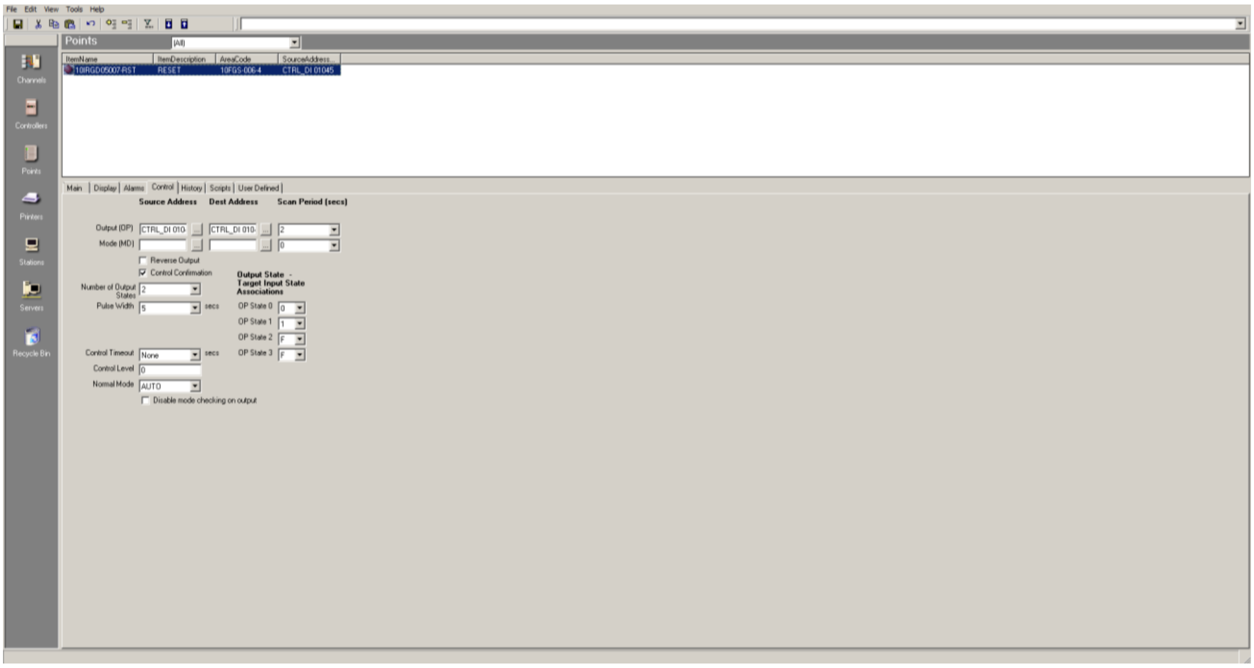
The server sends a scan request to each controller consecutively by controller item number (loop number) in ascending order. Each UDC can only communicate a maximum of 3 times per second. If the 2 loops in a UDC have consecutive controller item numbers, up to 1/3 of a second can be wasted while the server waits for the UDC to accept a scan request for the second loop.

You avoid this problem, and spread the scan load efficiently, by interspersing loop numbering across UDCs so that the second loop in a particular controller is never scanned before the first loop has responded. The following figure shows optimal numbering for 3 UDCs, each with 2 control loops.

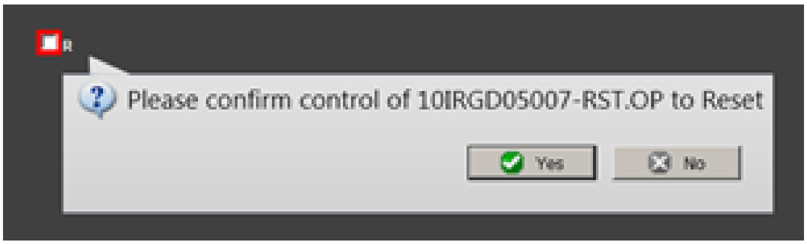


5.6 implement a confirmation box

For implement confirmation box open Quick Builder and fill control confirmation check box:

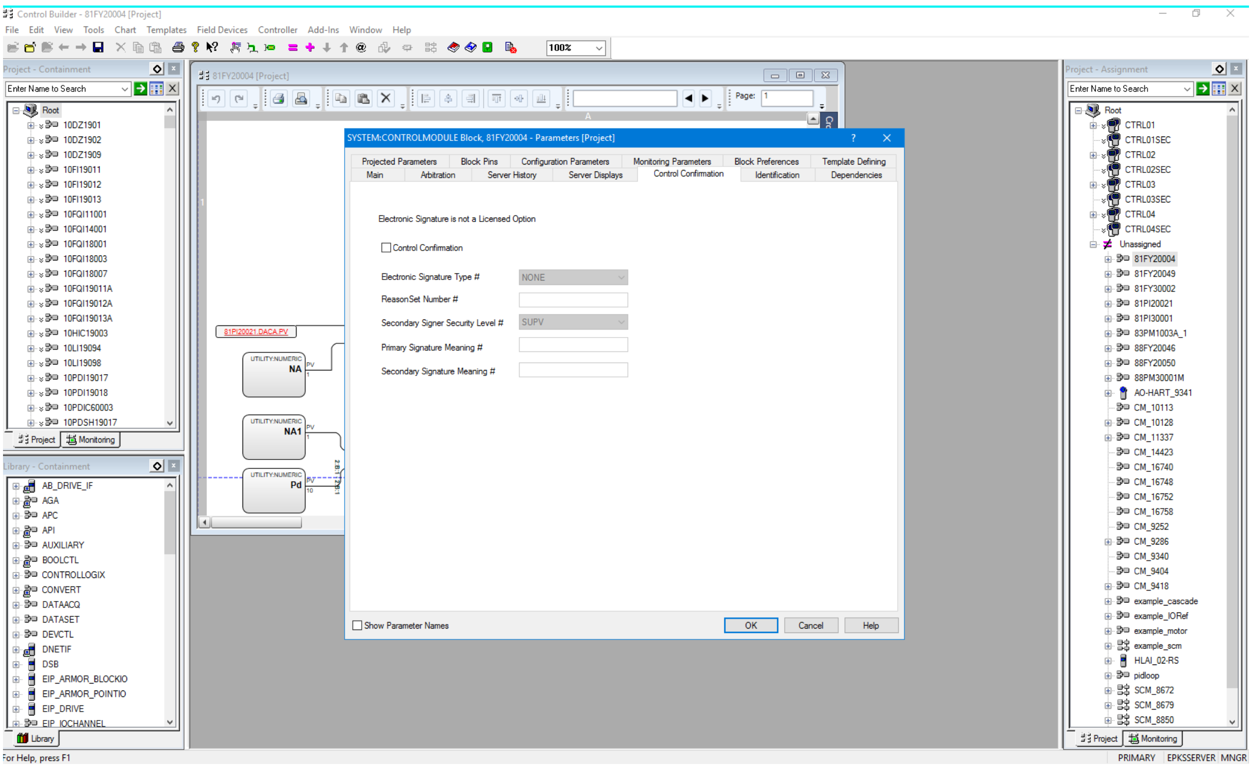


In this case in HMI pages confirmation notification will be appear for that tag, see below pictures:



Your action will not be done until click yes on this notification bar.

In case that using control modules this confirmation box will be available .
Right click on control module page and see control confirmation tab :



confirmation notification will be appear for this control moduloe too.

6 User privilege level and security

This chapter describes how securities policies have been set to meet project requirement.

6.1 Requirements

The project defines the following default users:

- VISITOR
- OPERATOR
- SUPERVISOR
- ENGINEER
- MNGR (mandatory built in access)

Each of these users has following rights:

- VISITOR: corresponds to monitoring only and no accessible command.
- OPERATOR: corresponds to normal process control and safety reset.
- SUPERVISOR: corresponds to first level of intervention, modification of tuning parameters and AOF, MOS and POS activation plus system building and configuration.
- ENGINEER: corresponds to all above access including system configuration and modifications.
- MNGR: Manager Mode (Unlimited access) including all level. Some functions (for example to configure the rights for all the others acces levels) are restricted to the MNGR role.

6.2 Scope of Responsibility

The scope of responsibility (SOR) for an operator on a particular station depends on the following

- Asset assignments and logon hours of the operator

- Station asset assignment
- Asset profile lists
- Operating groups

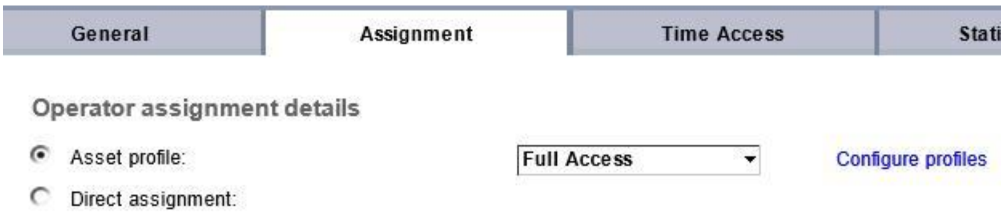
The following section specifies the setting for this system.

operation system is divided primarily in two main functional areas (Console). They are:

1. View only for all the OWS in the CCR
2. MAINTENANCE & DEVELOPMENT for all EWS in the Eng Room, the SOR is also segregated according to the operation area defined. In Honeywell EPKS, there can be broadly two ways of SOR.
3. Operator Based SOR
4. Station Based SOR

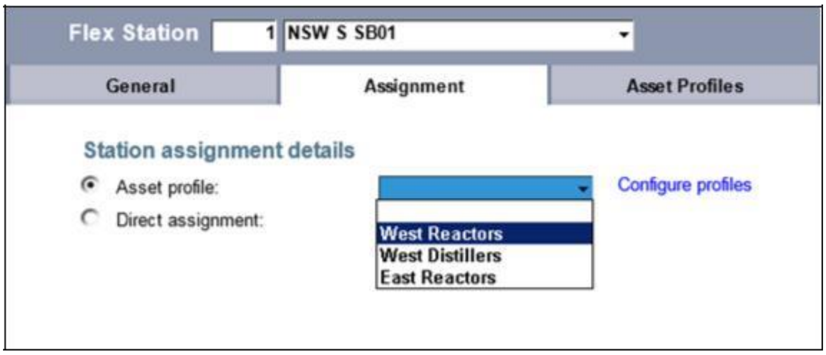
SUPERVISOR, ENGINEER and MANAGER users have access to whole data in each station.

Assignment is configured with Full Access profile, means all assets are available during all hours.



Supervisor can only modify parameters of all data that have an access level less (operator) or equal to supervisor.

- **Changing Scope of Responsibility with Asset Profiles**



Asset Profile : Collection of Asset permissions used as 'Scope of Responsibility' for a

Station or Operator.

A Profile consists of an

- Asset List, containing one or more areas,

- Asset Time Period.

Asset profiles is functionality in Experion to make it easier to configure scope of responsibility for operators and Stations. Like the screen shows above, instead of individually selecting which assets a Station can access, you can instead select from pre-configured asset profiles.

In R50
, instead of having the selection of asset profile a static configuration, Experion now allows you at run time to select from a list of asset profiles, there by changing the scope of responsibility on the fly.

Why Change Asset Profile? For times of temporary operator absence, for busy periods when you need more help such as for start ups or for operator handover situations.

6.3 Operator SOR

If operator-based security is used, one assign assets, Network tree nodes and System Components to operators (or Experion Windows's groups) to define scope of responsibility. When an operator performs an action, the assignment of that operator (or the Experion Windows group to which the operator belongs) is checked to ensure the action is within the defined scope of responsibility.

System Default Operators SOR is listed below.

In station server wide settings, the following option is check in order to take both station and operator asset assignment in to account while giving access to particular asset. For example, for asset U401, if station is given view and acknowledge rights and for particular operator, there is no rights given, then when the operator logs in to the station he will not have rights to access asset U401. Both station and operator asset assignment should have been configured to give rights to the operator to access the asset.

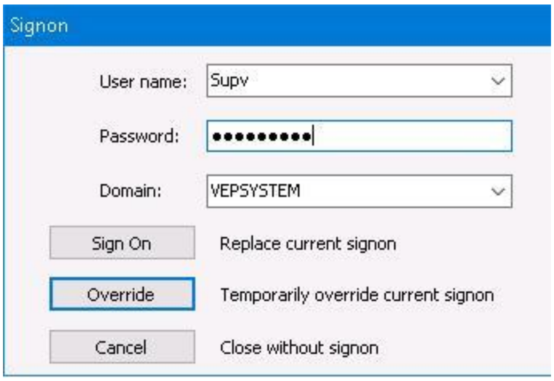
6.4 Login/Change of user procedure

Use of the Signon Experion feature:

Allows users to change credentials without logging off of Windows. The new credentials affect the station application. Signon manager credentials are applicable for the station application only when it uses operator-based security. Works with Integrated Accounts only. A major benefit of using Signon Manager is that operators can log on and off without losing view of the plant or critical processes.

Without single signon, the user must provide credentials twice – once to logon to Windows, and the second time to logon to station. When single signon is an enabled after invoking the station application, rather than prompting for credentials a second time, it uses the Windows credentials for the station logon.
Signon manager has two user interfaces – the signon bar and the signon dialog box. The signon bar is displayed at the top of the screen by default but can be moved to any edge of the screen.

It can also be configured to auto-hide so that it will only be displayed when the cursor is moved to the edge of the screen where the signon bar is hidden. The signon dialog box is displayed when a new signon is requested



After providing new credentials and clicking the signon button, new security levels are given in both the native window and the station application.

After logging on to Windows as operator, the station application can be invoked from from a logon script.

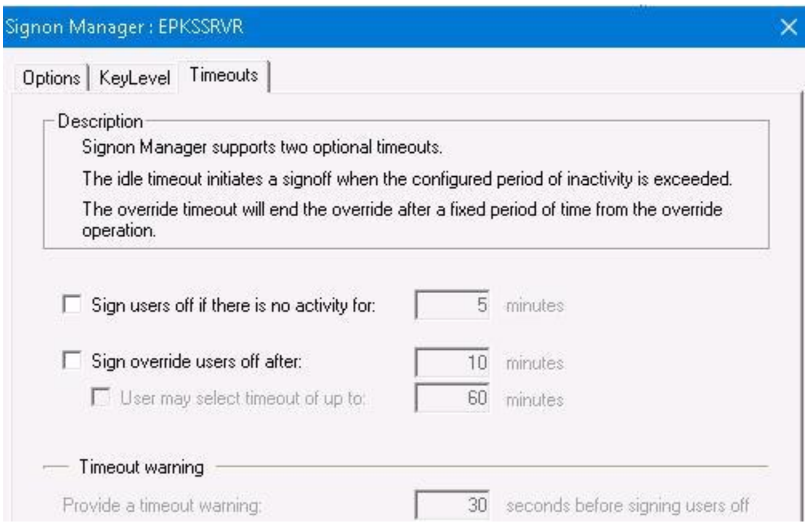
The 'Override' button on the signon dialog box is enabled only if a user is already signed on– that is, only if both 'Default' credentials and 'Signon' credentials have been provided and are being displayed on the signon bar. This allows the user to provide new credentials and click either the signon button or the override button. If the user clicks the signon button the credentials provided will replace the current signon credentials. If the override button is clicked, the credentials will be added as a third set of credentials on the signon bar. These credentials are known as the 'Override' credentials. In all cases, clicking the 'Sign Off' button will return to the 'Default' credentials. If an override operation was performed, then the 'End Override' button is active. Clicking the 'End Override' will return to the last 'Signon' credentials.

Signon Manager Timeouts:

There are two sign-on manager timeouts – the idle timeout and the override timeout. The idle timeout is displayed immediately to the left of the signon credentials on the signon bar. It starts when it detects no activity – that is, no activity from the keyboard, the mouse, or the touchscreen. The idle timer is reset upon activity. When the idle timer expires, the existing signon and override will end, and the credentials revert to the default user credentials.

Override timeout is displayed immediately to the left of the override credentials on the signon bar. It starts when an override operation is performed – that is, it starts when a user provides credentials in the signon dialog box and clicks the override button. The override timer is reset only when another override operation is performed. When the

override timer expires the override operation ends, and the credentials revert to the signon user credentials.



If you want Windows to start automatically without the operator entering a Windows password, you can set up automatic logon. If you set up automatic logon, the computer always logs on with the same user name and password.

6.5 User name and Passwords defined for the Project

Password set up for each User name will be applied on all control & monitoring stations. The active user name is shown at the lower right corner of the screen

A set of ten operators with name OPERATOR01 to OPERATOR10 with different passwords are created on the project.

For user with OPERATOR access level there is no automatic logoff

Users with SUPERVISOR OR ENGINEER level access have an automatic logoff after 15 minutes without activity.

User Name	**User group**	**Privilege level**	**Option**
OPERATOR 01 OPERATOR 02 OPERATOR 03	Local Operators	Oper	**Option**
SUPERVISOR	Local Supervisor	Supv	Automatic log off after 15 min
ENGINEER	Local Engineer	Eng	

6.6 Privilege level defined for the project

The security levels can be assigned to functions.

In order to use the function, the current Station security level must be equal to or greater than the security level assigned to the function.

For example, a button on a display might be assigned a security level of “Engineer” when a custom display is built. In order for an operator to use the push button, the Station security level must be either Engineer or greater.

There are two aspects to operator-based security; authentication and authorization. Authentication is the process of verifying that a user is known to the system, while authorization controls what a known user can do within the system. Accounts are used to restrict access and authority within Station. Traditional operator accounts or integrated accounts may be used.

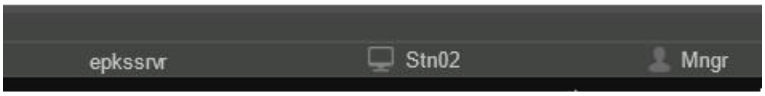
For traditional operator accounts, authentication of the user is done by the Experion PKS server against credentials stored in Experion PKS. Authorization is also controlled by Experion PKS using security levels and, if applicable, areas.

For integrated accounts, authentication of the user is done by Windows on the server computer against the Windows user account. Authorization is then controlled by the Experion PKS server using security levels and, if applicable, areas.

There are six different security levels in Experion PKS. The levels are shown in the following table:

Default Security Level Acronym	Default Meaning
View Only (LV1)*	View-only mode
Ack Only (LV2)*	Alarm acknowledgement mode
Oper	Operator mode
Supv	Supervisor mode
Engr	Engineer mode
Mngr	Manager mode

* Applicable to Operator-based security only



The current security level is shown at the lower right corner of the screen.

** The Maint security level is a new security level added to the standard security levels above-mentioned (to be confirmed)

What each level can do is documented in EPKS R501 Knowledge Builder/Server and Client Configuration Guide.

- for general actions

- for System Security and Asset Configuration

- for actions on Points

- for actions on System Hardware

- for actions on Alarms & Alarm Management

- for actions on Event journals and Event Management

- for actions on Groups and Trends

- For actions on Report

Actions permitted at each security level: The tables in the following section show the actions permitted at each security level in Experion PKS. Most of these actions correspond to options available from Station.

Permissions for General Experion PKS Actions

Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View start-up display	Yes	Yes	Yes	Yes	Yes	Yes
Call up displays	Yes	Yes	Yes	Yes	Yes	Yes
View message pad	Yes	Yes	Yes	Yes	Yes	Yes
Edit message pad			Yes	Yes	Yes	Yes
View redundant server status	Yes	Yes	Yes	Yes	Yes	Yes
Configure server redundancy				Yes	Yes	Yes
Fail over/synchronize redundant servers				(*1)	Yes	Yes
View History Assignment	Yes	Yes	Yes	Yes	Yes	Yes
Configure History Assignment				Yes	Yes	Yes
View server wide Station settings	Yes	Yes	Yes	Yes	Yes	Yes
Configure server wide Station settings						Yes
Print page	Yes	Yes	Yes	Yes	Yes	Yes
Change SUPV level display object				Yes	Yes	Yes
Change ENGR level display object					Yes	Yes
Change MNGR level display object						Yes

(*1) Default setting (Supv) modified. In System Configuration -> Redundant server – Advanced tab

\

System security and areas

Permissions for System Security and Areas						
Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View areas	Yes	Yes	Yes	Yes	Yes	Yes
Configure and assign areas						Yes
View area profiles	Yes	Yes	Yes	Yes	Yes	Yes
Configure and assign area profiles						Yes
Configure and assign area code						Yes
Configure area description						Yes
View Operator's ID and name	Yes	Yes	Yes	Yes	Yes	Yes
View Operator's full details				Yes	Yes	Yes
Configure Operators, excluding security level, command segregation and area assignment						Yes
Configure Operator command segregation and area assignment						Yes
View sign on administration	Yes	Yes	Yes	Yes	Yes	Yes
Configure sign on administration					Yes	Yes

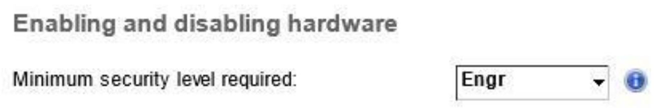
Points

Permissions for Points						
Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View point information	Yes	Yes	Yes	Yes	Yes	Yes
Configure point information excluding descriptor and area				Yes	Yes	Yes
Configure point information descriptor and area						Yes
Control points			Yes	Yes	Yes	Yes
View shifts	Yes	Yes	Yes	Yes	Yes	Yes
Configure shifts				Yes	Yes	Yes
Configure shift management				Yes	Yes	Yes
Initialize shift management					Yes	Yes

System hardware

Permissions for System Hardware						
Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View Station, printer, channel, point server, system interfaces, controller, DSA and redundant server status	Yes	Yes	Yes	Yes	Yes	Yes
Modify Station, printer, channel and controller status				(*1)	Yes	Yes

(*1) Default setting (Supv) modified. In System Configuration -> Server Wide Settings – Security tab



Alarms and alarm management

Permissions for Alarms and Alarm Management						
Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View alarms	Yes	Yes	Yes	Yes	Yes	Yes
Acknowledge alarms		Yes	Yes	Yes	Yes	Yes
Enable and disable alarms server wide						Yes
Configure server-wide setting for Alarm return-to-normal only non-alarm states						Yes
View system alarm priorities	Yes	Yes	Yes	Yes	Yes	Yes
Configure system alarm priorities						Yes
Configure server redundancy alarm priority						Yes
View server wide alarm color settings	Yes	Yes	Yes	Yes	Yes	Yes
Configure server wide alarm color settings						Yes
View Alarm, Event and Message Summary permission	Yes	Yes	Yes	Yes	Yes	Yes
Configure Alarm, Event and Message Summary permission						Yes

Events and event archiving

Permissions for Events and Event Archiving						
Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View events	Yes	Yes	Yes	Yes	Yes	Yes

Archive / Restore events			Yes	Yes	Yes	Yes
Configure event archive parameters						Yes

Groups and trends

Permissions for Groups and Trends						
Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View groups	Yes	Yes	Yes	Yes	Yes	Yes
Configure groups				Yes	Yes	Yes
View trends	Yes	Yes	Yes	Yes	Yes	Yes
Configure trends			Yes	Yes	Yes	Yes

Reports

Permissions for Report Functions						
Action	LV1	LV2	OPER	SUPV	ENGR	MNGR
View report definitions	Yes	Yes	Yes	Yes	Yes	Yes
Configure report definitions, excluding area code				Yes	Yes	Yes
Configure area codes on report definitions						Yes
Request reports	Yes	Yes	Yes	Yes	Yes	Yes

6.7 Control levels – Default access level for Data

- Authority-Security level Configuration

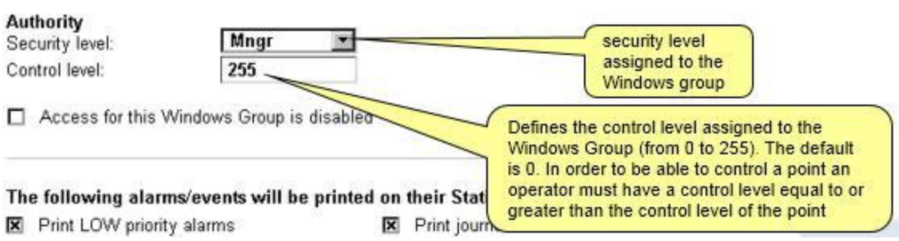
  - Configure control level for each Operator

  - Configure control level for each point

Operator sign-on/sign-off security provides up to 255 control levels, to limit operator (in the broadest sense e.g. Oper, Supv, Engr…) control of individual points.

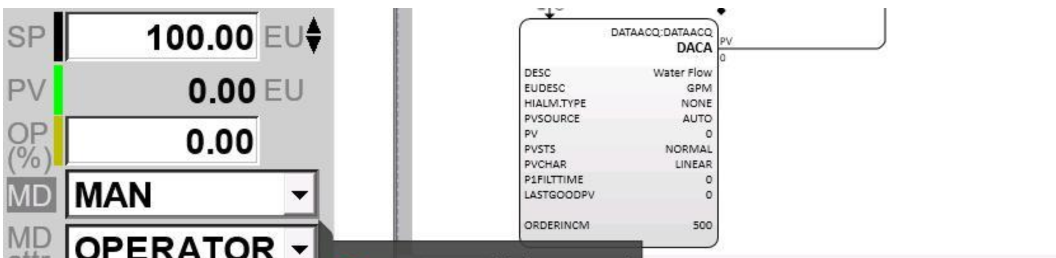
– Control actions to a given point are only allowed if the control level configured in the operator profile exceeds the level assigned to the point

– Applies to all access levels including manage



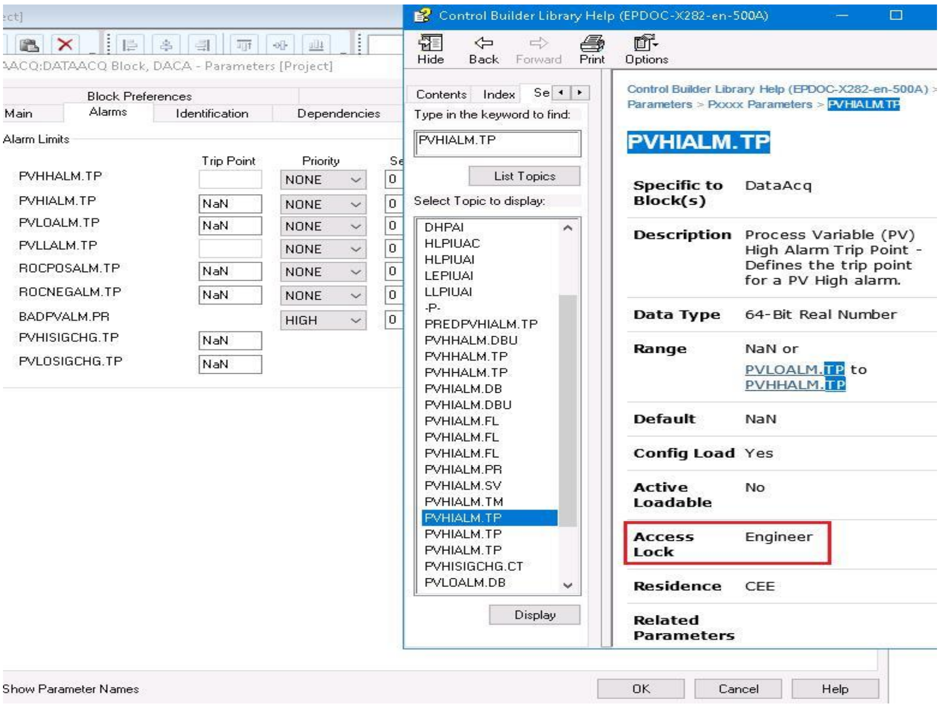
For each point, in CM block configuration – Server History tab. The Control level is defined

In Chart view or faceplate of the displayed point in station, if the Control level configured for this CM required is higher than the one defined for the logged user then the change is refused with a suitable system message



For the default parameters security levels of all C300 function Blocks or any items from the library in Control Builder, please refer to the EPKS PKS User Manuals "Experion PKS R501 Documentation Center" available with the system or with the online help.

See an example below:

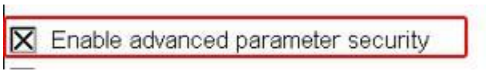


Advanced Parameter Security

Experion security levels can be customized to allow or deny access to individual point.parameters. Therefore, even if the station or the operator is given full access to a particular asset, the operator may not have access to all parameters of all points assigned to that asset. This feature is called advanced parameter security.

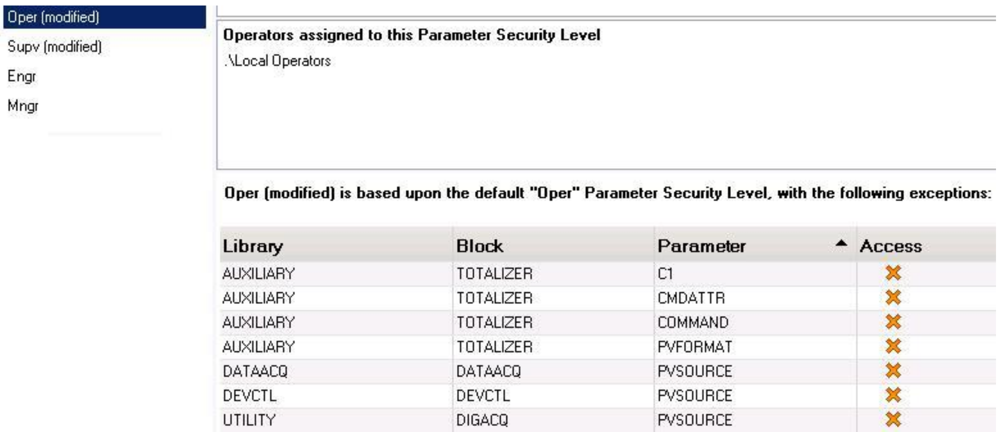
Advanced Parameter Security allows security levels to be created or customized. These security levels can be configured to allow or deny access to point.parameters – on a parameter by parameter basis.

First of all Advanced parameter security is an option which must be enabled from the security tab of server-wide settings



To Configure Parameter Security Level

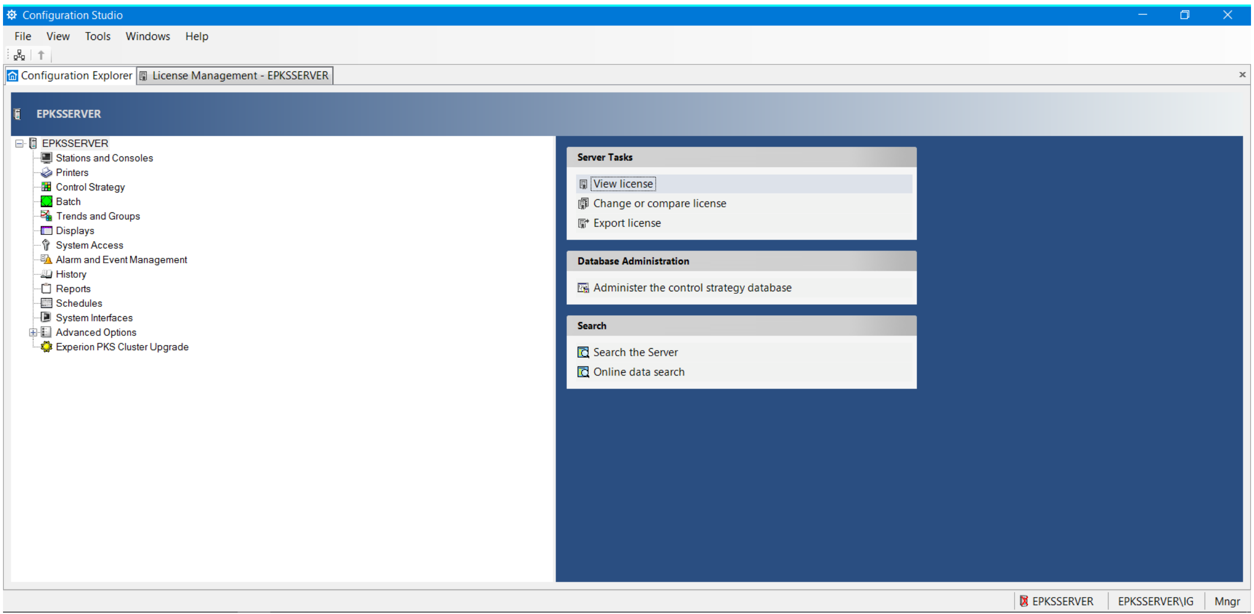
– Open Configuration Studio > System Access > Configure Advanced Parameter Security



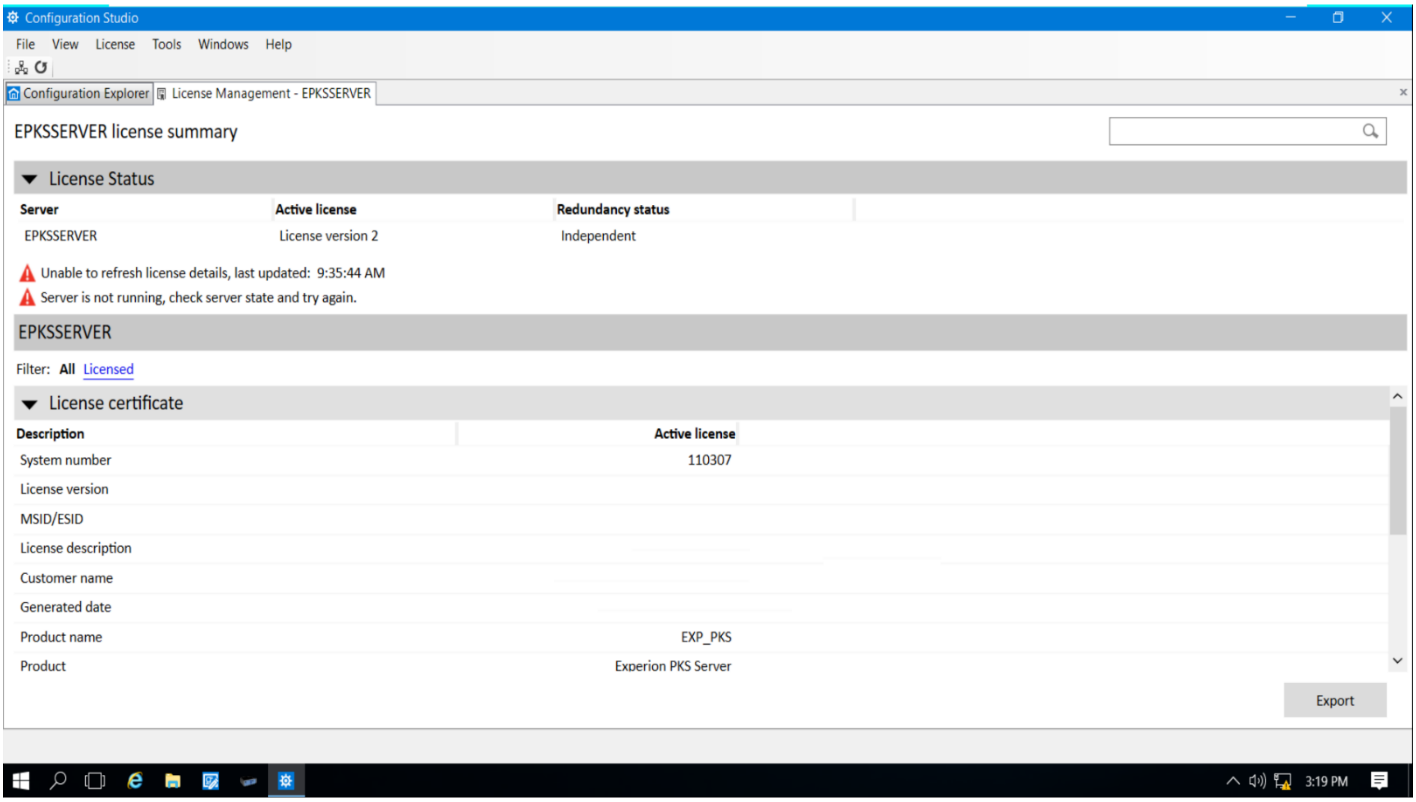
Note: Advanced parameter security will be defined at the beginning of the project according to the level of access, which is mostly approved by default and does not need to be changed during the project. In certain cases, if high-level access is required for ACTION in the control of a specific module, these issues can be resolved through the control of the module.

7 License Procedure

For collecting information about software license , you can refer to configuration studio and open view license tab and read related license.

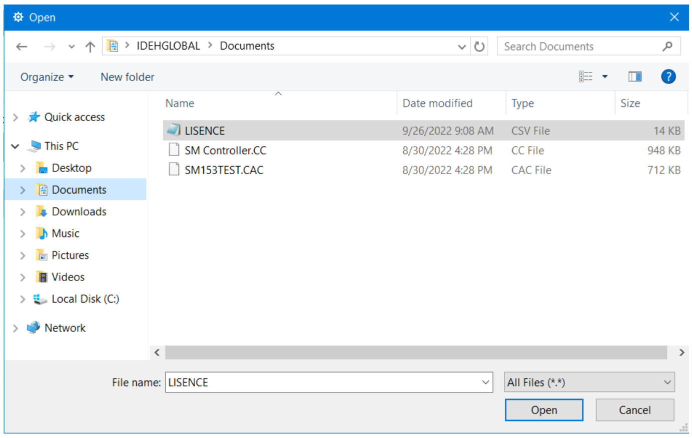


In this section its show each section license:



For more detail you can export license with export button.

In case of add or change the license , should click on (change or compare) and select the license and add it to software .



7.1 License list

According to BOM of project these list of license will be consider for control system softw

01

Description	Article number	Vendor	QTY
Experion Process Points (2000Points)	EP-DPR02K	Honeywell	1
Experion Scada Points (serial link signals) (1000Points)	EP-DSC01K	Honeywell	1
C300 (CEE) Redundant Control Solvers	TC-SWCS30	Honeywell	3
Number of client Control Builders	TC-SWCB31	Honeywell	1
Safety builder licens, SOE enable	EP-BLSE01	Honeywell	1
Display Builders	EP-TDSPBD	Honeywell	1
SM controller (all SM controller tags can be integrated into experion servers/ C300 controllers with this license BOM)	FS-SMCT-200	Honeywell	1
Console Stations	EP-STAC01	Honeywell	6
Flex Stations License	EP-STAT01	Honeywell	10
Multi Window Stations (dual monitoring)	EP-SMWIN1	Honeywell	6
SOE Station (including Flex station license for log sequence of event)	EP-STAT01	Honeywell	1
Experion SW Rel R501	EP-PKS501	Honeywell	1
safety historian (Including Honeywell Safety Manager & FSC Integration License for Activate safety historian)	-	Honeywell	1
MODBUS (RTU, PLUS, ASCII AND RTU) INTERFACE	EP-IMDBUS	Honeywell	1
Asset Manager PKS Base System	AM-AMPS00-200	Honeywell	1
Asset Manager Asset 500 Analog Tag	AM-AMAG5-100	Honeywell	1
Microsoft office Excel	-	Microsoft	20
Windows 10Pro	-	Microsoft	19

Microsoft Windows server	**-**	***Microsoft***	***1***

8 Tag naming Convention

This section defines naming convention to be assigned for components of Experion system of this project.

A point built to represent a loop in a process control always consists of a Control Module and its component blocks inside referred as Function Blocks. The same type of Function Blocks in different Control Module can have the same name but it must be unique within its Control Module. Since Experion recognizes a Control Module tag name as an independent tag, it must be unique throughout the entire system.

Both Control Module and Function Block tag has the same rule for its naming as follow:

- Control Module must have a unique tag name throughout the entire system.
- Tag name can has up to 16 characters maximum.
- Tag name must begin with alpha character.
- Tag name can consist with alpha character, alphanumeric character and '_'.
- A Function Block tag name must be unique within its associated Control Module name.

8.1 Physical Equipment Blocks Tag Names

The naming convention for this type of block is as follows:

8.1.1 Tag name for Control Processor Module (C300)

CTRLmm

Where

mm: 01~10, Controller no.

8.1.2 Tag name for Control Execution Environment (CEE)

CEE tag name is defined as follows:

CEEmm

Where

mm: 01~10, controller no.

CEE only resides in primary controllers.

8.1.3 IOLK Name

IOM tag name is defined as follows:

- **IOLKYY_mm**,1 for link01 and 2 for link02

Where

YY: Controller no

mm: 01~02, IOLINK no.

8.1.4 IOTA Name

[HH][mm]

Where

HH: Module type:

AI Analog Input Module

AO Analog Output Module

DI Digital Input Module

DO Digital Output Module

mm: 01~40, IOM no.

Example: AI23

8.2 Functional Blocks Tag Names

8.2.1 IO Channel Block Name

IO channel block name shall always refer to the IO LIST and P & ID as follows:

8.2.2 DACA Block Name

For single DACA block in a Control Module, the block will be named as follows:

DACA

For multiple DACA blocks in the same Control Module, the block will be named as follows:

DACAX

Where:

X: 1,2,3,...etc

8.2.3 PID Block Name

For single PID block in a Control Module, the block will be named as follows:

PIDA

For multiple PID blocks in the same Control Module, the block will be named as follows:

PIDX

Where:

X: 1,2,3,...etc

8.2.4 REGCALC Block Name

For single REGCALC block in a Control Module, the block will be named as follows:

REGCALCA

For multiple REGCALC blocks in the same Control Module, the block will be named as follows:

REGCALCX

Where:

X: 1,2,3,...etc

8.2.5 FANOUT Block Name

For single FANOUT block in a Control Module, the block will be named as follows:

FANOUTA

For multiple FANOUT blocks in the same Control Module, the block will be named as follows:

FANOUTX

Where:

X: 1,2,3,...etc

8.2.6 FLAG Block Tag Name

For single FLAG block in a Control Module, the block will be named as follows:

FLAGA

For multiple FLAG blocks in the same Control Module, the block will be named as follows:

FLAGX

Where:

X: 1,2,3,...A/B/C etc

For more clarification Flags are given the name also like START / STOP etc.

8.2.7 DEVCTL Block Name

For single DEVCTL block in a Control Module, the block will be named as follows:

DEVCTLA

For multiple DEVCTL blocks in the same Control Module, the block will be named as follows:

DEVCTLX

Where:

X: 1,2,3,...etc

8.2.8 AUXCALC Block Name

For single AUXACLC block in a Control Module, the block will be named as follows:

AUXCALCA

For multiple AUXCALC blocks in the same Control Module, the block will be named as follows:

AUXCALCX

Where:

X: 1,2,3,...etc

8.2.9 Logic Blocks Name

AND Block Name

For single AND block in a Control Module, the block will be named as follows:

ANDA

For multiple AND blocks in the same Control Module, the block will be named as follows:

ANDX

Where:

X: 1,2,3,...etc

OR Block Name

For single OR block in a Control Module, the block will be named as follows:

ORA

For multiple OR blocks in the same Control Module, the block will be named as follows:

ORX

Where:

X: 1,2,3,...etc

EQ Block Name

For single EQ block in a Control Module, the block will be named as follows:

EQA

For multiple EQ blocks in the same Control Module, the block will be named as follows:

EQX

Where:

X: 1,2,3,...etc

GE Block Name

For single GE block in a Control Module, the block will be named as follows:

GEA

For multiple GE blocks in the same Control Module, the block will be named as follows:

GEX

Where:

X: 1,2,3,...etc

GT Block Name

For single GT block in a Control Module, the block will be named as follows:

GTA

For multiple GE blocks in the same Control Module, the block will be named as follows:

GTX

Where:

X: 1,2,3,...etc

LE Block Name

For single LE block in a Control Module, the block will be named as follows:

LEA

For multiple LE blocks in the same Control Module, the block will be named as follows:

LEX

Where:

X: 1,2,3,...etc

LT Block Name

For single LT block in a Control Module, the block will be named as follows:

LTA

For multiple LT blocks in the same Control Module, the block will be named as follows:

LTX

Where:

X: 1,2,3,...etc

SEL Block Name

For single SEL block in a Control Module, the block will be named as follows:

SELA

For multiple SEL blocks in the same Control Module, the block will be named as follows:

SELX

Where:

X: 1,2,3,...etc

SELREAL Block Name

For single SELREAL block in a Control Module, the block will be named as follows:

SELREALA

For multiple SELREAL blocks in the same Control Module, the block will be named as follows:

SELREALX

Where:

X: 1,2,3,...etc

8.3 CM Name

Since a CM represents a point, so its naming shall always refer to P & ID and IO List as follows:

YYLLLAAA[A]

Where

YY: Unit Code ,as per the I/O schedule and P &IDs.

Y: Area Number, e.g.: 1, 2, etc. as per the I/O schedule / P &Ids. If there is no area No. or it is 0, then "_ " will be used.

LLL: Instrument Type as per the I/O schedule / P& Ids

AAA: Inst. / Control Loop Number

[A]: Some loop number ends with the alphabetic, eg. O for open, C for Close status of valve limit s/w

Eg.: 10LIC400, etc.

For additional information, CM description can be defined to the maximum of 24 characters, including spaces. It follows the description specified in database.

8.4 SCM Name

Since SCM contains some points related to a process control, e.g. shutdown logic, pump logic, etc. so the naming is not specified for one particular point.

YYSCMLLLAAA

Where

YY: Unit Number, e.g.: , etc. as per the I/O schedule / P &Ids

LLL: Loop Type as per the I/O Schedule

AAA: Control Loop Number without the alphabetical ending

Eg.: SCM_30HCV001

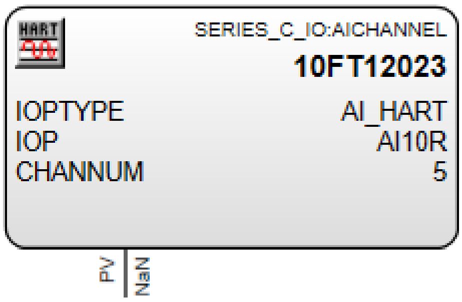
9 SYSTEM LIBRARY BLOCK TYPES

9.1 SERIES C IO BLOCK TYPES

9.1.1 AI

Description

Provides standard analog interface to control function blocks.



AI-HART

High Level Analog Input with HART, 16 Channel.
This block is used for the following Series C IOMs:
CU-PAIH01
CC-PAIH01
Each input channel is capable of scanning (100 ms sampling) a standard analog input (0 to 100%) and supporting digital data transfer using HART communications protocol.
The analog input point converts an analog PV signal received from a field sensor to engineering units for use by other data points in the control strategy. To accomplish this function, the AI-HART point performs the following functions.

- Analog-to Digital Conversion
- PV Characterization
- Range Checking and PV Filtering
- PV Source Selection
- Alarm Detection

Additionally, the IOM can issue HART protocol commands and receive data from HART capable devices. Device Id data is read from the device and cached in the IOM. Dynamic and device variable data and device status is collected from the device for use by the control system. IOM allows for servicing of any pass-through commands issued from host/master devices.

Function

The analog input channel converts an analog PV signal received from a field sensor to engineering units for use by other function blocks in the C300, and by the rest of Experion PKS

Inputs

Floating point value in engineering units.

Outputs

Floating point value in engineering units.

Associated Block

Prior to loading, block must be "associated" with 1 channel of corresponding IOM block that interfaces with the physical AI hardware module at execution runtime.

Parameters

ACCEPTDEV ***ACCEPTRNG*** ***ALMENBSTATE*** ***ASSOCASSET*** ***BADPVFL*** ***CHANNUM*** ***CJTACT*** ***COMMAND*** ***CONTAINEDIN*** ***DAMPING*** ***DECONF*** ***DEVICELOCATION*** ***DVRNGEXT*** ***EURNGEXT*** ***HARTVERSION*** ***HCFGDEV*** ***HCMD00*** ***HCMD12*** ***HCMD13*** ***HCMD14*** ***HCMD16*** ***HCMD48BT[1..200]*** ***HCMD48NOTIFY[1..200]*** ***HDEVID*** ***HDEVMFG*** ***HDEVREV*** ***HDEVST*** ***HDEVSTSTATUS*** ***HDEVTYPE*** ***HDEVTYPENAME*** ***HDYNCC[1..4]*** ***HDYNEU[1..4]*** ***HDYNST[1..4]*** ***HENABLE***	***HEXTDEVST*** ***HLOCKBYPRIMARYMASTER*** ***HLOCKPERMANENT*** ***HLOCKSTATUS*** ***HMAINTREQ*** ***HMAXDEVVARS*** ***HNCFGCHG*** ***HNSMMINPRE*** ***HPVCHNFLAGS*** ***HPVMISM*** ***HSCANCFG*** ***HSCANOVR*** ***HSLOTCC[1..4] (HART Revision 6.0)*** ***HSLOTST [1..4]*** ***HSLOTVAL [1..4]*** ***HTAG*** ***HVARALERT*** ***INPTDIR*** ***IOP*** ***IOPTYPE*** ***JOURNALONLY*** ***LRL*** ***LRV*** ***PNTFORM*** ***PNTTYPE*** ***PTEXECST*** ***PV*** ***PVCHAR*** ***PVRAWHI*** ***PVRAWLO*** ***PVSTS*** ***REDTAG*** ***SENSRTYP*** ***URL*** ***URV***

9.1.2 AI Channel

Overview

The AI channel block represents a single analog input point on either an AI-HART or AI-LLMUX I/O Module. The type of analog input IOM needed is based on the:

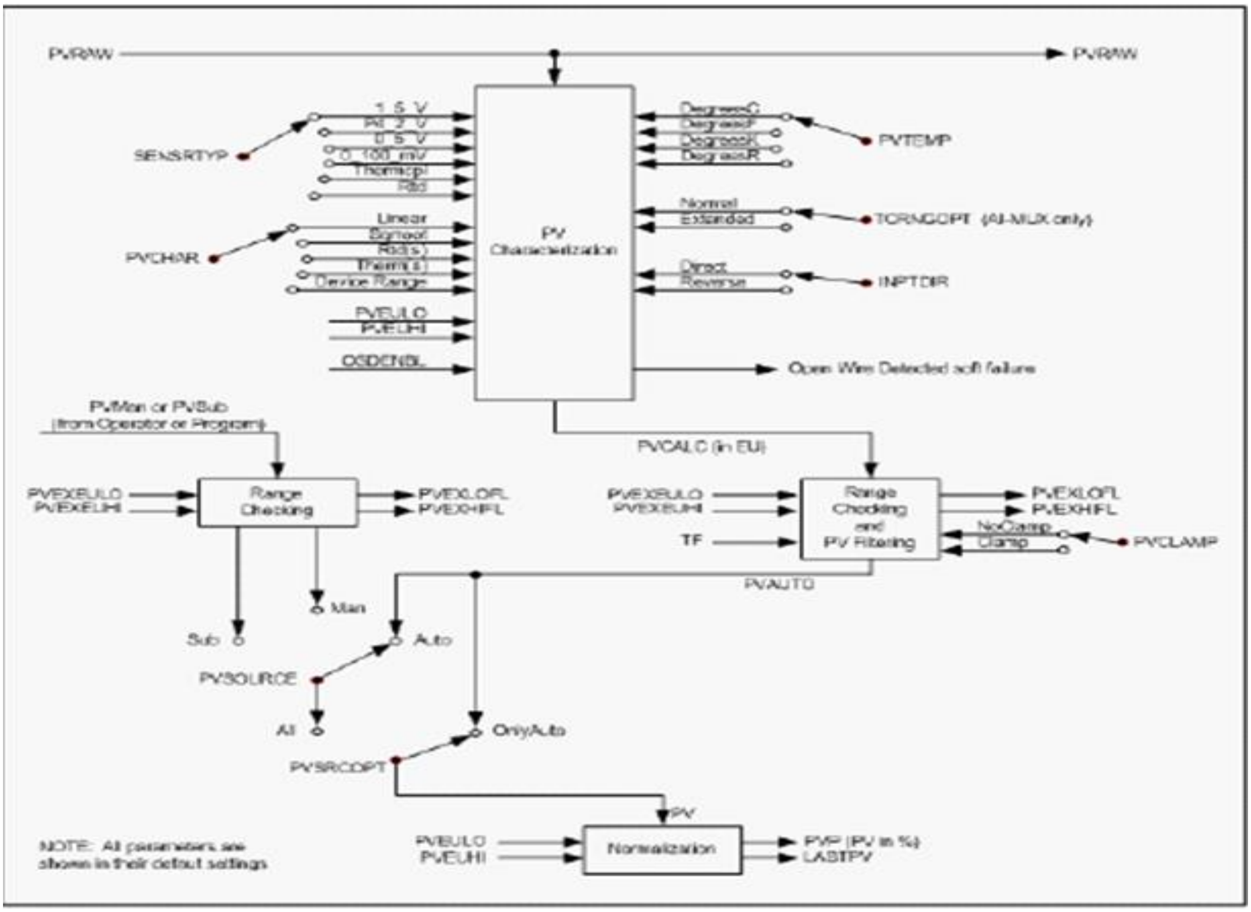
- type of field sensor that is providing the input to the channel
- PV characterization options you select (as listed in the table in Determining PV Characterization):

Channels contained within AI-HL and AI-HART IOM is generally used for control points. Channels located in the AI-LLMUX IOM are generally used for data acquisition points.

The analog input channel converts an analog PV signal received from a field sensor to engineering units for use by other function blocks in the C300, and by the rest of Experion.

To accomplish this function, the analog input channel, as displayed in figure below, performs:

- Analog-to digital conversion
- PV characterization
- Open Wire Detection on 4-20 mA inputs (configured as 1-5V) only
- Range Checking and PV filtering
- PV source selection



Analog Input conversion

Determining PV Characterization

The PV signal received from the field is characterized based on the entries that you make for the parameters:

- SENSRTYP
- PVCHAR
- PVTEMP
- INPTDIR, and
- TCRNGOPT

The input PV signal is:

3. to a raw PV signal (PVRAW) whose units can be %, ratio, millivolts, microvolts, or milliohms depending on the entry made for the SENSRTYP parameter

4. converted to the engineering units

The engineering unit conversions that are performed in the AI-HL, AI-HART and AI-LLMUX points are listed in the table below.

Sensor type (SENSRTYP)	AI Module type	PVCHAR Options	PVRAW (note 1)	PVCALC (note 2)	Bad PV detection
0-5-V (0 to 5 volts)	AI-HL[6] AI-HART (PAIH01 only)	Linear square rt.	percent	EU	Range check on PVCALC
P4_2_V (0.4 to 2 volts)	AI-HL[6] AI-HART (PAIH01 only)	Linear square rt.	percent	EU	Range check on PVCALC. Checks for open input
1_5_V (1 to 5 volts and 4 to 20 mA)[5])	AI-HL[6] AI-HART	Linear square rt. Device Range	percent	EU	Range check on PVCALC. Checks for open input
0_100_mV (0 to 100 mV)	AI-LLMUX	Linear	millivolts	EU (note 3)	Range check on PVCALC
Thermcpl (Thermocouple)	AI-LLMUX	See PVCHAR for complete list	microvolts	EU	Range check on PVCALC. Open thermocouple detection
RTD	AI-LLMUX	See PVCHAR for complete list	milliohms	EU	Range check on PVCALC

AI engineering unit conversions

Determining Thermal Conversion

Thermal linearization is performed on thermocouple and RTD input types. All thermocouples (#therm) listed in the PVCHAR parameter definition, are supported by the analog input point. The range of the thermocouple type used with the AI-LLMUX channel can be increased by selecting Extended as the entry for the TCRNGOPT parameter.

The AI-LLMUX channels calculate the reference junction compensation from the measured reference junction output level. This value is stored and then later converted back to microvolts, with respect to 0 degrees C, for each thermocouple that is to be compensated. The cold-junction reference compensation (PVREFJN) parameter is expressed in microvolts for the specified thermocouple and is added to the microvolt value for PVRAW.

All RTDs (*RTD) listed in the PVCHAR parameter definition, are supported by the analog input point:

For an RTD, the AI-LLMUX channels calculate the lead-wire compensation and then subtract the value from PVRAW. The maximum allowable lead-wire resistance and intrinsic safety barrier resistance for the RTDs are listed in the table below.

	Maximum allowable lead resistance - Note 1 (units are ohms)		Maximum allowable intrinsic safety barrier resistance (units are ohms)	
RTD type	**Entire loop**	**Per leg**	**Entire loop**	**Per leg**
Pt:100 DIN characterization	20	10	18	18
Pt:100 JIS characterization	20	10	18	18
Ni: 120 Edison type 7 characterization	20	10	18	18
Cu: 10 SEER Standard characterization	20	10	0	0
Note 1: Proper compensation for lead-wire resistance depends on the resistance being equal in each leg of the RTD. This includes resistance due to lead-wire resistance and intrinsic safety barriers. No provision is made to compensate for lead-wire resistance mismatch or intrinsic safety-barrier resistance mismatch. Both the lead resistance and the intrinsic-safety-barrier resistance are allowed simultaneously when connected to an RTD in a Division 1 area.				

RTD lead wire characteristics

Open Wire Detection

The open wire diagnostic detects and annunciates broken field wires. In addition, a seemingly valid PV from a channel diagnosed as having a broken-wire will not be made available (thus preventing incorrect control action). Open Wire Detection is available with AI-HART module.

If open wire detection is enabled (OWDENBL = ON) and the IOM detects the broken-wire condition, then

- Soft Failure 179 “Open Wire Detected” is generated, and
- PVRAW and PVAUTO will consequently be set to NaN.

Checking and Filtering PV Range

PV range checking ensures that the PVCALC output of PV characterization is within the limits defined by parameters PVEXEULO and PVEXEUHI. If either of the limits is violated, the output of the PVAUTO is set to NaN if clamping has not been specified. If clamping has been specified, the output of the PVAUTO is clamped to PVEXEUHI or PVEXEULO, except when PVRAW, PVCALC, and PVAUTO will consequently be set to NaN.

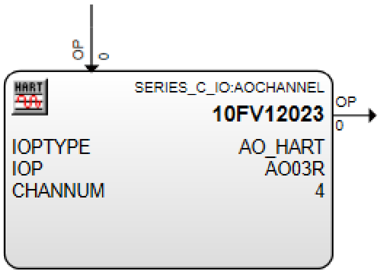
If the range-checked and filtered value is less than the value specified by the user-configured LOCUTOFF parameter, the final output called PVAUTO is forced to PVEULO.

First-order filtering is performed on PVCALC, as specified by the user through parameter TF (filter lag time).

9.1.3 AO

Description

Provides a standard analog output signal for operating final control elements.



AO-HART

Analog Output with HART IOM, 16 channel.

This block is used for the following Series C IOMs:

CU-PAOH01

CC-PAOH01

Each channel is capable of supplying a standard analog output (4 to 20mA) and supporting digital data transfer using HART communications protocol.

To accomplish this function, the AO-HART point performs the following functions.

- Analog-to Digital Conversion
- PV Characterization
- Range Checking and PV Filtering
- PV Source Selection
- Alarm Detection

The IOM can issue HART protocol commands and receive data from HART capable devices. Device Id data is read from the device and cached in the IOM. Dynamic and device variable data and device status is collected from the device for use by the control system. IOM allows for servicing of any pass-through commands issued from host/master devices.

Function

The AO channel block converts the output value (OP) to a 4-20 mA output signal for operating final control elements such as valves and actuators in the field. The OP parameter value can be controlled from a Experion PKS regulatory point, the operator, or an SCM.

Inputs

Only one control block can interface to this block.

Outputs

Floating point value in engineering units.

Associated Block

Prior to loading, block must be "associated" with 1 channel of corresponding IOM block that interfaces with physical AO hardware module at execution runtime.

Parameters

ACCEPTDEV CHANNUM COMMFAILFL CONTAINEDIN DEVICELOCATION FAULTOPT FAULTVALUE HALARMENABLE HARTVERSION HCFGDEV HCMD00 HCMD12 HCMD13 HCMD14 HCMD16 HCMD48BT[1..200] HCMD48NOTIFY[1..200] HDEVREV HDEVST HDEVST HDEVSTSTATUS HDEVTYPE HDEVTYPENAME HDYNCC[1..4] HDYNEU[1..4] HDYNST[1..4] HENABLE HEXTDEVST	HLOCKBYPRIMARYMASTER HLOCKPERMANENT HLOCKSTATUS HMAINTREQ HMAXDEVVARS HNCFGCHG HNSMMINPRE HPVCHNFLAGS HSCANCFG HSCANOVR HSLOTCC[1..4] (HART Revision 6.0) HSLOTST[1..4] (HART Revision 6.0) HSLOTVAL [1..4] HTAG HVARALERT IOP IOPTYPE INITREQ INITVAL OP OPCHAR PNTFORM PNTTYPE PTEXECST REDTAG

9.1.4 AO Channel

Overview

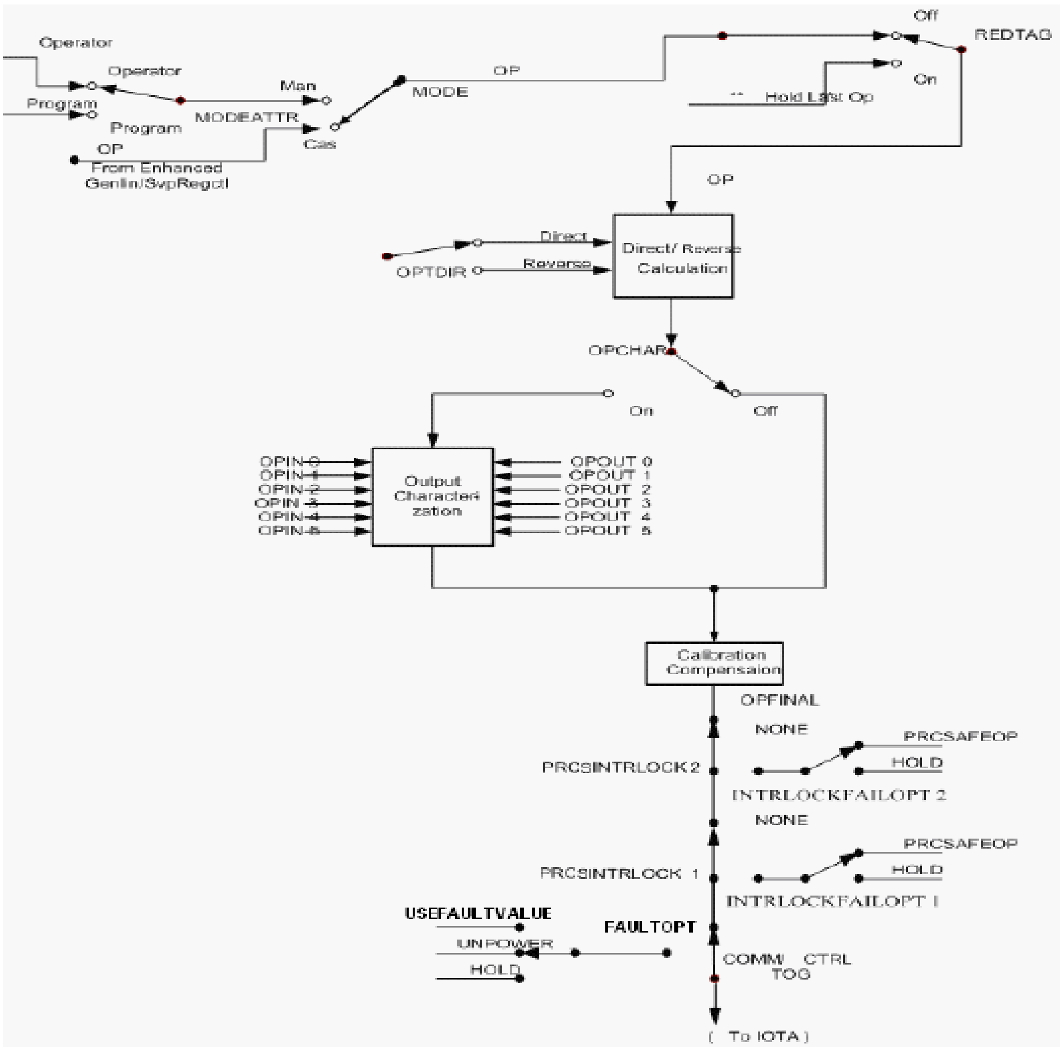
SVP_AO channel supports unipolar and bipolar current outputs in addition to the standard 4-20mA analog output. It converts the output value (OP) to the output signal for operating the final control elements, such as valves and actuators, in the field. It accepts inputs only from ENHGENLIN block executing in the C300 – 20mS CEE Controller.

To convert the OP value to a configured signal value, the SVP_AO channel performs

- Direct/Reverse Output Function
- Nonlinear Output Characterization

Determining Output Characterization

Output characterization can be implemented only if the OPACTION parameter is configured as "FullValue." The OP value is calculated when the OPACTION parameter is configured as "FullValue" as follows:



AO block execution diagram when OPACTION is "FullValue"

Determining Direct/Reverse Output

The OPTDIR (output direction) parameter allows you to specify whether the output of the data point is "Direct or Reverse" when OPACTION is configured as "FullValue." When OPACTION is configured as "Incremental," the OPTDIR is grayed out and its value is "Direct." The following table lists the OP value mapping based on the OPTDIR value.

OPTDIR value	OP value mapping
Direct	OP 0% maps to OPLOCURRENT
	OP 100% maps to OPHICURRENT
Reverse	OP 0% maps to OPHICURRENT
	OP 100% maps to OPLOCURRENT

AO OPTDIR processing

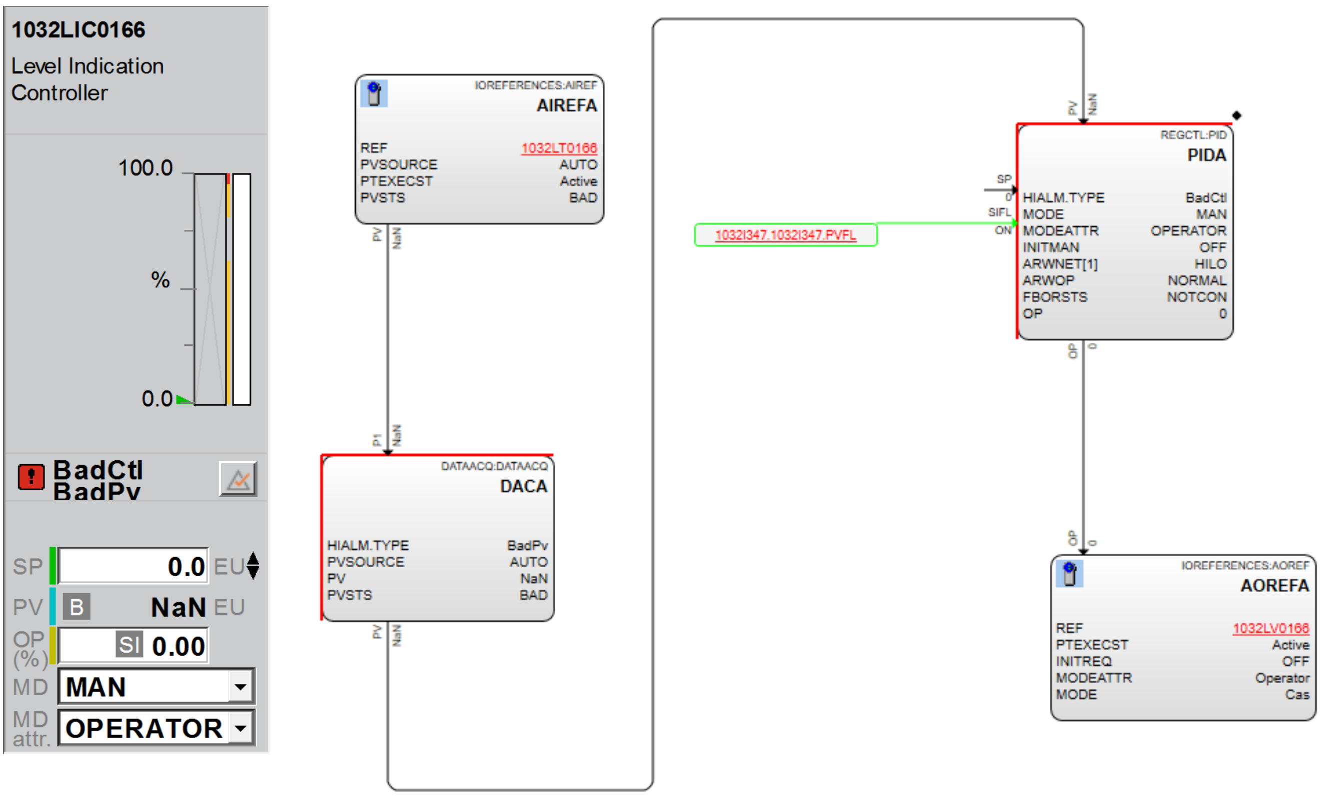
Determining Modes

The MODE parameter determines the operating mode for the channel block. The following operating modes are applicable to the both AO and DO channel blocks:

- Manual (Man) - provides the operator or the program with direct control over the output value of the channel, regardless of any continuous control strategy.
- Cascade (CAS) - data point receives its output value from a primary data point.

However, if the OPACTION is configured as “Incremental” and MODE is changed to “MAN,” the AO drives OPBIASCURRENT, until the operator manually enters a new OP value.

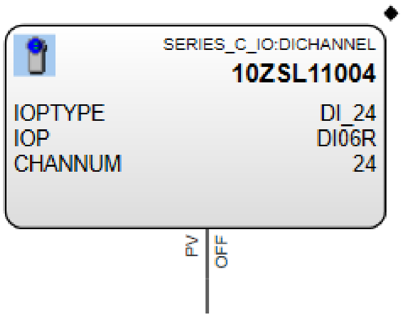
For implementation of fail-safe status in this block , its should be refer to related faceplate or PIDA block , in this case they shown MODE BAD or BAD PV or BAD CTL. For more detail refer to following figures.



9.1.5 DI

Description

Provides a standard digital interface to control blocks.



DI-24

Low Voltage Digital Input (24 volts DC); 32 Channels

This block is used for the following Series C IOMs:

CU-PDIL01

CC-PDIL01

Function

The DI channel block converts a PVRAW signal received from the field to a PV that can be used by other data points in the Experion PKS system.

Inputs

Digital (PV) signals received from the field.

Outputs

PV status value that can be used by other data points in system.

Associated Block

Prior to loading, block must be "associated" with 1 channel of corresponding IOM block that interfaces with physical Digital Input hardware module at execution runtime.

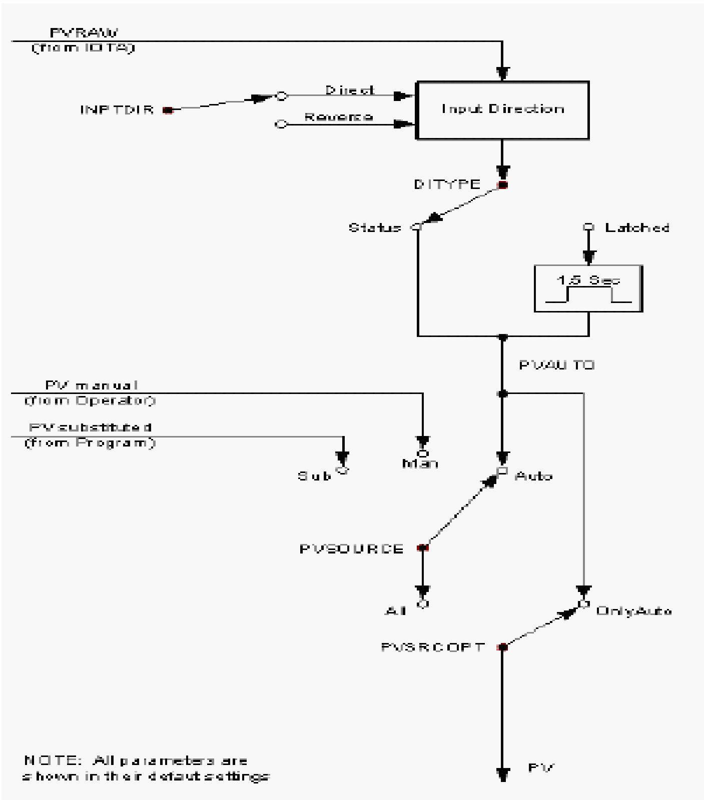
Parameters

ALMOPT	IOP
BADPVFL	IOPTYPE
CHANNUM	PNTFORM
CONTAINEDIN	PNTTYPE
DEBOUNCE	PTEXECST
DEVICELOCATION	PV
DITYPE	PVSOURCE
DLYTIME	PVSRCOPT
EVTOPT	REDTAG

9.1.6 DI Channel

Overview

The DI channel block represents a single digital input point on the SVP – DI channel block. The SVP – DI channel block converts a digital PVRAW signal received from the field to a PV that can be used only for Turbine Control solutions.



Speed Digital Input conversion

Determining Status Digital Input Channel

For this digital input type, the PVAUTO value represents the state of the raw input signal after the direct/reverse conversion is performed. The status digital input channel is selected by setting the DITYPE parameter to “Status,” and this block can be configured for PV source selection.

The current PV state is available as an input to logic blocks and other Experion PKS control function blocks. It is also available for SP – DO permissive check configuration parameters.

PV Source Selection – The PV source parameter (PVSOURCE) option determines the source of the PV for a status input channel. The source can be

- PV input from the field (PVRAW),
- PV state entered by the operator (PV manual), or

- PV supplied by a user program (PV substituted).

PVSOURCE has no effect on the DITYPE of the digital input channel if PVSOURCE is AUTO, and PV tracks PVRAW.

Determining Latched Digital Input Channel

To capture the occurrence of momentary digital inputs, such as from pushbuttons, the digital input channel is configured as a latched input.

Configuring the channel as latched is accomplished by setting:

- DITYPE to "Latched"

When the digital input channel is configured as a latched input channel, an input pulse that is ON for a minimum of 5 milliseconds is latched TRUE for 1.5 seconds. This ensures that any control function block, that needs to monitor this input, executes at least once during the time that the signal is latched.

Open Wire Detection

The SP – DI channel supports open wire diagnostics to detect and annunciate broken field wires. In addition, PV is displayed as "BAD" to prevent incorrect control action.

Ensure that a bleed resistor (~ 22kΩ) resistor is installed at the switching device providing the switched signal. If this resistor is not installed and open wire detection is enabled (OWDENBL = ON), a false open wire alarm is generated whenever the input device is not closed (i.e., PVRAW = OFF).

If open wire detection is enabled and the IOM detects the broken-wire condition,

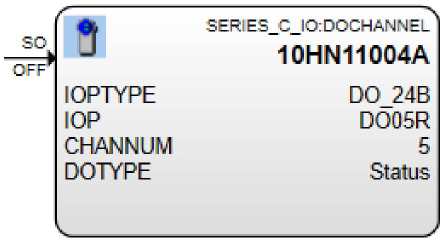
- Soft Failure 179 "Open Wire Detected" is generated, and
- PVRAW and PVAUTO is set to OFF

The DI channel supports PV flag with reset option, which can be used as an interlock option in the SPD – DO channel block.

9.1.7 DO

Description

Generates status output [0 or 1), pulsed output (ON or OFF) for specified pulse time based on origin of input and parameters.



DO-24B

Bussed Low Voltage Digital Output (24 volts DC); 32 Channels

This block is used for the following Series C IOMs:

CU-PDOB01

CC-PDOB01

Function

The DO channel block provides a digital output to the field based on the origin of the input and the configured parameters.

Inputs

SO, PO, ONPULSE, or OFFPULSE value from

- a single Regulatory Control block
- an operator input
- a program
- an SCM block

Outputs

Digital (Boolean) value or pulsed (real) value.

Associated Block

Prior to loading, block must be "associated" with 1 channel of corresponding IOM block that interfaces with physical DO hardware module at execution runtime.

Parameters

CHANNUM **COMMFAILFL** **CONTAINEDIN** **DEVICELOCATION** **DOSTYPE** **DOTYPE** **FAULTOPT** **FAULTVALUE**	**INITREQ** **IOP** **IOPTYPE** **OP** **PNTFORM** **PNTTYPE** **PTEXECST** **REDTAG** **SO** **SOREADFAIL**

9.1.8 DO Channel

Overview

The digital output channel provides a digital output to the field, based on the origin of the input and the configured parameters.

There are four types of digital output points:

- status output (SO) - the default type,
- pulse-width modulated (PWM) output,
- pulse-on output (PULSEON), and
- pulse-off output (OFFPULSE)

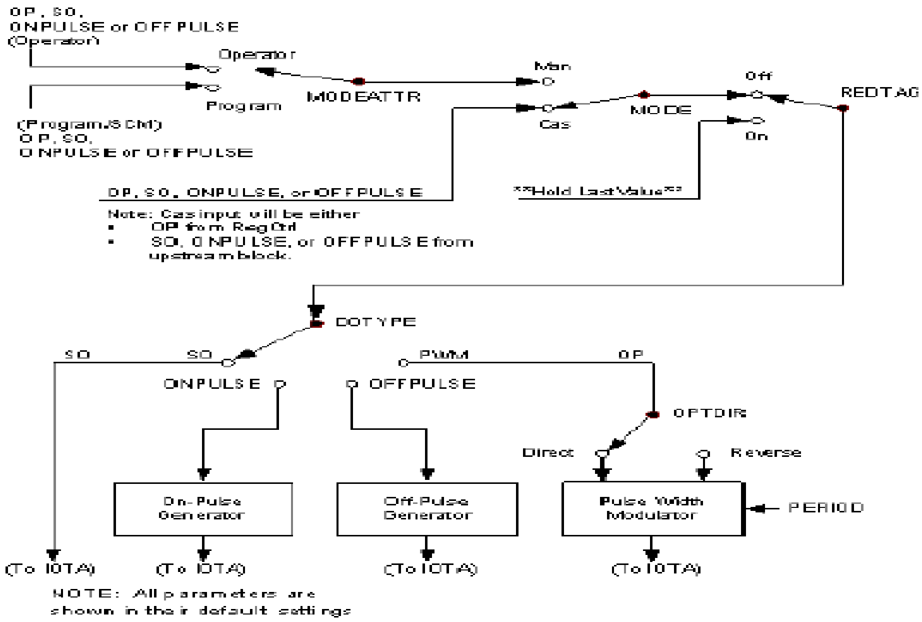
The DOTYPE parameter determines the output type. The PWM type is used in combination with RegCtl algorithms to provide true proportional control. The status and pulsing output types are for digital outputs that are connected to device control blocks.

Actual output action can be:

- status,
- latched or
- momentary

It is dependent on the configuration of the device control point.

A functional diagram of the digital output channel is displayed below.



Digital output conversion

Determining Status Output type

The status output type can be controlled from a:

- device control block output,
- logic block output, or
- RegCtl block (that has been configured for the PosProp algorithm)

as determined by the parameter connection.

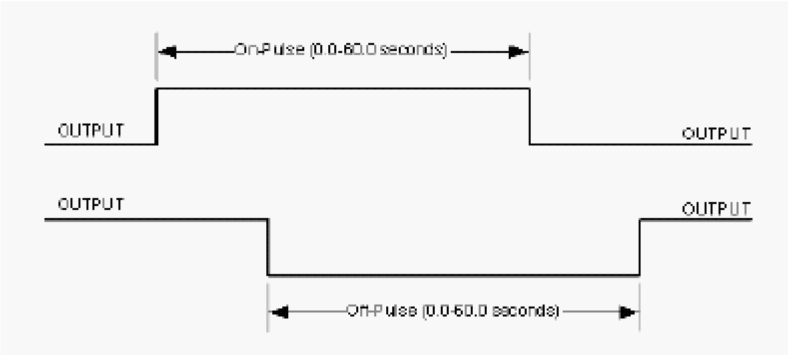
Determining On-Pulse and Off-Pulse Output type

The On-Pulse and Off-Pulse output types can be controlled from:

- a device control block output,
- a logic block output, or
- a RegCtl block (that has been configured for the PosProp algorithm) as determined by the parameter connection.

Pulsed operation (pulse on or pulse-off) can be obtained by linking the output connections to the ONPULSE and

OFFPULSE parameters, respectively.



On-Pulse and Off-Pulse Output types

The Status Output (SO) setting is impacted by the ONPULSE and OFFPULSE parameters as indicated in the following table.

Parameter	Status Output (SO) settin	If SO is set to 0.0
ONPULSE	On - for specified duration	SO is immediately set to OFF
OFFPULSE	Off - for specified duration	SO is immediately set to ON

Setting DOTYPE to ONPULSE or OFFPULSE

To provide consistent and safe behavior, the following occurs when setting DOTYPE to either ONPULSE or OFF PULSE.

If	Then
MODE is MAN	the ONPULSE and OFFPULSE parameters only accept operator writes. (Program access level writes and NOT all other writes are accepted.) operator access level writes to SO are accepted and writing SO terminates an active pulse.
PTEXECST is ACTIVE	changing MODE to CAS sets the output to the quiescent state
PTEXECST is ACTIVE	always sets the output to the quiescent state

Setting DOTYPE to ONPULSE or OFFPULSE

Determining Modes

The MODE parameter determines the operating mode for the channel block. The following operating modes are applicable to the both DO and AO channel blocks:

- Manual (Man) - provides the operator or the program with direct control over the output value of the channel, regardless of any continuous control strategy.
- Cascade (CAS) - data point receives its output value from a primary data point.

Determining Output Verification

Outputs are verified by periodically reading back the value on the output screw and comparing the read back value with the database value. This includes an output wiggle (for the safety system) to prove they are not stuck in any one state.

Determining Over-current protection

Digital outputs are protected from inadvertent over-current conditions. If a DO channel consumes more current than it should the IOM posts a soft failure and sheds to manual control. Supervisor intervention is required to return the channel to normal operation. Over-current conditions are typically the result of a shorted device or capable.

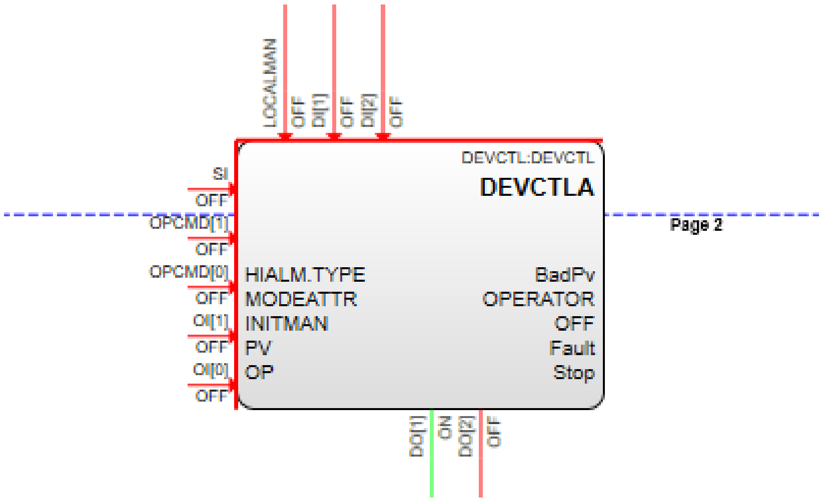
Comparing parameters between Series C and PMIO that support DO

The following parameters are:

- specific to DO and found on various tabs on the DO channel block.
- work identically to the same named PM I/O counterparts.

DO supported parameters for current Series C I/O that work identically to the same named PM I/O parameters		
OPTDIR	SOINITVAL	COMMFAILFL
SO	ONPULSE, OFFPULSE	PERIOD
SOREADFAIL	OP, OPINITVAL	

For implementation of fail-safe status in this block , its should be refer to related faceplate or DEVCTLA block , in this case they shown MODE BAD or BAD PV or BAD CTL. For more detail refer to following figures.



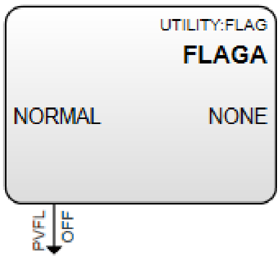
9.10 FUNCTIONAL BLOCK TYPES

9.11.1FLAG

Description

The FLAG function block provides storage for a single 2-state value. The value can be accessed as a simple Boolean (Off or On) using the PVFL parameter, or as one of two user-configured State values (for example, Running and Stopped) through the PV parameter. It looks like this graphically.

Supports pulse length measurement in all channels.	Does not support pulse length measurement in the last two channels.
Supports configuration of all eight channels through use of PICHANNEL block.	Needs two blocks for configuring channels: • Pulse Input Channel block for configuring the first six channels. • Pulse Input Channel with Fast Cutoff block for configuring the last two channels.
The last two channels can be configured as Pulse Input channel types or Pulse Input with Fast Cutoff channel types.	The last two channels can be configured only as Pulse Input with Fast Cutoff channel types.



Function

Used to define two separate states (for example, Running/Stopped, Off/On) to indicate status of a particular input.

There are 2 user-configurable state descriptors, STATETEXT[0] and STATETEXT[1] which are used to describe STATE0 and STATE1 respectively.

Current state of flag can be changed/read using PVFL (Boolean) or using PV (either STATETEXT[0] or STATETEXT[1]).

Block also supports:

configurable access lock which determines who can write a value to the block (such as operator, engineer, or other function block). an Off-Normal Alarm whereby one of the flag's states is configured as the normal state; whenever the flag changes state, the Off-Normal Alarm is generated.

Input/Output

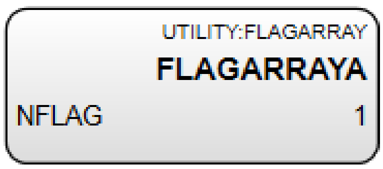
The block has one output flag (PVFL). But, all block pin parameters are available to be exposed and connected to using Control Builder graphical connections.

9.10.2 FLAGARRAY

Description

The FLAGARRAY function block provides storage for up to 1000 2-state values. The value can be accessed as a simple Boolean (Off or On) using the PVFL[n] parameter.

Where "n" is the number of the flag. It looks like this graphically:



Function

Used to define two separate states (Off/On) to indicate status of a particular input.

Number of flag values (NFLAG) is user configurable.

Current state of flags can be changed/read using flag value (PVFL[n]) (Boolean).

Block also supports configurable access lock which determines who can write a value to the block (such as an operator, engineer, or other function block).

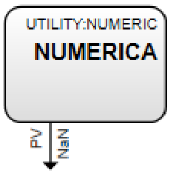
Input/Output

The block has up to 1000 output flags(PVFL[n]). But, all block pin parameters are available to be exposed and connected to using Control Builder graphical connections.

9.10.3 NUMERIC

Description

The NUMERIC block provides storage for a floating-point value which is accessible through the PV configuration parameter. It looks like this graphically.



Function

Used to store up to 8 bytes of a floating point value within defined upper and lower limits for use in a control strategy.

Configurable high and low limits are also provided.

Also supports a configurable access lock which determines who can write a value to the block (such as operator, engineer, other function block).

Input/Output

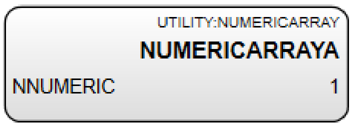
The block has one output (PV). But, all block pin parameters are available to be exposed and connected to using Control Builder graphical connections.

9.10.4 NUMERICARRAY

Description

The NUMERICARRAY block provides storage for up to 200 floating point values which are accessible through the corresponding PV configuration parameter (PV[n]).

Where "n" is the number of the numeric. It looks like this graphically:



Function

The NUMERICARRAY block outputs (PV[n]) can be used as source parameters to provide predefined analog constants to other function blocks. A bad numeric output parameter typically has the value NaN (Not-a-Number). The block supports these user configurable attributes.

A configurable Access Lock (ACCLOCK) which determines who can write a value to the block (such as operator, engineer, or other function block).

A configurable PV Format (PVFORMAT) which lets you select the decimal format to be used to display the PV[n] values. The selections are D0 for no decimal place (-XXXXXX.), D1 for one decimal place (-XXXXX.X), D2 for two decimal places (-XXXX.XX), and D3 for three decimal places (-XXX.XXX). The default selection is D1 for one decimal place.

A configurable Number of Numeric Values (NNUMERIC) which lets you specify the desired number of numeric values to be supported.

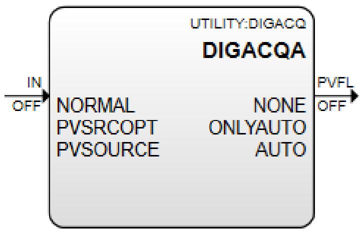
Input/Output

The block has up to 200 outputs (PV[n]), depending on the number of numeric values (NNUMERIC) configured. But, all block pin parameters are available to be exposed and connected to using Control Builder graphical connections.

9.10.5 DIGACQA (Digital Acquisition)

Description

The Digital Acquisition function block, available under "Utility" in the function block library, uses a combination of a DICHANNEL and SEL/FLAG when PVSOURCE is defined by the operator. The Digital Acquisition block receives input from DI Channel block and allows the user to define the PVSOURCE. This block is independent of the Channel type feeding the block. The following is a graphical representation of the Digital Acquisition function block.



Inputs

IN - Input parameter.

Outputs

PV - Current selected input based on the PVSOURCE selection

PVFL - Actual State Flag

INVPVFL - Inverted State Flag

Function

Enables better utilization of processor computing and memory resources.

Supports alarm generation when the current process variable state differs from the configured NORMAL state.

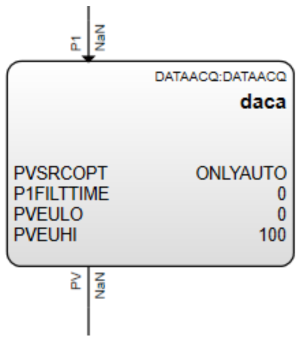
Parameters

BADPVALM.FL	OFFNRMALM.SV
BADPVALM.PR	PV
BADPVALM.SV	PVAUTOFL
DABLOCKSINCM	PVFL
HIALM.PR	PVSOURCE
HIALM.SV	PVSRCOPT
HIALM.TYPE	PVSTS
IN	PVSTSFL.BAD
INALM	PVSTSFL.MAN
INVPVFL	PVSTSFL.NORM
NAME	PVSTSFL.UNCERTN
NORMAL	STATE0
OFFNRMALM.FL	STATE1
OFFNRMALM.PR	STATETEXT[0..1]

9.10.6 DACA (Data Acquisition)

Description

The DATAACQ (Data Acquisition) block processes a specified process input value (P1) into a desired output value (PV). It looks like this graphically.



Function

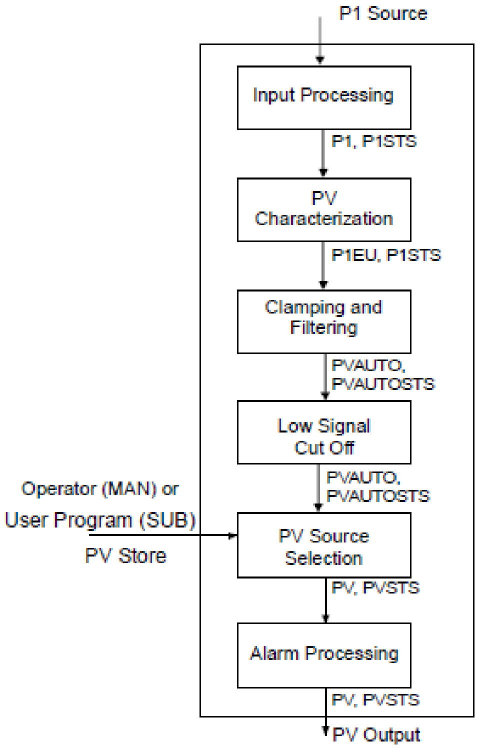
The DATAACQ block is normally configured to fetch an analog input from an AI Channel function block. As shown in the following figure, it performs the following major functions:

- Input Processing - fetches input data from another block through process connections, checks its validity, and updates input parameters P1 and P1STS as appropriate.
- PV Characterization - converts input parameter P1 to Engineering Units, when the user configurable PV Characterization option is configured as Linear or Square Root. The converted P1 value is stored in a read-only parameter (P1EU).

- Filtering and Clamping - performs filtering and clamping on the read-only parameter P1EU and stores the result in PVAUTO. There are user configurable parameters associated with both the filtering (P1FILTTIME) and clamping (P1CLAMPOPT) functions.
- Low Signal Cut Off - Applies a user configurable low signal cut off limit to the PVAUTO value after filtering and clamping.
- PV Source Selection - normally copies the filtered and clamped value of PVAUTO to the output PV, but also allows for instances where the operator or user program can store to PV, if the user configurable PV Source selection is configured for MAN or SUB, respectively.

- Alarm Processing - generates alarm flags when PV exceeds any of a number of user specified alarm trip points for longer than a designated time interval.

These functions are discussed in more detail in the following paragraphs.



Input

The DATAACQ block requires one process input value - P1. P1 must be brought from another block.

Input ranges and limits

PVEUHI and PVEULO define the full range of P1 in engineering units.

PVEUHI represents the 100% of full-scale value.

PVEULO represents the 0% of full-scale value.

PVEXHILM and PVEXLO.LM define the high and low limits of P1 in engineering units.

If P1 clamping is desired (P1CLAMPOPT = Enable), the DATAACQ block clamps the input within the range defined by PVEXHILM and PVEXLOLM:

Alarm processing

The DATAACQ block may be configured to generate an alarm when PV exceeds one of the following trip points (HIHI, HI, LOW, LOWLOW) for more than a specified time:

Bad PV alarm

The DATAACQ block may be configured to generate a "Bad PV" alarm if PV = NaN (Not a Number).

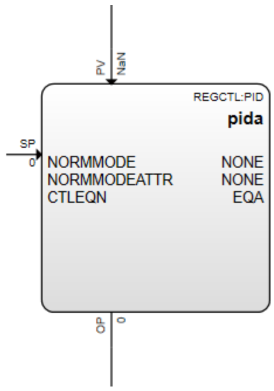
Parameter

ALMDB	PVHHALM.TM
ALMDBU	PVHHALM.TP
ALMTM	PVHIALM.DB
BADPVALM.FL	PVHIALM.DB
BADPVALM.PR	PVHIALM.DBU
BADPVALM.SV	PVHIALM.FL
DESC	PVHIALM.PR
EUDESC	PVHIALM.SV
HIALM.PR	PVHIALM.TM
HIALM.SV	PVHIALM.TP
HIALM.TYPE	PVHISIGCHG.CT
INALM	PVHISIGCHG.TP
INSBLOCK[1..10]	PVLLALM.DB
INSFAILFL	PVLLALM.DBU
INSFAIL.PR	PVLLALM.FL
INSFAIL.SV	PVLLALM.PR
LASTGOODPV	PVLLALM.SV
LOCUTOFF	PVLLALM.TM
NAME	PVLLALM.TP
ORDERINCM	PVLOALM.DB
NUMINSERT	PVLOALM.DBU
P1	PVLOALM.FL
P1CLAMPOPT	PVLOALM.PR
P1EU	PVLOALM.SV
P1FILTINIT	PVLOALM.TM
P1FILTTIME	PVLOALM.TP
P1STS	PVLOSIGCHG.CT
PV	PVLOSIGCHG.TP
PVAUTO	PVP
PVAUTOSTS	PVSOURCE
PVCHAR	PVSRCOPT
PVEUHI	PVSTS
PVEULO	PVSTSFL.BAD
PVEXHIFL	PVSTSFL.MAN
PVEXHILM	PVSTSFL.NORM
PVEXLOFL	PVSTSFL.UNCER
PVEXLOLM	PVVALSTS
PVFORMAT	ROCNEGALM.FL
PVHHALM.DB	ROCNEGALM.PR
PVHHALM.DBU	ROCNEGALM.SV
PVHHALM.FL	ROCNEGALM.TP
PVHHALM.PR	ROCPOSALM.FL
PVHHALM.SV	ROCPOSALM.PR
	ROCPOSALM.SV
	ROCPOSALM.TP

9.10.7 PID Controller (PIDA)

Description

The PID block is a regulatory control block that operates as a proportional-integral derivative (PID) controller. It supports the Ideal form of calculating the PID terms. The Ideal form is often called the digital-computer version of the PID controller. The PID block looks like this graphically:

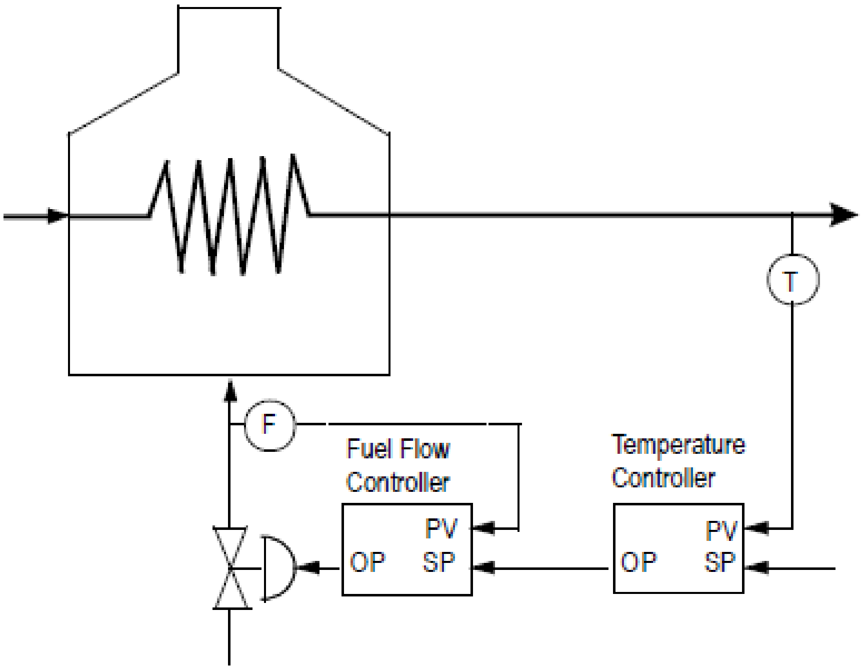


The PID block has two analog inputs - a process variable (PV) and a set point (SP). The difference between PV and SP is the error and this block calculates a control output (OP) that should drive the error to zero.

The following equations are supported:

- Proportional, Integral, and Derivative (PID) on the error
- Proportional and Integral (PI) on the error and Derivative (D) on changes in PV
- Integral (I) on the error and Proportional and Derivative (PD) on changes in PV
- Integral (I) only
- Proportional (P) only

The PID block may be used in a single control loop or with multiple PIDs in a cascade strategy. The following figure shows two PID controllers being used for simple cascade control where the output of a temperature controller is used as the set point of a flow controller.



Function

A PID requires two inputs - a process variable (PV) and a set point (SP):

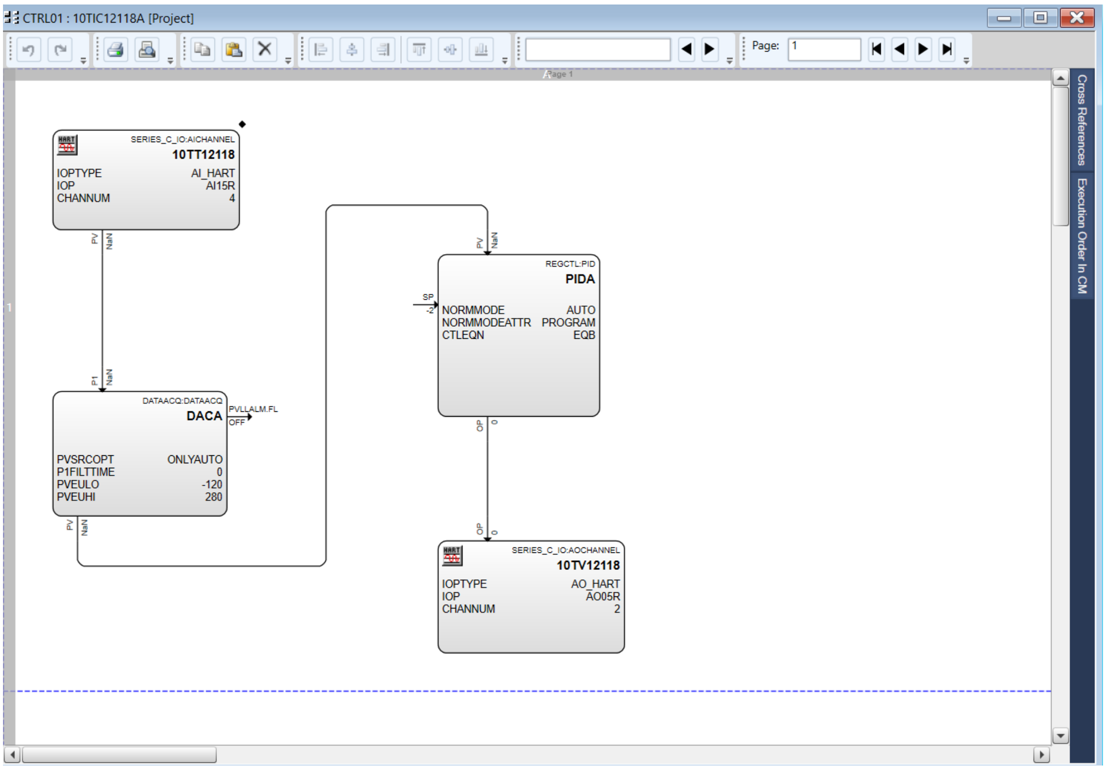
PV is typically pulled from a Data Acquisition (DATAACQ) function block which performs PV limit checking and alarming.

SP is pulled from another function block, or stored by the operator or a user program.

If SP is pulled from a primary, the PID's Mode must be Cascade; and if it is stored by the operator or a user program, Mode must be manual or automatic. If Mode is Cascade, the PID must perform timeout checking on SP (to make sure the primary is periodically updating it).

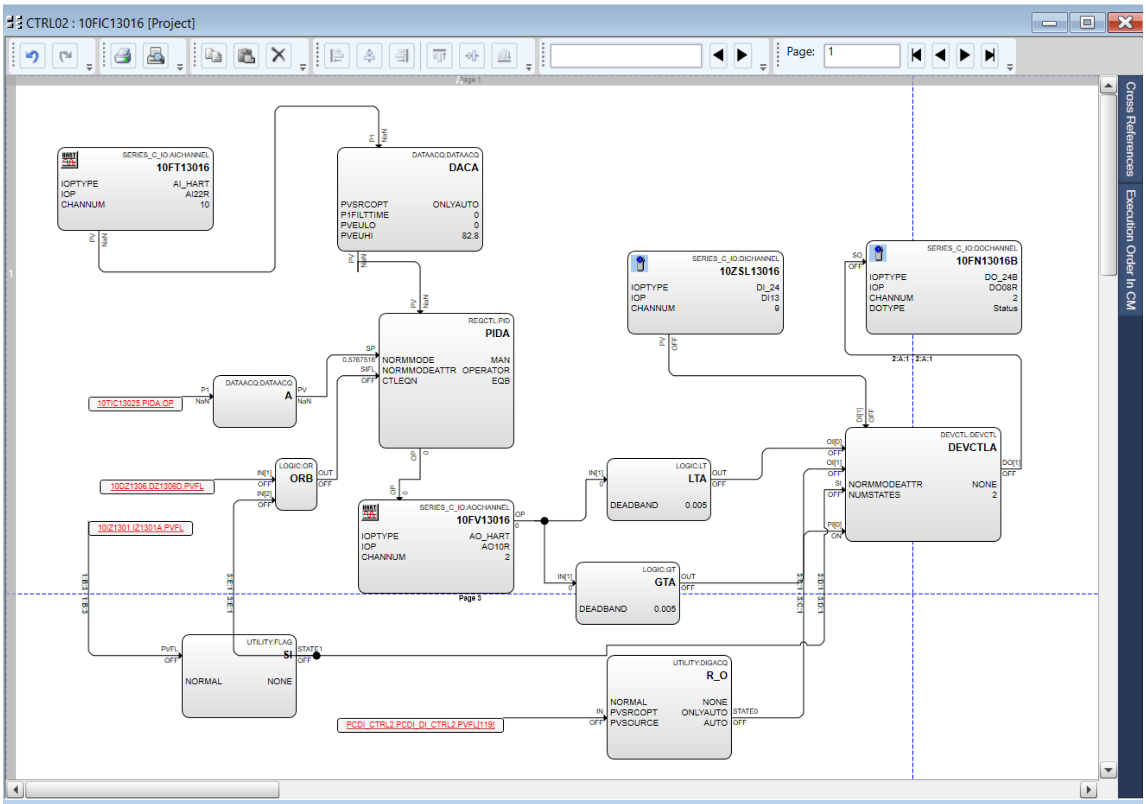
Single PID Loop

The following figure and its companion callout description table show a sample configuration that uses a PID block to form a single control loop for quick reference. The view in following figure depicts a loaded configuration in Monitoring mode.



Cascade PID Loop

The following figure and its companion callout description table show a sample configuration that uses two PID blocks to form a cascade control loop for quick reference. The view in the following figure depicts a loaded configuration in Monitoring mode



Operating modes and mode handling

The PID block operates in the following modes:

MAN (Manual)

If mode is Manual, OP may be stored by the operator or a user program; PV and SP are ignored - if a primary exists, it goes to the initialized state.

AUTO (Automatic)

If mode is Automatic, SP (or SPP) may be stored by the operator or a user program; if a primary exists, it goes to the initialized state. SP contains set point value in engineering units and SPP contains the value in percent.

CAS (Cascade)

If mode is Cascade, SP is pulled from a primary; if the primary is off-control (that is, inactive or initializing) or the connection is bad, the PID block invokes timeout processing.

Required inputs

The required number of inputs is determined by the mode of the PID block.

If Mode is Cascade, two inputs are required - PV and SP.

If Mode is Automatic or Manual, only PV is required.

SP is an initializable input; PV is non-initializable.

PV must be pulled from another block; you cannot store to it - typically it is connected to the output of an auxiliary or data acquisition (DATAACQ) block.

If Mode is Cascade, SP is pulled from another block; if Mode is Automatic, it may be stored by the operator or a user program.

The PID block may have one primary or none, depending on whether SP is configured or not; there is one primary per initializable input.

Input ranges and limits

You must specify a PV engineering unit range, PVEUHI and PVEULO.

PVEUHI and PVEULO define the full range of PV in engineering units.

PVEUHI represents the 100% of full scale value.

PVEULO represents the 0% of full scale value.

PVEUHI and PVEULO also define the engineering unit range of SP - PV and SP are assumed to have the same range.

The PID block assumes PV is within PVEUHI and PVEULO - it applies no range check - however, PV typically comes from a data acquisition (DATAACQ) block which applies its own limit and range check.

SPHILM and SPLOLM define set point operating limits in engineering units.

The operator is prevented from storing a set point value that is outside these limits; if the primary or a user program attempts to store a value outside of the limits, the PID block clamps it to the appropriate limit and sets the primary's windup status.

SP contains set point value in engineering units and SPP contains the value in percent.

If Mode is Automatic, the operator or a user program may store to either SP or SPP.

Output ranges and limits

CVEUHI and CVEULO define the full range of CV in engineering units.

If the PID block has a secondary, its CV range must be the same as the secondary's input range - if this PID function has a secondary, it brings the secondary's input range through BACKCALC and sets its CV range to that.

If the PID block has no secondary, CVEUHI and CVEULO must be specified.

Direct or reverse control

A PID block may be configured for direct-control action or reverse-control action.

Changing the control action effectively changes the sign of the gain.

With direct-control action, an increase in the error (PV - SP) increases the PID output (CV).

With reverse-control action, an increase in the error (PV - SP) decreases the PID output (CV).

For example, if SPP = 50% and PVP = 51%, then the error is 1%. With direct-control action, if PVP changes to 52%, the error increases causing CV to increase.

With reverse-control action, if PVP changes to 52%, the error increases causing CV to decrease.

Control equation type and PID coefficient

Default type of controller for all loops is EQB in control equation type.

All of type are shown in following figure:

Control Equation Type

Select the appropriate equation for the application as follows:

- Equation A:

  all three terms (proportional, integral and derivative) act on error.
  Error = (Process Variable - Set Point)

- Equation B:

  proportional and integral terms act on error and derivative term acts on changes in process variable (PV).
  used to eliminate derivative spikes in control action as result of quick changes in set point (SP).
  Error = (Process Variable - Set Point)

- Equation'C:

  integral term acts on error and proportional and derivative terms act on changes in process variable (PV).
  provides the smoothest and slowest response to set point (SP) changes.
  Error = (Process Variable - Set Point)

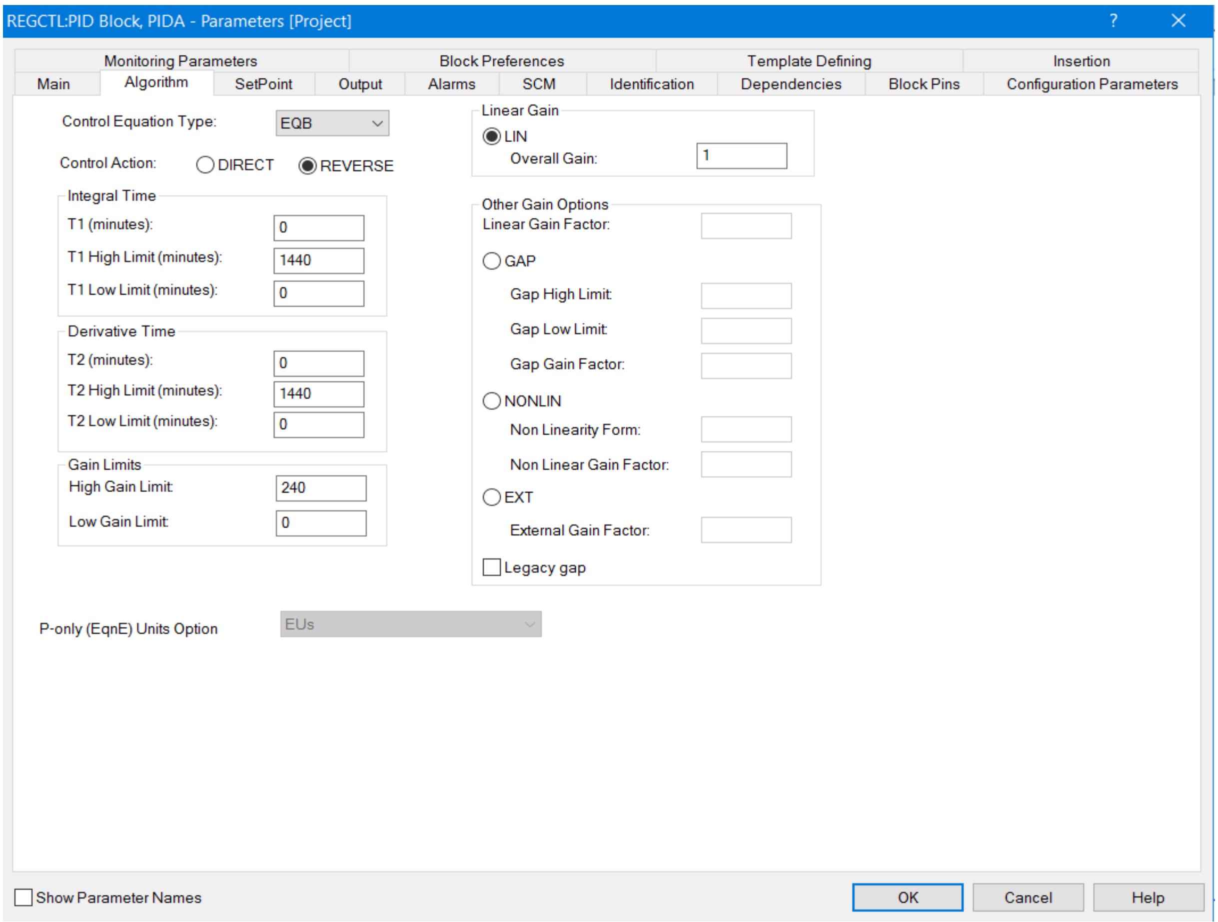
- Equation D:

  this equation only provides integral control.

- Equation E:

  this equation only provides proportional control.

and preliminary values for PID coefficient are shown in following figure:



Parameters

ADVDEVALM.DB	OPBIAS
ADVDEVALM.DBU	OPBIAS.FIX
ADVDEVALM.FL	OPBIAS.FLOAT
ADVDEVALM.PR	OPBIAS.RATE
ADVDEVALM.SV	OPEU
ADVDEVALM.TM	OPEXHIFL
ADVDEVALM.TP	OPEXHILM
ADVDEVOPT	OPEXLOFL
ADVSP	OPEXLOLM
ADVSPP	OPHIALM.DB
ALMDB	OPHIALM.DBU
ALMDBU	OPHIALM.FL
ALMTM	OPHIALM.PR
ARWNET[1..8]	OPHIALM.SV
ARWOP	OPHIALM.TM
ASTEPID	OPHIALM.TP
BADCTLALM.FL	OPHIFL
BADCTLALM.PR	OPHILM
BADCTLALM.SV	OPLOALM.DB
BADCTLFL	OPLOALM.DBU
BADCTLOPT	OPLOALM.FL
BADOCOPT	OPLOALM.PR
BADOCOPTENB	OPLOALM.SV
CASREQFL	OPLOALM.TM
COMPUTEARW	OPLOALM.TP
CTLACTN	OPLOFL
CTLEQN	OPLOLM
CTLINIT	OPMINCHG
CTLSTATE	OPREQ
CV	OPROCLM
CVEUHI	OPROCNEGFL
CVEULO	OPROCPOSFL
CVTYPE	OPTYPE
DELCV	ORDERINCM
DESC	OUTIND
DEV	OUTTYPE
DEVHIALM.DB	PRIM.[1..8].INITIALIZABLE
DEVHIALM.DBU	PRIMDATA.[1..8].HISELECT
DEVHIALM.FL	PRIMDATA.[1..8].ORFBSTS
DEVHIALM.PR	PRIMDATA.[1..8].ORFBVAL
DEVHIALM.SV	PRIMDATA.[1..8].OROFFSET
DEVHIALM.TM	PRIMDATA.[1..8].PROPOVRD

DEVHIALM.TP	PUSHSP
DEVLOALM.DB	PV
DEVLOALM.DBU	PVEUHI
DEVLOALM.FL	PVEULO
DEVLOALM.PR	PVFORMAT
DEVLOALM.SV	PVMANOPT
DEVLOALM.TM	PVP
DEVLOALM.TP	PVSTS
EQNEUNITSOPT	PVSTSFL.BAD
ESWENB	PVSTSFL.MAN
ESWFL.AUTO	PVSTSFL.NORM
ESWFL.BCAS	PVSTSFL.UNCER
ESWFL.CAS	PVTRAKOPT
ESWFL.MAN	PVTRAKOPTAI
ESWPERM	REDTAG
EUDESC	RESTARTOPT
FBORSTS	SAFEOP
GAINHILM	SECDATAIN.ARWSTS
GAINLOLM	SECDATAIN.EUHI
GAINOPT	SECDATAIN.EULO
GAPHILM	SECDATAIN.HISELECT
GAPLOLM	SECDATAIN.INITSTS
HIALM.PR	SECDATAIN.INITVAL
HIALM.SV	SECDATAIN.LOCALMAN
HIALM.TYPE	SECDATAIN.ORFBSTS
HOLDOPT	SECDATAIN.ORFBVAL
HOLDRATE	SECDATAIN.OROFFSET
HOLDVAL	SECDATAIN.PROPOVRD
INALM	SECINITOPT[1..8]
INITMAN	SIALM.FL
INITREQ[1..8]	SIALM.OPT
INITVAL[1..8]	SIALM.PR
INSBLOCK[1..10]	SIALM.SV
INSFAILALM.FL	SIFL
INSFAILALM.PR	SIOPT
INSFAILALM.SV	SP
INSFAILFL	SPEUHI
K]	SPEULO
KLIN	SPFORMAT
KMODIFEXT	SPHIFL
KMODIFGAP	SPHILM
KMODIFNL	SPLOFL
LASTGOODPV	SPLOLM
LASTMODEREQ	SPP
LASTOPREQ	SPRATEREQ
LASTOPTYPE	SPREQ

LASTRATEREQ	SPTV
LASTREQFL	SPTVDEVFL
LASTSPREQ	SPTVDEVMAX
LASTSPTVREQ	SPTVNORMRATE
LASTSTEP	SPTVOPT
LEGACYGAP	SPTVP
MODE	SPTVRATE
MODEAPPL[1..4]	SPTVREQ
MODEATTR	SPTVSTATE
MODEATTRFL.NORM	SPTVTIME
MODEATTRFL.OPER	STARTOPT
MODEATTRFL.PROG	STARTRATE
MODECHANGE	STARTVAL
MODEFL.AUTO	STOPOPT
MODEFL.BCAS	STOPRATE
MODEFL.CAS	STOPVAL
MODEFL.MAN	T1
MODEFL.NORM	T1HILM
MODEPERM	T1LOLM
MODEREQ	T2
MODETRACK	T2HILM
NAME	T2LOLM
NLFORM	TMOUTFL
NLGAIN	TMOUTMODE
NORMMODE	TMOUTTIME
NORMMODEATTR	UNCMDCHGALM.FL
NUMINSERT	UNCMDCHGALM.OPT
NUMPRI	UNCMDCHGALM.PR
NUMSEC	UNCMDCHGALM.SV
OP	

Some parameters of Algorithm tab

1.KLIN

Specific to Block(s)	PID, PIDFF
Description	Linear Gain factor. Used if GAINOPT = Gap, Nonlin, or Ext(configured).
Data Type	64-Bit Real Number
Range	GAINLOLM to GAINHILM
Default	1.0
Config Load	Yes
Active Loadable	No
Access Lock	Engineer
Residence	CEE
Related Parameters	GAINHILM, GAINLOLM.

2.GAPHILM

Specific to Block(s)	PID, PIDFF
Description	Gap High Limit. Used if GAINOPT = Gap (configured).
Data Type	64-Bit Real Number
Range	N/A
Default	0.00
Config Load	Yes
Active Loadable	No
Access Lock	Engineer
Residence	CEE
Related Parameters	

3.NLFORM

Specific to Block(s)	PID, PIDFF
Description	Non-linear Gain form. Used if GAINOPT = Nonlin (configured)
Data Type	Integer
Range	0 or 1
Default	1
Config Load	Yes
Active Loadable	No
Access Lock	Engineer
Residence	CEE

4.KMODIFEXT

Specific to Block(s)	PID, PIDFF
Description	External Gain Modifier. Used if GAINOPT = Ext (configured).
Data Type	64-Bit Real Number
Range	GAINLOLM to GAINHILM
Default	1.0
Config Load	Yes
Active Loadable	No
Access Lock	Program
Residence	CEE
Related Parameters	GAINHILM,GAINLOLM.

9.10.8 SEQUENTIAL CONTROL MODULE block (Sequential Control)

Description

A system container block that consists of sequences of STEP and TRANSITION blocks grouped by specific HANDLER blocks.

- The SCM block may only contain its own components (that is, HANDLER, STEP and TRANSITION blocks); it cannot contain other basic blocks such as PID or logic blocks.

Function

Used to organize normal- and exception-based sequential control logic.

- **HANDLER Block**

Description

SCM HANDLER blocks are execution modules that group STEP and TRANSITION blocks.

- Multiple Handler blocks may be contained within an SCM block, each modeled as a set of STEP and TRANSITION blocks, based on the following categories:
  - Edit Handler
  - Main Handler
  - Check Handler
  - Interrupt Handler
  - Restart Handler
  - Hold Handler
  - Stop Handler
  - Abort Handler
- Choices of which HANDLER block of each category to invoke are manifested through a HANDLER block selection list on the SCM block.
- A HANDLER block is invoked when
  - its invoke conditions, modeled in its Invoke TRANSITION block, are met

- when the SCM block is commanded to invoke the Handler (for example, the STOP command causes the STOP Handler to execute)

- .

Function

Used to describe, group, and categorize sequential control behavior.

- **STEP Block**

Description

An SCM block which defines specific output actions.

- A specified output action usually generates a request to a control device to do something (for example, open a valve, start a pump, set furnace temperature).
- The source value of each output can be an expression (thereby enabling calculations in each output).

Function

Organizes the output expressions of an SCM HANDLER block at a specific stage of the HANDLER's execution thread.

Outputs

- Up to 16 outputs may be defined per SCM STEP block.
- The STEP block whose outputs are active is called the Active Step.

- **TRANSITION Block**

Description

An SCM block that defines specific input conditions for a Handler.

- Input conditions and conjoining TRANSITION blocks define a distinct process state that must be achieved in order for the output actions specified by the next STEP block to be performed.
- The input conditions grouped into a TRANSITION block are the condition expressions that direct sequential execution flow.

Note: Nesting of Transitions may be required when a single Transition cannot accommodate all inputs in required in a logical expression. Transition is considered Free Standing when the input pin, DESC, and the Output pin, NEXTCOMP, are not connected to any other EBM component Block.

For more information on Nesting Transition.

Function

Defines the distinct process state that must be achieved in order to allow the SCM HANDLER to advance to the control step (that is, the STEP block) so that it can perform the output actions specified.

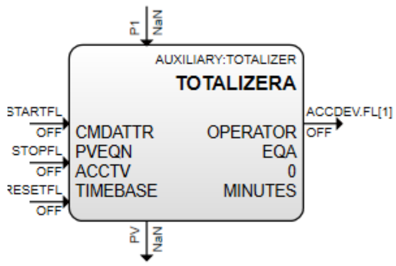
Input conditions

- A maximum of 10 standard input conditions are supported per SCM TRANSITION block.
- The Invoke TRANSITION block in the MAIN HANDLER of the SCM block provides the Start Conditions for the SCM.
- Logic gates may be AND, OR, NAND, NOR, NOT, XOR, CONNECT, NONE, OFF, or ON.
  - XOR must have two inputs.
  - CONNECT and NOT have only one input -- the output is the same as the input and the output is the logical negation of the input, respectively.
  - NONE, ON, and OFF have no inputs.

9.10.9 TOTALIZER

Description

The TOTALIZER block periodically adds an input value (P1) to an accumulator value] (PV). It looks like this graphically:



You specify a target value for the accumulator, and up to four trip points, which are "near" and "nearer to" the target value. The TOTALIZER block sets status flags to indicate when the accumulator value is near (and nearer to) the user-specified target values.

A trapezoidal-integration method of accumulation is used to improve accuracy.

Accumulation proceeds even when the target value is exceeded. An external operator or program command is required to stop the block from further accumulating.

Function

The TOTALIZER block is typically used to accumulate total flows. For situations where the flow transmitter may not be precisely calibrated near the zero-flow value, a zero-flow cutoff feature is provided such that when P1 is below the cutoff value it clamps to zero.

Input

The TOTALIZER block requires one input (P1):

P1 is the value to be accumulated - the input value may be real, integer, or Boolean, but is always stored as a real number.

P1 must be brought from another block.

Outputs

The TOTALIZER block produces the following outputs:

The accumulated value (PV) and its status (PVSTS).

Flags, indicating if the accumulated value has reached the user-specified target value or one of the accumulator deviation trip points (ACCTVFL and ACCDEV.FL [1-4]).

TOTALIZER states

The TOTALIZER block has two possible states: Stopped and Running. The STATE parameter identifies the current state and the following parameters may be used to change the state:

COMMAND

The operator or a user program may command the accumulator to Start, Stop, or Reset by storing to the COMMAND parameter. Since COMMAND is a write-only parameter, its displayed value does not reflect the last entered command.

Possible choices are:

Start - requests the TOTALIZER to start the accumulation (change STATE to Running). The Totalizer block must be reset using the reset pin (RESETFL) prior to counting. Stop - requests the TOTALIZER to stop the accumulation (change STATE to Stopped).

Reset - requests the TOTALIZER to reset the accumulated value (PV) with a user-specified reset value (RESETVAL). STATE will not change; if the accumulator is running, it continues from the reset value.

Totalizer must be reset using the reset pin before the totalizer can start counting. Otherwise P1 will have a good value, but PV will remain at zero.

When the TOTALIZER receives a reset command, it copies the current value of PV to OLDAV (old accumulation value), and then sets PV equal to RESETVAL. This allows other system functions using the totalized value to reset the TOTALIZER without losing any "accumulation".

CMDATTR

Specifies who may store to COMMAND (that is, either the operator or a user program through another function block). CMDATTR is used to prevent the operator from inadvertently changing the accumulator while it is under program control and allows the operator to override a program.

Possible choices are:

Operator - only the operator may store to COMMAND.

Other FB- only a program through another function block may store to COMMAND; the operator may override the program by setting CMDATTR = Operator.

STARTFL (Start Flag): Allows either a Logic block or user-written program to store to COMMAND.

Off-to-On transitions cause the TOTALIZER state to change to Running.

STOPFL (Stop Flag): Allows either a Logic block or user-written program to store to COMMAND.

Off-to-On transitions cause the TOTALIZER state to change to Stop.

RESETFL (Reset Flag): Allows either a Logic block or user-written program to store to COMMAND.

Off-to-On transitions cause the TOTALIZER to be reset.

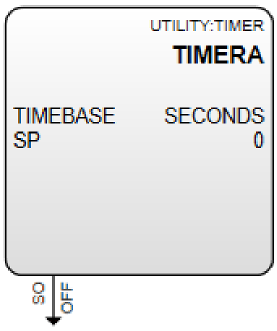
9.10.10 TIMER

Description

The TIMER block provides the capability to time process events or create known delays.

It looks like this graphically.

Function



Used to keep track of elapsed time during a process and provides indication when elapsed time reaches predefined limit. The TIMEBASE can be configured to represent seconds, minutes, or cycles (number of execution cycles).

Input/Output

The block has one status output (SO). But, all parameters are available to be exposed and connected to using Control Builder graphical connections.

Commands

Commands are sent to the timer in one of two ways:

By the operator, using the COMMAND parameter through connections to the parameters STARTFL, STOPFL, RESETFL, and RESTARTFL

You can give a Reset command any time, even if the TIMER is not running, and it will always be executed.

However, the Stop command is only valid while the TIMER is running. For example, giving a Stop command directly after a Reset command is not allowed.

The Start and Restart commands are not interchangeable. A Start command is only executed after a prior Reset, when the timer is starting from the beginning (PV = 0).

Similarly, a Restart command is only executed after a prior Stop command, which froze the timer when it was running (PV usually = non-zero).

When more than one of the Boolean command parameters is set at the same time, the following priority is used:

RESETFL - highest priority

STOPFL

RESTARTFL

STARTFL - lowest priority

For example, when both RESETFL and STARTFL are ON, the TIMER executes the Reset command and nothing else will happen until RESETFL goes Off. This leaves the STARTFL as the only Boolean command ON, at which time the TIMER is started.

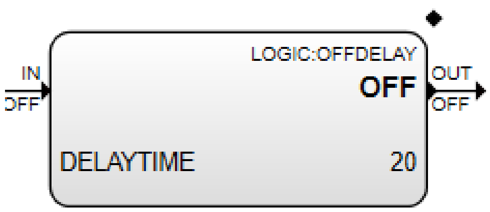
If you use both methods for issuing commands to the TIMER at the same time, the same priority described above for the flags also applies for the commands. For example, if

STARTFL is ON and a Stop command is given (through COMMAND), the Stop command is executed and all lower priority command flags are automatically turned OFF.

9.10.11 OFFDELAY

Description

When the input state changes from ON to OFF, an internal timer starts counting down the delay specified by DLYTIME. When it times out, the input is monitored again, and if it is still OFF, the output is set OFF, When the input state transitions too ON, the output is set to ON immediately and the timer is shut off.



Function

Used to delay the input by a specified delay time after an ON/OFF device transitions from the ON state to the OFF state.

Delay time in seconds is specified by the DELAYTIME parameter.

Inputs

IN = Boolean value

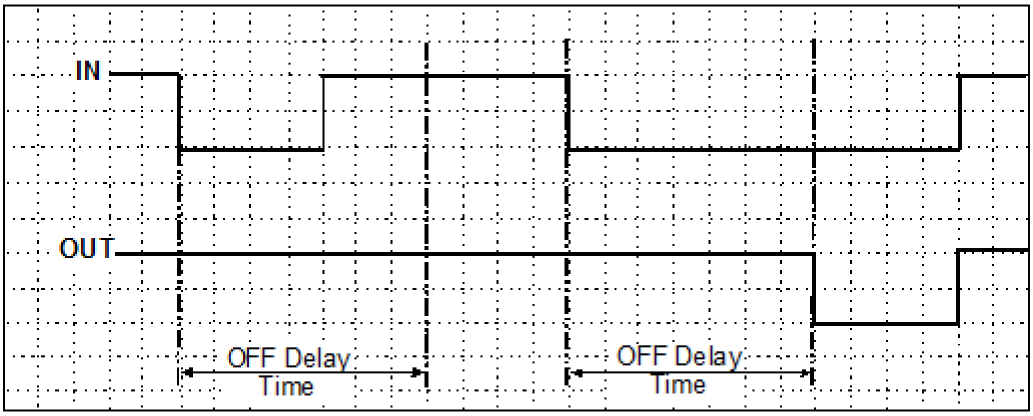
No delay is provided when the input goes from the OFF state back to the ON state.

Outputs

OUT = Boolean value

When the input transitions from the OFF state to the ON state, the output is set to ON immediately.

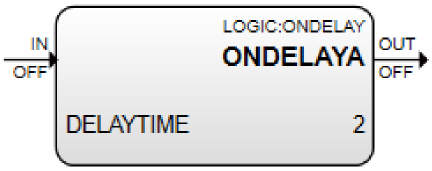
OFFDELAY timing diagram:



9.10.12 ONDELAY

Description

When the input state changes from OFF to ON, an internal timer starts counting down the delay specified by DLYTIME. When it times out, the input is monitored again, and if it is still ON, the output is set ON, When the input state transitions to OFF, the output is set to OFF immediately and the timer is shut off.



Function

Used to delay the input by a specified delay time after an ON/OFF device transitions from the OFF state to the ON state.

Delay time in seconds is specified by the DELAYTIME parameter.

Input

IN = Boolean value

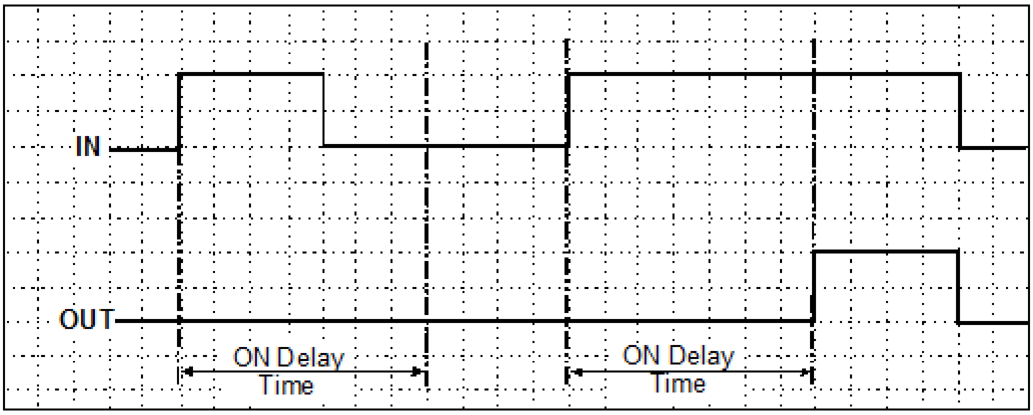
No delay is provided when the input goes from the ON state back to the OFF state.

Output

OUT = Boolean value

When the input transitions from the ON state to the OFF state, the output is set to OFF immediately.

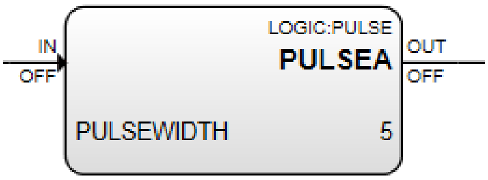
ONDELAY timing diagram:



9.10.13 PULSE

Description

Provides a fixed pulse output (OUT) each time the input (IN) transitions from OFF to ON. You specify the fixed output pulse width (PULSEWIDTH) in seconds through configuration.



Function

Used to define the fixed output (OUT) pulse width.

If the input (IN) pulse time is less than or equal to the fixed PULSEWIDTH time, output (OUT) pulse width equals the fixed PULSEWIDTH time.

If the IN pulse time is greater than the fixed PULSEWIDTH time, OUT pulse width is restricted to the fixed PULSEWIDTH time. Another output pulse cannot be generated until the preceding pulse has completed.

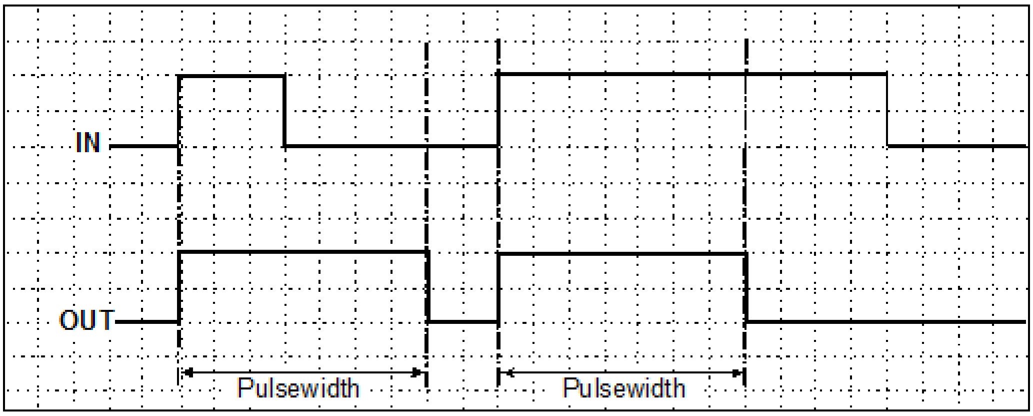
Inputs

IN = Boolean value

Output

OUT = Boolean value

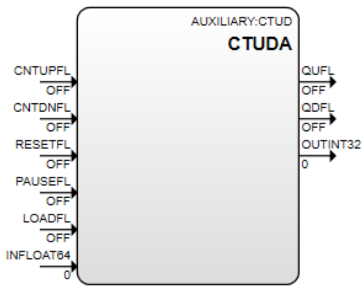
PULSE timing diagram:



9.10.14 Counter(CTUD)

Description

A new general purpose Up-Down Counter (CTUD) is introduced in the Auxiliary library to simplify event count strategies. The counter function block starts functioning based on the configured algorithm. The count inputs may be wired to other function blocks or stored by a program.



Function

The CTUD block is an up-down counter function block. The counter of the CTUD block can change its state (Up or Down) depending on the configuration of Count Up Flag (CNTUPFL) and Count Down Flag (CNTDNFL) parameter.

The counting also depends on a valid IN (ININT32/INFLOAT64) configuration if the input is fed through wired connection.

Up-down counting is evaluated as edge trigger quantity or level trigger quantity depending on the value configured for Count On Level (CNTLVLFL) Parameter.

The CTUD Block supports pause (PAUSEFL), load (LOADFL) and reset (RESETFL) operation for the counter.

If the Count Up and Count Down flags are set to “TRUE”, the same block execution results in a net internal counter change of zero.

Inputs

The CTUD block accepts a combination of Integer 32, Boolean, and Float 64 inputs.

• Either of ININT32 or INFLOAT64 can be used during block execution. Selection of which IN parameter to use is determined by Input Specifier (SELINT32FL) parameter.

– If SELINT32FL is set to “TRUE”, value of ININT32 parameter is used.

– If SELINT32FL is set to “FALSE”, value of INFLOAT64 parameter is used.

• All inputs are processed synchronously with the block execution.

Outputs

The current Counter output value is available in Float 64 (OUTFLOAT64) and Integer 32 (OUTINT32) formats.

• CARRYUPFL is set to TRUE for one block execution following a counter overflow.

• CARRYDNFL is set to TRUE for one block execution following a counter underflow.

• QUFL indicates count Up reached.

• QDFL indicates count Down reached

• The output values are fetched through a wired connection or read directly by a program.

• An output connection to an input that can be initialized does not create a back calculation or function block connections.

The flag outputs (QUFL, QDFL, CARRYUPFL, and CARRYDNFL) are transitory.

Downstream blocks, which sample these outputs, should sample at a rate at least twice the execution rate of the counter block in order to recognize all transitions.

9.10.15 MESSAGE

Description

The MESSAGE block provides up to 16 user configurable messages (MESSAGE[n]) that can be triggered by a client of the block. Where "n" is the number of the message.

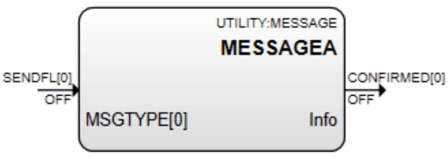
A client can be the output from a Step block in a Sequential Control Chart module (SCM).

You can also configure each message type (MSGTYPE[n]) to be either:

Information,

Confirmable,

Single Signature, or Double Signature.



Function

When a client triggers a given send flag (SENDFL[n]) input, the corresponding message (MESSAGE[n]) is sent to the Message and the Event Summary displays in the Station application. For information only type (INFO) messages, the client trigger sets the corresponding SENDFL[n] to True. Since the SENDFL[n] is a pulse trigger, it is automatically set to False during the next execution cycle. this means the MESSAGE block is ready to send the same message again in the next cycle.

For confirmation type (CONFIRM) messages, the client trigger pulses the corresponding SENDFL[n] to send the MESSAGE[n] to the Server.

The client of the MESSAGE block checks for the confirmed parameter (CONFIRMED[n]) to be set to True. The CONFIRMED[n] parameter indicates whether the MESSAGE block has received a confirmation.

For single signature type (SINGLESIGNATURE) messages, the client trigger pulses the corresponding SENDFL[n] to send the MESSAGE[n] to the Server. Once a user acknowledges the message twice to confirm it through the Message Summary display in Station, a Single Signature user interface appears for the user to record an electronic signature.

Configuration and Operation Considerations

Some general considerations for configuring and operating MESSAGE blocks are listed here for reference.

- Each message has a maximum length of 132 characters.
- A new message **cannot** be sent when the message is awaiting/blocked on a confirmation (CONFIRMED[n] parameter).
- You **cannot** configure the message type (MSGTYPE[n]) or minimum level secondary signature (MINLVLSECSIG[n] when the message is awaiting/blocked on a confirmation (CONFIRMED[n] parameter).
- You **cannot** configure a message (MESSAGE[n], meaning primary signature (MEANINGPRI[n] or meaning secondary signature (MEANINGSEC[n] through the Monitoring tab. You must configure messages through the Project tab and then load them to the Controller.
- When you acknowledge an Information message, it is removed from the Message Summary display. Confirmation type messages are confirmed by a second acknowledgement and then removed from the display.

Input/Output

The block has up to 16 inputs (SENDFL[0..15]) and 16 outputs (CONFIRMED[0..15]), depending on the message types configured.

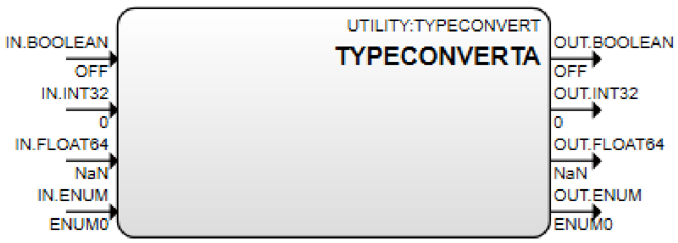
9.10.16 TYPE CONVERTOR

Description

The TYPECONVERT block provides the ability to convert one data type to another for connecting parameters of different data types. It supports data type conversions for all combinations among the following major data types:

- Boolean
- Integer (unsigned/signed 8/16/32-bit integers)
- Real (32-bit and 64-bit IEEE floating point numbers)
- Enumeration

It looks like this graphically:



Function

The TYPECONVERT block is used to connect one input parameter to one or many output parameters with different data types. For example, a Boolean input (IN.BOOLEAN) can be converted to a 32-bit integer (OUT.INT32), a 64-bit floating point number (OUT.FLOAT64), and an enumeration (OUT.ENUM) outputs. The general Control Builder configuration rule about only connecting parameters of the same data types for block inputs and outputs still applies. The TYPCONVERT block reads the input value and only provides the converted output when the block connected to its output runs.

Input/Output

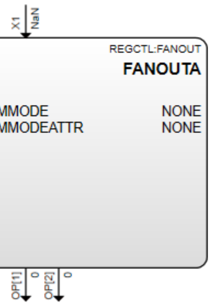
The block has up to nine inputs and nine outputs. The pins for the four most common inputs (IN.BOOLEAN, IN.INT32, IN.FLOAT64, IN.ENUM) and outputs (OUT.BOOLEAN, OUT.INT32, OUT.FLOAT64, OUT.ENUM) are exposed by default.

But, all block pin parameters are available to be exposed and connected to using Control Builder graphical connections.

9.10.17 FANOUT

Description

The FANOUT block has one input and up to eight initializable outputs. It may also have up to eight secondaries, since there is one secondary per initializable output. You may specify a separate gain, bias, and rate for each output. Each specified value can be fixed or external. A fixed value is stored manually or by a program, and an external value comes from another function block. This block calculates a separate floating bias for each output following an initialization or mode change. This provides a "bumpless" transition for each output. It looks like this graphically:



Function

The FANOUT block provides a "bumpless" output for each of up to eight outputs following initialization or mode changes.

Inputs

The FANOUT block requires one input - X1:

X1 = initializable input which must come from another block (it cannot be set by an operator or a program).

You must specify an engineering unit range (XEUHI and XEULO) for X1.

The block applies no range check. It assumes that X1 is within the specified range.

XEUHI and XEULO define the full range of X1:

XEUHI represents the 100% of full scale value.

XEULO represents the 0% of full scale value.

Outputs

The FANOUT block may have up to 8 initializable outputs as follows:

OP[1..8] - calculated output, in percent.

OPEU[1..8] - calculated output, in engineering units.

Initializable inputs and outputs

"Initializable input" and "initializable output" are variable attributes, similar to data type or access level. A parameter with the "initializable" attribute has an associated BACKCALC parameter and, when a connection between an initializable input and initializable output is created, you can also create a BACKCALC connection between them. Control Builder automatically builds the required BACKCALC connections, so you don't have to create them manually. These "implicit" build connections are "hidden" from view and the related parameter pins are not exposed on the control chart.

For example, if you connect OP from a FANOUT block to an AUTOMAN block or an AOCHANNEL block, Control Builder automatically creates the BACKCALCOUT to BACKCALCIN connection.

- For a given secondary, a connection to OP or OPEU may be created, but not to both. (The default OP connection is exposed, but the implicit/hidden connection function automatically makes a connection to a value/status parameter (OPX/OPEUX) when it is required.)
- A separate gain and bias may be specified for each output.
- The FANOUT block applies a separate floating bias to each output.
- Gain limits may be configured with negative values, thereby making it possible to reverse outputs by using negative gains.
- The FANOUT block provides the X1 input range (XEUHI/XEULO) to the primary through BACKCALC. The primary uses this for its output range (CVEUHI/CVEULO).

Output ranges

- CVEUHI[1..8] and CVEULO[1..8] define the full range of CV in engineering units for each given output.
  - The FANOUT block does separate ranging for each output by maintaining a separate CV range for each output which tracks the input range of the corresponding secondary.
  - The CV range for each output must be the same as the input range of each secondary. The FANOUT block brings the input range from each secondary (through BACKCALC) and stores it as the corresponding CV range. As a result, each output may have a different CV range. For example, a FANOUT block has its outputs OP[1] and OP[2] connected to blocks PID1 and PID2, respectively. It brings the input ranges of PID1 and PID2 and sets its CV ranges of OPX[1] and OPX[2] to these input ranges, respectively.
  - The FANOUT block brings the secondary's input range regardless of SECINITOPT (that is, regardless of whether the secondary's initialization and override data will be used).

- OPHILM and OPLOLM define the normal high and low limits for OP as a percent of the CV range. These are user-specified values. The same limits apply to all outputs.
  - OP is clamped to these limits if the algorithm's calculated result (CV) exceeds them or another function block or the user program attempts to store an OP value that exceeds them. However, the operator may store an OP value that is outside these limits.

OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range. These are user-specified values. The same limits apply to all outputs. The operator is prevented from storing an OP that exceeds these limits.

Output bias

The output bias (OPBIAS) is added to the algorithm's Calculated Value (CV) and the result is stored in CV. CV is later checked against OP limits and, if no limits are exceeded, copied to the output. Since the FANOUT block can have up to eight outputs, a separate output bias is determined for each output. This means that the parameters referenced in this discussion are actually indexed to the given output. For example, OPBIAS[1] and CV[1] are indexed to OP[1], and so on for the other seven outputs numbered 2 to 8.

The OPBIAS is the sum of the user-specified fixed bias (OPBIAS.FIX) and a calculated floating bias (OPBIAS.FLOAT). The purpose of the floating bias is to provide a bumpless transfer when the function block initializes or changes mode as long as the FANOUT block is the first initializable block.

- OPBIAS is recomputed under the following conditions to avoid a bump in the output. (Note that the function block only applies OPBIAS.FLOAT to the output for the latter two conditions, when it is the first initializable block.)
    - When the function block starts up (that is, goes Active).
    - When the function block initializes (for example, the secondary requests initialization).
    - When the mode changes to Cascade (as applicable for the given block).

- The following occurs when you set the OPBIAS value.
    - The total bias (OPBIAS) and fixed bias (OPBIAS.FIX) are both set to the entered value.
    - The floating bias (OPBIAS.FLOAT) is set to zero.
    - There are no limit checks applied when you set an OPBIAS or OPBIAS.FIX value. However, after the total bias is added to CV, the result is compared against the output limits and clamped, if necessary.
    - You configure the value for the fixed bias (OPBIAS.FIX) and it is never overwritten by the floating bias (OPBIAS.FLOAT). This means the total bias will eventually equal the OPBIAS.FIX , if you configure OPBIAS.RATE to ramp down OPBIAS.FLOAT.
    - You may store to OPBIAS.FIX only if the function block is inactive or the MODE is Manual; or if it is a PID or PIDFF function block with the CTLEQN set to E. When you store to OPBIAS.FIX, the following occurs:
    - The total bias (OPBIAS) and fixed bias (OPBIAS.FIX) are both set to the new value.
    - The floating bias (OPBIAS.FLOAT) is set to zero.

The OPBIAS.FLOAT is calculated as follows.

OPBIAS.FLOAT	=	CV_{INIT} - ($CV_{UNBIASED}$ + OPBIAS.FIX)
Where:		
CV_{INIT}	=	initialization value received from the secondary
$CV_{UNBIASED}$	=	unbiased calculated value (based on input from the primary)
OPBIAS.FIX	=	fixed bias (user-specified)

- If the primary accepts this block's initialization request, then CV + OPBIAS.FIX should be the same as CVINIT and OPBIAS.FLOAT will be zero. In most cases, OPBIAS.FLOAT will be zero. However, if the primary does not accept this block's initialization request because the primary is a FANOUT block or it was configured to ignore initialization, then OPBIAS.FLOAT value will not be zero.
  If OPBIAS.FLOAT is not zero, you can configure it to ramp down to zero through the OPBIAS.RATE parameter.
- You configure the OPBIAS.RATE to apply a ramprate to the OPBIAS.FLOAT. It is only used when the OPBIAS.FLOAT is not zero. The OPBIAS.RATE is expressed in Engineering Units per minute and may have the following values.
  - Zero:
    If OPBIAS.RATE is zero, a OPBIAS.FLOAT is calculated and bumpless transfer is guaranteed. However, if OPBIAS.FLOAT is not zero, it will never ramp down.
  - Non-zero:
    If OPBIAS.RATE is not zero, an OPBIAS.FLOAT is calculated and bumpless transfer is guaranteed. If the OPBIAS.FLOAT is not zero, it is ramped to zero at the rate you configured for the OPBIAS.RATE parameter.
  - The function block ramps the OPBIAS.FLOAT to zero by applying the following calculation each time it executes.

OPBIAS.FLOAT	=	OPBIAS.FLOAT - (OPBIAS.RATE / cycles_per_Min)
Where:		
cycles_per_min	=	number of times the function block executes per minute (calculated)

  - NaN: When the OPBIAS.RATE is Not a Number (NaN), no OPBIAS.FLOAT is calculated. This means a bump in the output will occur, if the primary does not accept this block's initialization value.

Mode handling

The FANOUT block supports both the Cascade and Manual modes:

- If mode is CAScade, then: X1 must be pulled from another block.
- If mode is MANual, then: OP may be stored by the operator or a user-program

(X1 is ignored).

Timeout monitoring

If mode is CAScade, the FANOUT block performs timeout monitoring on X1. If the X1 value is not updated within a predefined time (TMOUTTIME), the FANOUT block invokes timeout processing as follows:

- Sets the "input timeout" flag (TMOUTFL).
- Sets the input value to Bad (NaN - Not a Number).
- Requests the X1 primary to initialize (through X1BACKCALCOUT).

The FANOUT block does not support mode shedding on timeout.

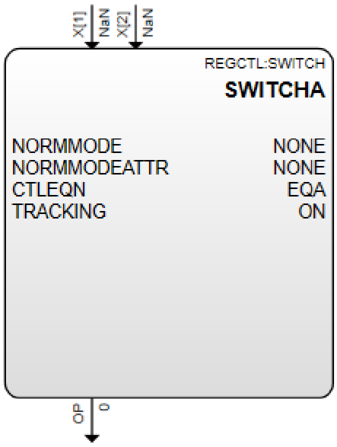
The maximum time between updates is specified by TMOUTTIME (in seconds)

- Enable timeout monitoring by setting TMOUTTIME to a non-zero value.
- Disable timeout monitoring by setting TMOUTTIME to zero.

9.10.18 SWITCH Block

Description

The SWITCH block accepts up to eight initializable inputs and operates as a single-pole, eight-position rotary switch. The switch position may be changed by the operator, a user program, or another function block. It looks like this graphically.



Function

This block lets you select one input from as many as eight, and outputs the selected value.

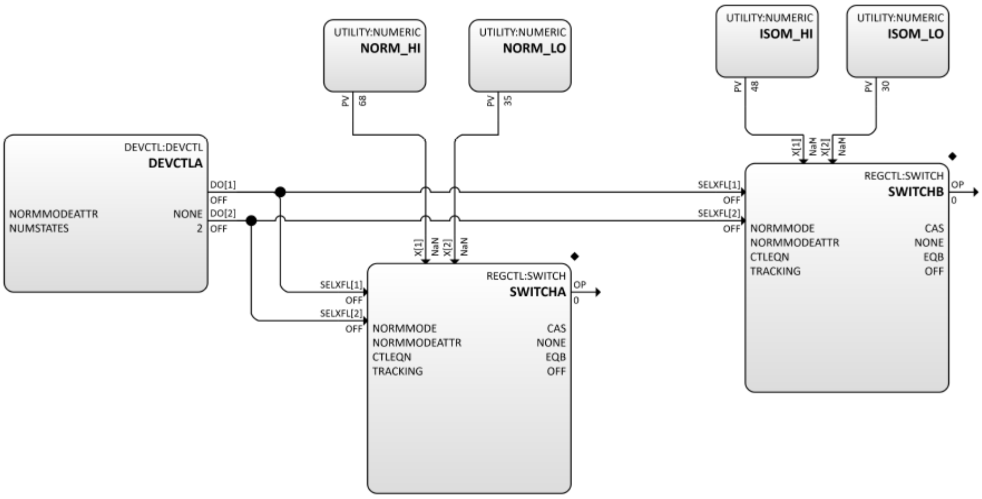
It provides these three methods for selecting an input:

Equation A. You store the number of the input to be selected to SELXINP.

Equation B. You set one of the selection flags (SELXFL[1..8]) to On. Each flag corresponds to an input. The block turns all of the other flags Off and updates SELXINP.

Equation C. You set or reset one of the selection flags (SELXFL[1..8]). The block does not change any of the other flags. Instead, it scans all flags in ascending order (1 to 8) and selects the first one that is On.

You can use this block to assign a different primary to a secondary. The example configuration shown in the following figure has five primary PID blocks connected to a SWITCH block. The active primary is selected by turning ON the corresponding SELXFL[1..5] input or storing the appropriate number to the SELXINP input, depending on the SWITCH block equation selected. The SELXINP parameter requires an integer data type and is usually set by an operator. The default SELXINP value is 1 and you cannot change it until the Control Module containing the SWITCH and primary blocks is activated at least once in Monitoring mode. Note that the configuration shown in the following figure is incomplete and is intended to only give you an idea of the general construction of a typical SWITCH block configuration.



Inputs

The SWITCH block accepts up to eight inputs - X[1] through X[8].

X[1] through X[8] are initializable inputs.

The inputs must be pulled from other function blocks; you cannot store to them.

This block may have two to eight primaries, depending on the number of inputs that are configured. (There is one primary per initializable input.)

Input ranges and limits

You must specify an engineering unit range for the X inputs, by entering values for XEUHI and XEULO.

XEUHI and XEULO define the full range of the inputs. XEUHI is the value that represents 100% of full scale, and XEULO is the value that represents 0%. XEUHI and XEULO apply to all of the X inputs.

This block assumes all of the X inputs are within XEUHI and XEULO; it applies no range checks.

Input descriptors

This block lets you define a 15-character descriptor (name) for each X-input. The descriptors reside in the XDESC parameter, and when an input is selected, the corresponding descriptor is copied to SELXDESC.

Initializable Outputs

"Initializable output" and "initializable input" are variable attributes, similar to data type or access level. A variable with the "initializable" attribute has an associated BACKCALC variable, and when a connection is created between an initializable input and initializable output, you can also create a BACKCALC connection. Control Builder automatically builds the required BACKCALC connections, so you don't have to create them manually.

These "implicit" build connections are "hidden" from view and the related parameter pins are not exposed on the control chart.

For example, if you connect OP from a SWITCH block to SP on a PID block, Control Builder automatically creates the BACKCALCOUT to BACKCALCIN connection.

OP = Calculated output, in percent.

OPEU = Calculated output, in engineering units.

You may create a connection to OP or OPEU but not both. Therefore, this block may have only one secondary. If you do not create a connection to OP or OPEU, then the block does not have a secondary. Alternately, if you connect OP or OPEU to a noninitializable input, then this block does not have a secondary. (Note that the default OP connection pin is exposed on the blocks and the implicit/hidden connection function automatically makes the appropriate value/status parameter (OPX/OPEUX) connection when required.

For example, if you connect the output from a SWITCH block (SWITCH.OP) to the set point of a PID block (PIDA.SP), the implicit/hidden connection is made to SWITCH.OPX to provide value/status data.)

Mode handling

This block supports the Cascade and Manual modes:

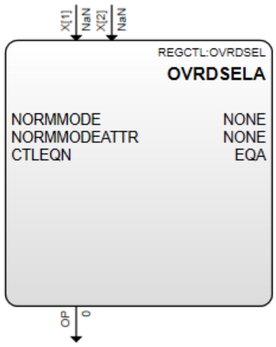
If MODE = Cascade, all inputs are pulled from other function blocks.

If MODE = Manual, OP may stored by the operator or user program; inputs are ignored.

9.10.19 OVRDSEL (Override Selector)

Description

The OVRDSEL block accepts up to four inputs (primaries) and selects the one with the highest or lowest value. It looks like this graphically:



Function

This block always forces the unselected inputs to track the selected input by enabling the override feedback option. You select the override option by setting the parameter OROPT to ON or by selecting the Enable Override Option check box on the block's parameter configuration form.

Inputs

The OVRDSEL block accepts one to four inputs - X[1] through X[4]. It requires at least two inputs, but they can be any of the four.

X[1] through X[4] are initializable inputs.

The inputs must be pulled from other function blocks; you cannot store to them.

This block may have two to four primaries, depending on the number of inputs that are configured. (There is one primary per initializable input.)

Input ranges

XEUHI and XEULO define the full range of inputs.

XEUHI represents the 100% of full scale value.

XEULO represents the 0% of full scale value.

This block assumes that all X-inputs are within XEUHI and XEULO. It applies no range checks.

Outputs

The OVRDSEL block has the following initializable outputs:

OP = Calculated output, in percent.

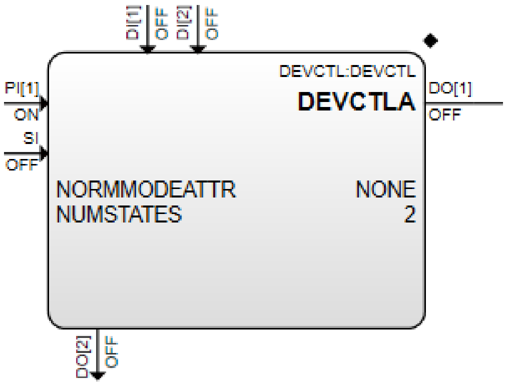
OPEU = Calculated output, in engineering units.

9.10.20 DEVCTLA (Device Control Block)

Description

The DEVCTL (Device Control) block is a multi-input, multi-output function that provides an interface to discrete devices, such as motors, solenoid valves, and motor operated valves.

This block provides built-in structures for handling interlocks and supports display of the interlock conditions in group, detail and graphic displays. It looks like this graphically.



Function

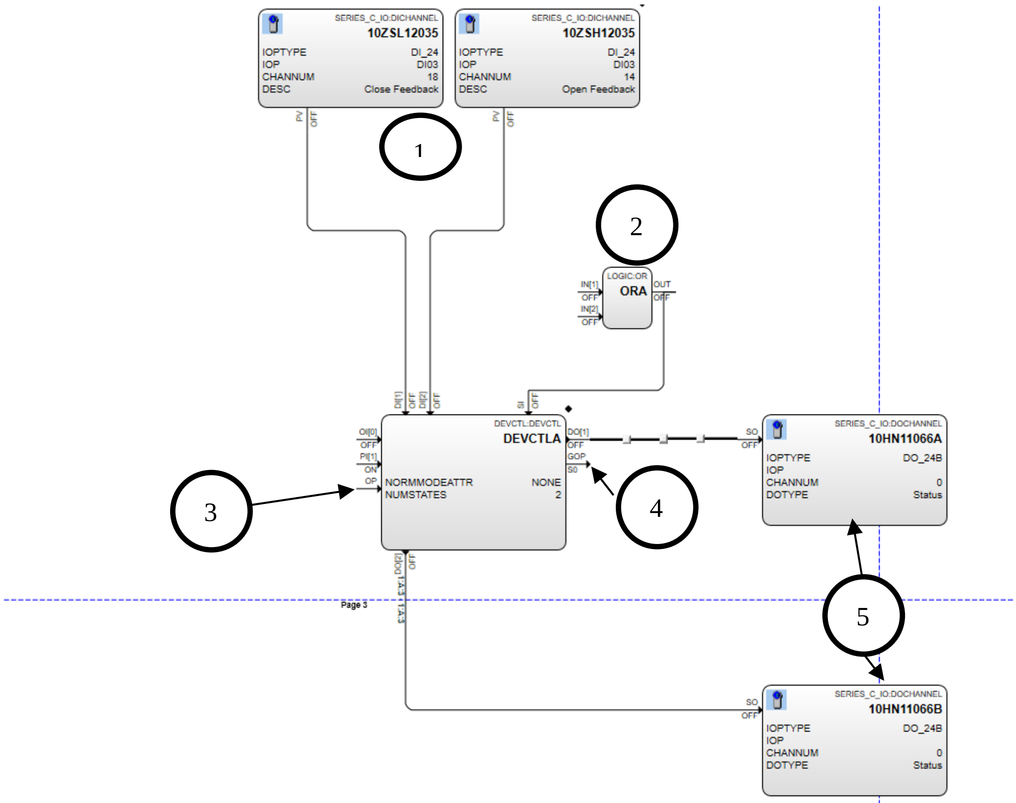
The DEVCTL block allows manipulation of sets of digital outputs and interprets corresponding feedback of digital inputs. Operation consists of transmitting the commands represented by the state parameter OP (the Commanded Output State), monitoring PV (the Current Active State), and producing alarms based on various configurations such as whether or not the PV has achieved the state commanded in OP.

In summary, the DEVCTL block provides these major features.

- Up to 4 inputs, 3 states, and 3 outputs.
- PV Source Selection, PV has 3 basic states plus in-between and faulty.
- Latched and pulsed outputs.
- Momentary states.
- Initialization, Local Manual and Redtagging.
- BADPV, Command Disagree, Uncommanded Change and Command Fail alarms.
- PV Change of state event.
- Permissive and Override Interlocks for each state.
- Interlock trip alarms.
- Seal In option.
- Maintenance statistics.
- The Safety Interlock enforces the defined safe state.
- Safe State explicitly configured. Cannot be momentary.
- Generic State parameters defined as consistent data types.
- Initialization has OPFINAL based configuration.
- Boolean Command option
- Batch level 1 driver option.
- OFF Normal Alarm associated with requested OP.

Configuration examples

Status Output - The following figure and its companion callout description table show a sample configuration that uses a DEVCTL block to command two status outputs. The view in the following figure depicts a loaded configuration in Monitoring mode.



The following table includes descriptions of the callouts in the figure above.

Callout	Description
1	Use the PVFL parameter connection to carry data from the DICHANNEL block to the DEVCTL block. In device control, the inputs provide the feedback that the commanded action has or has not taken place.
2	You can use appropriate interlock logic to activate the safety interlock function.
3	You can command the device through the output (OP), which shows the state names you configured for the block through Control Builder.
4	You can have the device commanded by another block or Sequential Control Module through the generic output (GOP), which shows the state as S0 to S2. The GOPSCADA parameter provides a link to Station detail displays and custom schematics to show the state as STATE_0 to STATE_2.
5	Use the BACKCALCIN/BACKCALOUT connection to carry secondary data from the DOC block to the DEVCTL. (Note that the individual BACKCALCIN/BACKCALCOUT connections for each DEVCTL output used are automatically built by Control Builder as implicit/hidden connections.) The secondary data contains this information for DEVCTL blocks. Initialization request flag - requests continuous initialization. If this flag is set and this block is configured to accept secondary initialization, this block goes to the initialized state and stays there until the flag is reset. Initialization value - provides continuous and one-shot initialization.

States

A "state" represents the present condition of a device. For example, Run and Stop could represent the "states" of a two-state motor, with Stop being the safe or failsafe state. A three-state motor could have the states of Run, Stop, and Reverse. Open and Close could represent the states of a valve. You can configure your given device states through the State Assignments tab of the DEVCTL block configuration form. This lets you associate states with Boolean combinations of process feedback inputs from the field. Each input combination is assign to a specific state. The PV parameter represents the present state of a device in the DEVCTL block.

You can also configure the number of output states as two or three through the State Assignments tab. These output states are mapped to specific combinations of digital outputs. These outputs command the field device to the associated state, such as Run or Stop. The OP parameter represents the commanded state or the device state commanded by an operator. The DEVCTL block transmits the OP, monitors the PV, and produces alarms based on the State Assignment configurations, which represent whether or not the process feedback has achieved the state commanded in OP.

Inputs

May have from 0 to 4 inputs (DI [1 .. 4]). Each input is a Boolean value, which may represent the state of any other block output or a field DICHANNEL (Digital Input Channel) block.

Outputs

May have from 0 to 3 outputs (DO [1 .. 3]). Each output may be Boolean or pulsed (On Pulse or Off Pulse). Each output is a Boolean value, which may be connected to any other block parameter or to a field DOCHANNEL (Digital Output Channel) block.

Mode and mode attribute

- Mode (MODE) is fixed at MANual. The Normal Mode (NORMMODE) parameter is also fixed at MANual.
- Mode Attribute (MODEATTR) - determines where state commands to the DEVCTL block may originate - that is, who may set the commanded output state (OP), as follows:
  - OPERATOR = only the operator may command the output state.
  - PROGRAM = only other function blocks (such as Logic blocks, SCM programs) may command the output state by setting OPREQ.
  - NORMAL = the setting specified by the Normal Mode Attribute (NORMMODEATTR) is assumed.

Two-State motor example

You can represent a simple two-state motor with one input (DI[1]). In this case, when the input is ON, the motor is in the Run mode. When the input is OFF, the motor is stopped.

DI[1] Input State	Configured State Name
0	Stop

1	Run
(bad) *	Fault

using two outputs to provide different outputs for two states. Of course, the NUMSTATES parameter is set to two.

Configured State Name	Related GENSTAT	Output States	
		DO[1]	DO[2]
Stop	S0	0	0
Run	S1	1	0

Two-State valve example

You can represent a valve as a device with two digital inputs. One input could represent the contact at the Open end of the valve travel, and the other could represent the contact at the Closed end of the valve travel.

Input States		Configured State Name
DI[1]	DI[2]	
0	0	Moving
1	0	Open
0	1	Closed
1	1	Fault

two outputs to open and close a valve. when Close is commanded, DO[1] only is set. When Open is commanded, DO[2] only is set.

⍰ Configured State Name	Related GENSTAT	Output States	
		DO[1]	DO[2]
Close	S0	1	0
Open	S1	0	1

Parameters

ASTEPID BADPVALM.FL	MOMSTATE NAME

BADPVALM.PR	NORMMODE
BADPVALM.SV	NORMMODEATTR
BYPASS	NULLPVFL
BYPPERM	NUMDINPTS
CLROPREQFL	NUMDOUTS
CMDDISALM.FL	NUMSIOVRD
CMDDISALM.PR	NUMSTATES
CMDDISALM.SV	NUMTRANS[0..2]
CMDDISALM.TM[0..2]	OFFNRMALM.FL
CMDFALALM.FL	OFFNRMALM.PR
CMDFALALM.PR	OFFNRMALM.SV
CMDFALALM.SV	OI[0..2]
CMDFALALM.TM[0..2]	OIALM.FL[0..2]
CONTROLREQ	OIALM.OPT[0..2]
DESC	OIALM.PR[0..2]
DI[1..4]	OIALM.SV[0..2]
DIPVMAP[0..15]	OP
DO[1..3]	OPCMD[0..2]
EUDESC	OPDOMAP[0..3][1..3]
GOP	OPFINAL
GOPFINAL	OPREQ
GOPREQ	OPTYPE
GOPSCADA	ORDER
GPV	ORDERINCM
GPVAUTO	PI[0..2]
HIALM.PR	PO[1..3]
HIALM.SV	POCONNECTED[1..3]
HIALM.TYPE	PULSEWIDTH[1..3]
HOLDOPT	PV
INALM	PVAUTO
INBETFL	PVFL[0..2]
INITCONNECTD[1..3]	PVSOURCE
INITMAN	PVSRCOPT
INITOPOPT	PVSTS
INITREQ[0..2]	REDTAG
LASTGOPREQ	RESETFL
LASTOPREQ	RESTARTOPT
LASTOPTYPE	SAFEOP
LASTREQFL	SAFEREDTAG
LASTSTEP	SEALOPT
LOCALMAN	SI
MAINTOPT	SIALM.FL
MAXTIME[0..2]	SIALM.OPT
MAXTRANS[0..2]	SIALM.PR
MODE	SIALM.SV
MODEATTR	STARTOPT

MODEATTRFL.NORM MODEATTRFL.OPER MODEATTRFL.PROG MODETRACK	STATETEXT[0..6] STATETIME[0..2] STOPOPT UNCMDALM.FL UNCMDALM.PR UNCMDALM.SV

Description of significant and commonly used pin or parameter:

Alarming: The following alarms are configurable to represent disagreements between the commanded state (OP) and the feedback state (PV). These alarms are disabled if there are no inputs or outputs.

- Command Disagree (CMDDISALM.FL): This alarm is generated when the commanded output state (OP) changes and the feedback state (PV) does not change to the same state within the specified feedback time. This alarm returns to normal when the PV state becomes the same as the OP state. This alarm does not apply for momentary commanded states.
- Command Fail (CMDFALALM.FL): This alarm checks to see if the PV state changed from its original state to any other state within a specified feedback time after the OP state is commanded. For slow responding devices, absence of this alarm indicates that the device responded to the command, even if it has not yet moved to its commanded position.
- Uncommanded Change (UNCMDALM.FL): This alarm is configured in conjunction with the Command Disagree alarm function. This alarm is generated, if an OP state has not been commanded and the PV state changes for any reason except BADPV.
- Bad PV (BADPVALM.FL): This alarm is generated whenever PV is detected in the Null state. The Null state can result from a BadPV condition for an input provided by a source block, or because input combinations represent a Null state as defined by the DIPVMAP[0..15] parameter

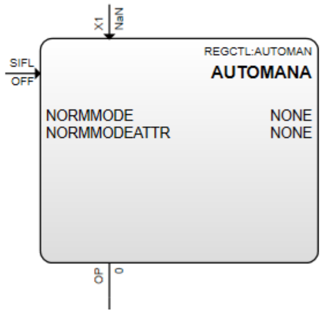
SAFEOP: Safe output Lets you select the state that defines the DEVCTL block in a safe state. The default is S0 (State 0). State 2 (S2) selection is only applicable if number of states (NUMSTATES) is three.

SEALOPT: Seal-In Option Lets you specify whether the Seal-In Option is to be enabled or disabled. See the Seal-In Option section for this block for information about this option. The default is an unchecked box or disabled. To enable the Seal-In Option, the Momentary state must be None. When the Seal-In Option is enabled, the Momentary State selection becomes void.

9.10.21 AUTOMAN (Auto Manual) Block

Description

The AUTOMAN (Auto Manual) block applies a user-specified gain and bias to the output. The user-specified values can be fixed or external. A fixed value is stored manually or by a program, and an external value comes from another function block. It looks like this graphically:



With R410, you can configure the on-delay time, off-delay time, deadband value, and deadband unit for the individual alarms. For example, you can use the following parameters to configure the on-delay time, off-delay time, deadband values, and deadband units for the OPHIALM parameter.

- OPHIALM.TM
- OPHIALM.TMO
- OPHIALM.DB
- OPHIALM.DBU

Each AUTOMAN block supports the following user configurable attributes. The following table lists the given name of the "Tab" in the parameter configuration form and then briefly describes the attributes associated with that Tab. This data is only provided as a quick document reference, since this same information is included in the on-line context sensitive Help.
The block calculates the output value (CV) using the following equation:

CV = KX1 + OPBIAS.FIX + OPBIAS.FLOAT
Where:

K	=	gain for CV (user specified)
X1	=	input value
OPBIAS.FIX	=	fixed output bias (user specified)
OPBIAS.FLOAT	=	floating output bias (calculated)

- K and OPBIAS.FIX may either be fixed (that is, stored manually or by the program) or external (that is, brought from another block).

- After an initialization, the block calculates OPBIAS.FLOAT as follows:

  $OPBIAS.FLOAT = CV_{INIT} - [KX1 + OPBIAS.FIX]$

  where: CV_{INIT} = initialization value from the secondary

Function

The AUTOMAN block is typically used:

- in a cascade control strategy where one of the upstream blocks may not accept an initialization request from its secondary.
- between a FANOUT block and a final control element to provide a "bumpless" output on return to cascade

Inputs

The AUTOMAN block requires one input - X1:

- X1 = initializable input which, if used, must be pulled from another block (it cannot be stored to).
- An engineering unit range for X1 (XEUHI and XEULO) must be specified.
- XEUHI and XEULO define the full range of X1:
- XEUHI represents the 100% of full-scale value.
- XEULO represents the 0% of full-scale value.

(Note that this block applies no range checks and assumes that X1 is within the XEUHI and XEULO range.)

Output

The AUTOMAN block has the following initializable outputs:

- OP = calculated output, in percent.
- OPEU = calculated output, in engineering units.

Initializable inputs and outputs

"Initializable input" and "initializable output" are variable attributes, similar to data type or access level. A parameter with the "initializable" attribute has an associated BACKCALC parameter. When a connection between an initializable input and initializable output is created, you can also create a BACKCALC connection between them. Control Builder automatically builds the required BACKCALC connections, so you don't have to create them manually. These "implicit" build connections are "hidden" from view and the related parameter pins are not exposed on the control chart.

For example, if you connect OP from an AUTOMAN block to a PID block or an AOCHANNEL block, Control Builder automatically creates the BACKCALCOUT to BACKCALCIN connection.

Output ranges

CVEUHI and CVEULO define the full range of CV in engineering units.

If the AUTOMAN block has a secondary, it brings the secondary's input range through BACKCALC and sets its CV range to that. If it has no secondary, CVEUHI and CVEULO track the X-input range (XEUHI and XEULO).

- OPHILM and OPLOLM define the normal high and low limits for OP as a percent of the CV range. These are user-specified values.

OP is clamped to these limits if the algorithm's calculated result (CV) exceeds them or another block or user program attempts to store an OP value that exceeds them. However, the operator may store an OP value that is outside these limits.

- OPEXHILM and OPEXLOLM define the extended high and low limits for OP as a percent of the CV range. These are user-specified values. The operator is prevented from storing an OP that exceeds these limits.

Mode Handling

The AUTOMAN block supports both the Cascade and Manual modes:

- If Mode is CAScade: X1 must come from another block.
- If Mode is MANual: an operator or a user program (X1 is ignored) may store OP.

Timeout Monitoring

If mode is CAScade, the AUTOMAN block performs timeout monitoring on X1. If the X1 value is not updated within a predefined time (TMOUTTIME), the AUTOMAN block invokes timeout processing as follows:

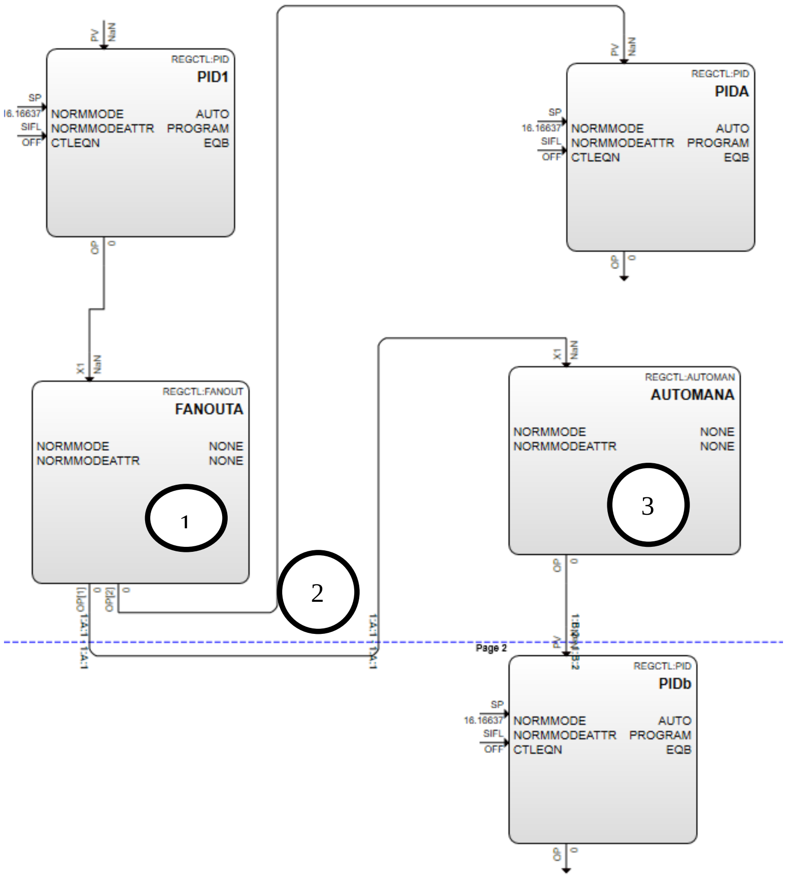
- Sets the "input timeout" flag (TMOUTFL).
- Sets the input value to Bad (NaN - Not a Number).
- Requests the X1 primary to initialize.

Note that the AUTOMAN block does not support mode shedding on timeout.

The maximum time between updates is specified by TMOUTTIME (in seconds)

Configuration_example:

The following figure and its companion callout description table show a sample configuration that uses an AUTOMAN block between a FANOUT block and a downstream PID block for quick reference.



Example of CB configuration using AUTOMAN block

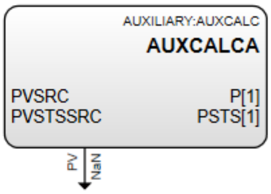
The following table includes descriptions of the callouts in the figure above.

Callout	Description
1	You can use the FANOUT block to distribute a single primary output to multiple secondaries. (Note that the individual BACKCALCIN/BACKCALCOUT connections for each FANOUT output used are automatically built by Control Builder as implicit/hidden connections.) Since the FANOUT block only initializes when all of its secondaries request it, insert an AUTOMAN block for individual downstream blocks (like PIDB in this example) to ensure bumpless transfer during mode changes.
2	You can specify a gain and bias for each of the FANOUT block outputs.
3	The primary purpose of this AUTOMAN block is to ensure a bumpless output upon return to Cascade mode. The AUTOMAN block is typically used between a FANOUT block and a final control element. Upon a return to Cascade, each secondary provides an initialization request to its primary. In most cases, the primary adjusts its output accordingly. However, if the primary is a FANOUT block, it may ignore the initialization request, since all of its secondaries may not be requesting it. In this case, the AUTOMAN block compensates for this by applying a floating bias to the output. This block applies a user-specified gain and bias to the output. The user-specified values can be fixed or external. A fixed value is stored manually or by a program, and an external value comes from another function block. The AUTOMAN block uses the following equation to calculate its output. • CV = K*X1 + OPBIAS.FIX + OPBIAS.FLOAT • where: o K = gain for CV o X1 = input value o OPBIAS.FIX = fixed output bias (user-specified) o OPBIAS. FLOAT = floating output bias (calculated)

9.10.22 AUXCALC (Auxiliary Calculation)

Description

The AUXCALC (Auxiliary Calculation) block lets you write up to eight expressions for computing a PV value. Each expression may perform arithmetic or logic operations, test conditions, etc. Status information is made available for both the inputs, as well as the expression results. Through configuration, you can assign the result of an expression, a status, or an input to PV and PVSTS parameters. It looks like this graphically.



Function

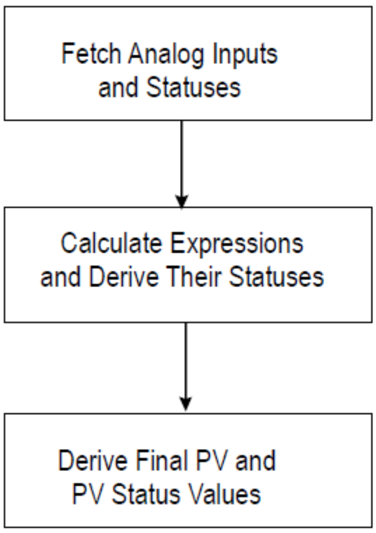
The AUXCALC block evaluates user-defined expressions and conditions to compute the desired output and status for the control strategy.

As shown in the following figure, the block may bring values from up to six inputs and determines their statuses in every execution cycle of the Control Module. It evaluates up to eight expressions and determines their statuses. It derives values for PV and PV status based on the configuration choices for the PVSRC and PVSTSSRC block parameters.

You can enter expression strings and configure PV and PV status selections at build time before the CM is loaded. The block performs syntax checking and conversion of the expression string during entry. If any errors are detected, they are displayed to inform you of the problem.

You must re-enter the string to correct the error. You can only enter an expression in the Project tab during block configuration. You cannot change an expression online in Monitoring tab.

The block checks and accepts other configuration parameters when the Control Module is active. If there are any invalid entries, it generates appropriate error messages to help identify the cause.



Input

This function block accepts as many as six inputs (P[1..6]):

All inputs are optional.

Must fetch all inputs from other function blocks.

The number of process input connections are equal to the number of inputs; the default is 1.

Output

This block produces the following outputs:

PV and its status (PVSTS)

As many as eight expression results (C[1] through C[8]) and their statuses.

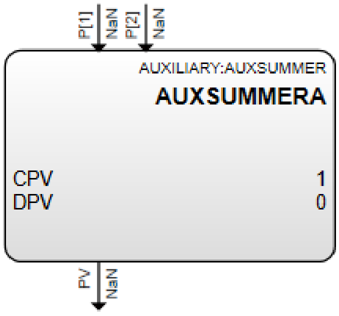
Expressions

You can write up to eight expressions, each expression can contain any valid combination of inputs, operators, and functions.

9.10.23 AUXSUMMER (Auxiliary Summer) Block

Description

The AUXSUMMER (Auxiliary Summer) block lets you configure up to ten separate inputs to calculate a process variable (PV) value that can be scaled and biased. Status information is made available for each input, as well as the PV value. Through configuration, you can define a scale factor, bias value, and description for each input. You can also choose to disable an input. All inputs are enabled by default. It looks like this graphically.



Function

The AUXSUMMER block uses the following equation to calculate the PV value based on up to ten configured inputs.

PV = CPV {((C [1] P[1]) + D [1]) + ... ((C [i] P[i])+ D [i])} + DPV

Where: CPV = Overall scale factor for PV

DPV = Overall bias for PV

C [i] = Scale factor for input 'i'

D [i] = Bias for input 'i'

P [i] = Input value 'i'

i = 1 to 10

The AUXSUMMER block brings values from other function blocks and determines their statuses in every execution cycle of the Control Module. It evaluates up to ten inputs and determines their statuses. It derives values for PV and PV status based on its calculation of the inputs and the configuration entries for the overall PV scale factor (CPV) and overall PV bias factor (DPV) parameters.

You can also choose to disable an input (PENABLE[1..10]) and define a substitute value (PSUB[1..10]) for the disabled input.

Input

This function block accepts as many as ten inputs (P[1…10]).

At least one input (P[i]) must be configured for the block to operate.

All inputs must be fetched from other function blocks.

The number of process input connections (NUMPINPT) that can be made to other blocks is equal to the number of inputs. The default is 1.

Output

This block produces the following outputs:

PV and its status, PVSTS

9.10.24 ENHAUXCALC (Enhanced Auxiliary Calculation)

Description

The ENHAUXCALC block provides the following enhancements over the AUXCALC block.

Expands existing arrayed input parameters PSTS and P from six to ten.

These arrayed parameters are added to correspond to each of the ten inputs.

Input Description

Scaling Factor

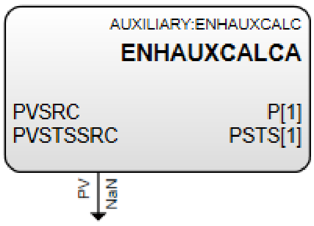
Enable/Disable Switch

PSUB Substitute Parameter

PP Scaled Input

Both the ENHAUXCALC and AUXCALC blocks are optimized so that expressions use memory based on the number of expressions configured, pcode size of each expression, the number of references in the expression and the offset needed for each expression.

It looks like this graphically.



Function

The ENHAUXCALC block evaluates user-defined expressions and conditions to compute the desired output and status for the control strategy.

As shown in the following figure, the block may bring values from up to 10 inputs and determines their statuses in every execution cycle of the Control Module. It evaluates up

to eight expressions and determines their statuses. It derives values for PV and PV status

based on the configuration choices for the PVSRC and PVSTSSRC block parameters.

An input switch parameter (PENABLE[1..10]) lets you enable or disable each corresponding input (P[1..10]). You can also configure a scaling factor (CP[1..10] for each corresponding input (P[1..10] to provide a corresponding scaled input (PP[1..10]).

The scaled input is computed as follows.

If PENABLE = 0 (Disable), then:

PP[i] = PSUB[i] CP[i]

Else: If PENABLE = 1 (Enable), then:

PP[i] = P[i] CP[i]

Where: i = 1 to 10

9.10.25 SIGNALSEL (Signal Selector)

Description

The Signal Selector function block accepts as many as six input signals, and may be configured to do one of the following on these inputs:

Select the input with the minimum value.

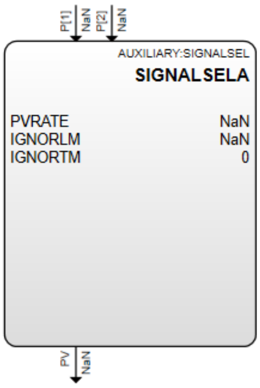
Select the input with the maximum value.

Select the median input.

Calculate the average of the inputs.

Select an input based on the Multiplex value; i.e., act as a multiplexer.

It looks like this graphically:



Function

This function block supports the following methods for selecting an input:

Method Processing

MIN Select the input with the minimum value. Ignored inputs are excluded.

MAX Select the input with the maximum value. Ignored inputs are excluded.

MED Select the median input. Ignored inputs are excluded.

AVG Calculate the average of the inputs. Ignored inputs are excluded.

MUX Select an input based on the Multiplex value; i.e., act as a multiplexer. Inputs are not ignored.

Input

This function block accepts between two to six selectable inputs, P[1] through P[6].

Minimum two inputs are required (P[1] and P[2]).

All inputs shall be fetched from other function blocks.

If less than two inputs are connected a warning "At least two inputs needs to be connected" shall be given during load and activation of the block shall be prevented.

Output

This auxiliary PV block shall have output PV and its status PVSTS.

It shall have a parameter SELIN denoting which input, if any has been selected as the output.

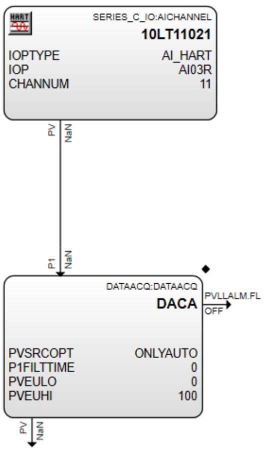
The block shall have the following output flags:

One flag denoting if any of the inputs is ignored or not (IGNORD).

Individual flags for each input indicating if it was ignored (IGNORDFL[1...6]).

9.11 Typical Loops

9.11.2 Typical AI Monitoring Loop



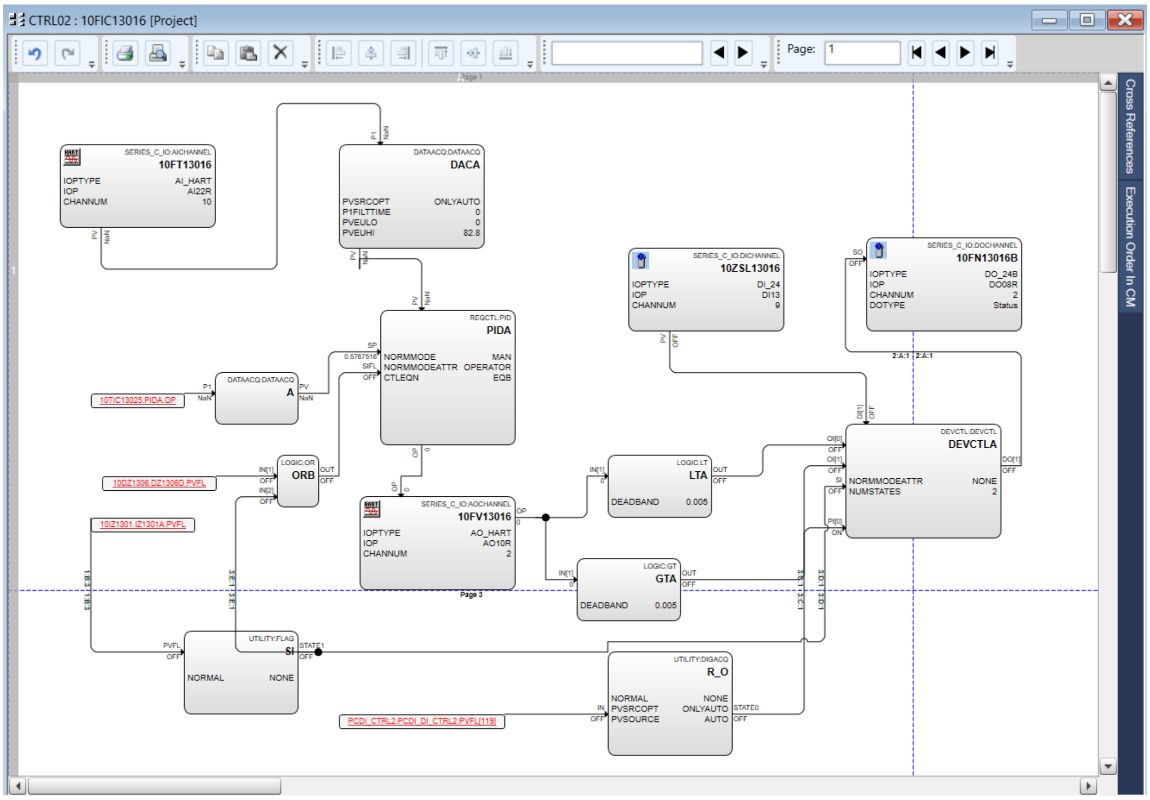
Example of AI Monitoring Loop

This typical Data Acquisition point is basically an indication point as shown in the following figure. This is our typical configuration for data acquisition point and it consists of Analog Input Channel and Data Acquisition function block. Both of the function blocks connections are “soft wire” between the parameter.

It will be built and assigned to the same controller where the physical Analog Input (process variable of the PID) point located. This data acquisition point will be assigned with a tag name and can be view at client station.

9.11.3 Typical PID CASCADE Loop

The PID block is used in a single PID Loop Control in a cascade strategy. Figure below shows the typical PID Loop- as configured in the Control Builder.



Example of PID CAS Loop

As can be seen above, typical PID Loop Control Module consists of the following function blocks:

- Analog Input Channel
- Data Acquisition
- Regulatory Control PID
- Analog Output Channel

To form desired control strategies, all the connection are "soft-wired" between the parameters of function block.

Functional Scenario

PV Source

PV is pulled from another function block. PV is typically pulled from a Data Acquisition (DATAACQ) function block, which performs PV limit checking and alarming.

The data acquisition function block is configured to fetch process input from an Analog Input channel and processes it input value (P1) without filtering into an output value. The PV source of Data Acquisition Function block is:

- PV AUTO - PV value is fetched from another function block

Mode Attributes and Modes

There are two mode attributes, which determines whether the operator (OPER) or a program (PROG) can change the point's mode. If the attribute is OPER, the operator enters the required value(s) at the station. If the attribute is PROG, the value is supplied by program.

Mode determines the role of the regulatory in a control strategy. An operator, internal logic, or a control program can change the mode.

- SP is stored by the operator from the Station
- With Experion System, the engineer can make it easy for the operator to select the normal mode for a control loop. For example, when the operator detect something abnormal and takes the control strategy out of it current operating mode and mode attribute. If after correcting the problem, the operator wants to return the control to the program but has forgotten the mode and attribute the point had, simply selecting the "Normal" option to returns it pre-configured mode and attribute.
- In this way, the operator never needs to remember the mode and attribute the point would be in if normal operation had been interrupted; the system remembers and responds when the Normal option is selected.

Shedding Manual PV Option

This selection is to indicate what the PID block should do if PV is manual, as follows:

- No Shed -- does nothing (that is, the PID block behaves the same as if PVSTS = Normal).
- Shed Hold -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and holds OP at the last good value. (This is the default selection.)
- ShedLow -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and sets OP to its extended low limit (OPEXLOLM).
- ShedHigh -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and sets OP to its extended high limit (OPEXHILM).

- ShedSafe -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and sets OP to the configured safe value (SAFEOP). If SAFEOP is NaN (Not a Number), OP is held at the last good value.

  Manual PV Option selected for above Loop:No Shed

Safety Interlock Option

The Safety Interlock Option determines the MODE and OP of the PID block is to assume upon a safety interlock alarm. This selection is to indicate what the PID block should do when Safety Interlock flag is on, as follows:

- NO_SHED -- does nothing (that is, the PID block behaves the same as if PVSTS = Normal).
- SHEDHOLD -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and holds OP at the last good value. (This is the default selection.)
- SHEDLOW -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and sets OP to its extended low limit (OPEXLOLM).
- SHEDHIGH -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and sets OP to its extended high limit (OPEXHILM).
- SHEDSAFE -- sets Mode to Manual and Mode Attribute to Operator, disables External Mode Switching, and sets OP to the configured safe value (SAFEOP).

If SAFEOP is NaN (Not a Number), OP is held at the last good value.

Safety Interlock Option for the above Loop: SHEDSAFE (SAFEOP= 0%)

When the value of HZA002 is 0 (Abnormal) then it will enable the SI as HZA002 is inverted by NOT1. So, in that case O/P will follow the SAFEOP, which is set at 0%.

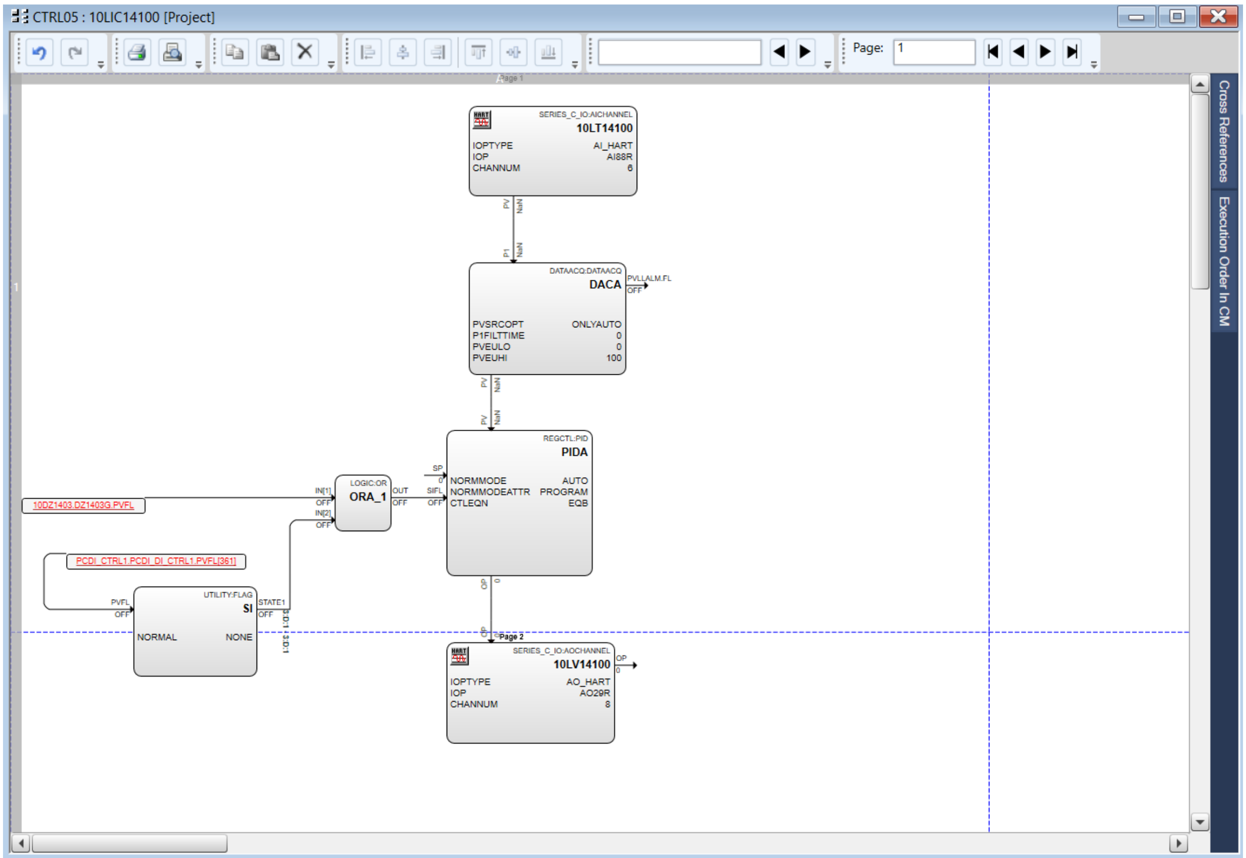
Bad Control Option

The Bad Control Option indicates what to do when CV = NaN (Not a Number) thereby causing the Bad Control flag (BADCTLFL) to be on. The Bad Control option determine how the MODE and OP of PID block is to assume if CV goes bad, as follows:

- NO_SHED -- MODE remains unchanged and OP is held at the last good value. (Default selection = NO_SHED)
- SHEDHOLD -- MODE is shed to Manual, MODEATTR goes to OPERATOR, external mode switching is disabled and OP is held at the current value.

- SHEDLOW -- MODE is shed to MANUAL, MODEATTR goes to OPERATOR, external mode switching is disabled and OP goes to OPEXLOLM.
- SHEDHIGH -- MODE is shed to MANUAL, MODEATTR goes to OPERATOR, external mode switching is disabled and OP goes to OPEXHILM
- SHEDSAFE -- MODE is shed to MANUAL, MODEATTR goes to OPERATOR, external mode switching is disabled and OP goes to SAFEOP. If SAFEOP = NaN, OP is held at the last good value.
- Bad Control Option for the above Loop: NO_SHED

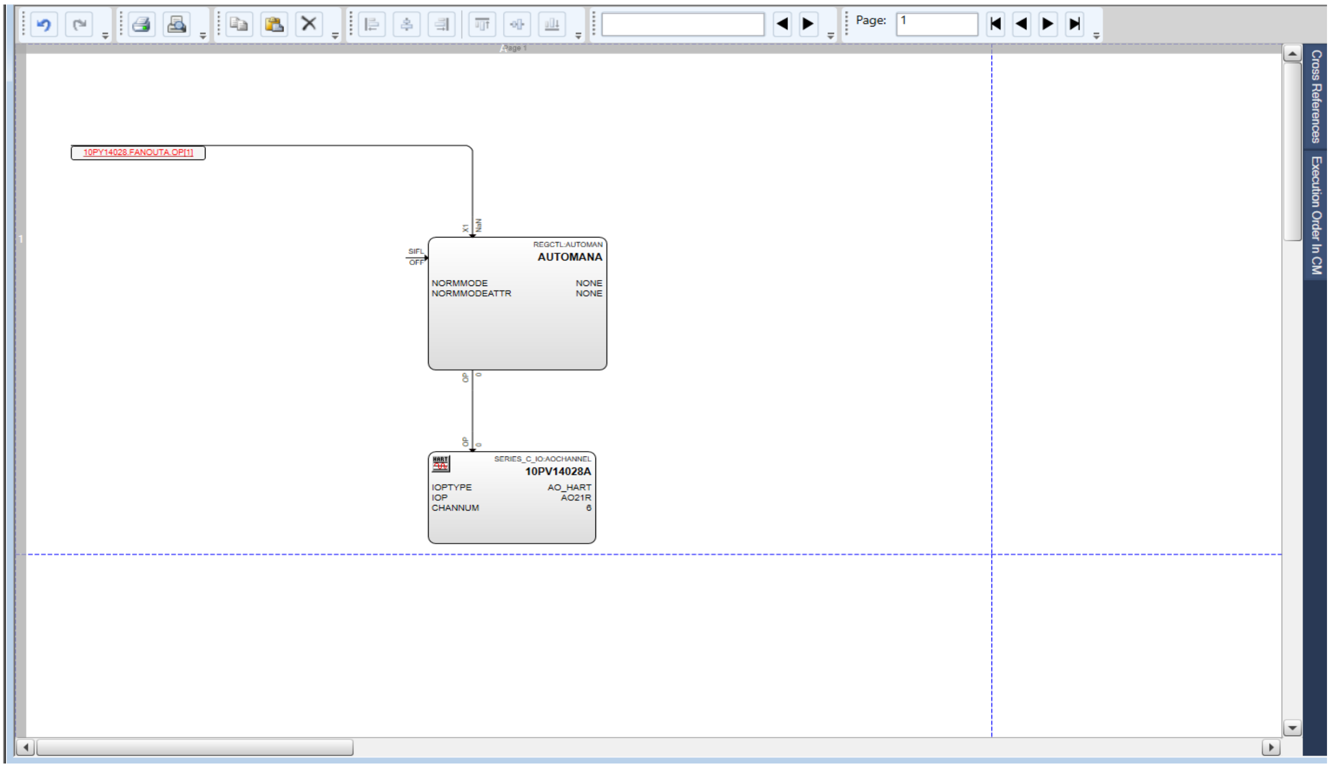
9.11.4 Typical Close Control loop Function with initialization by shut down

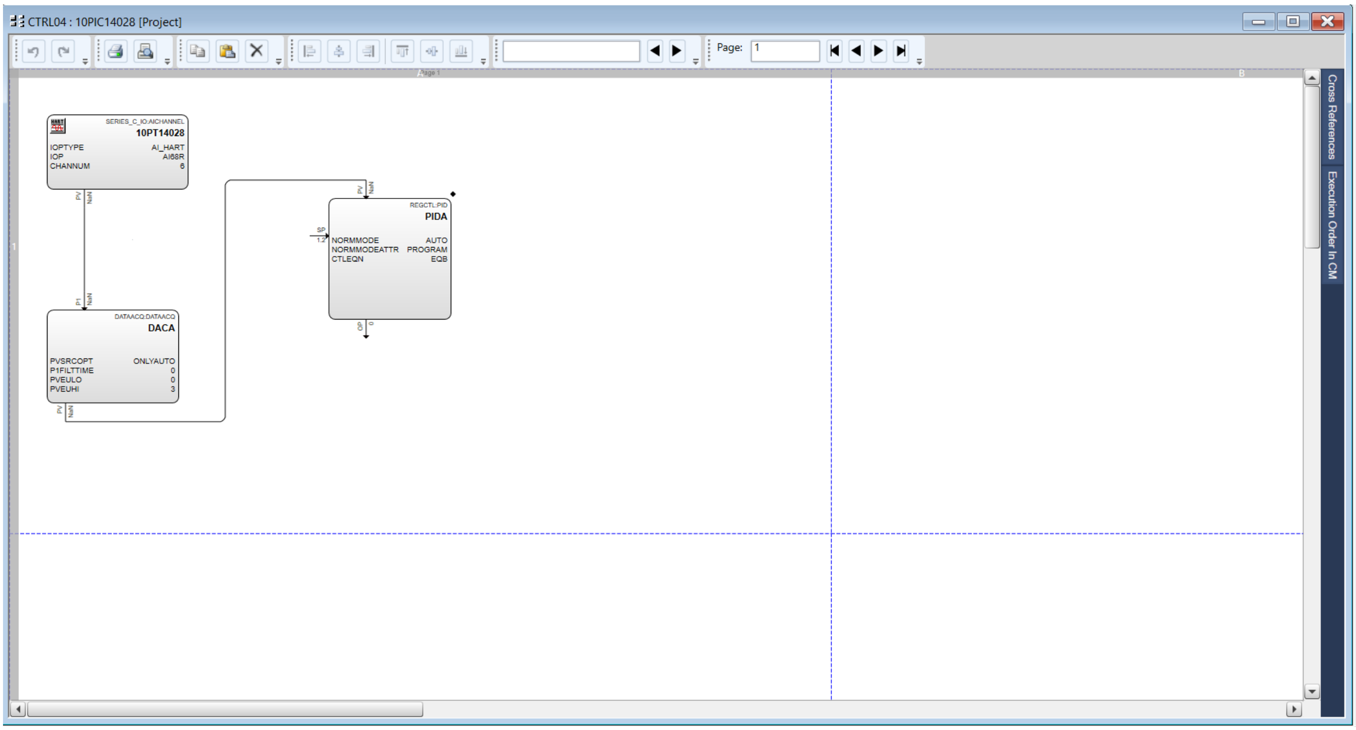


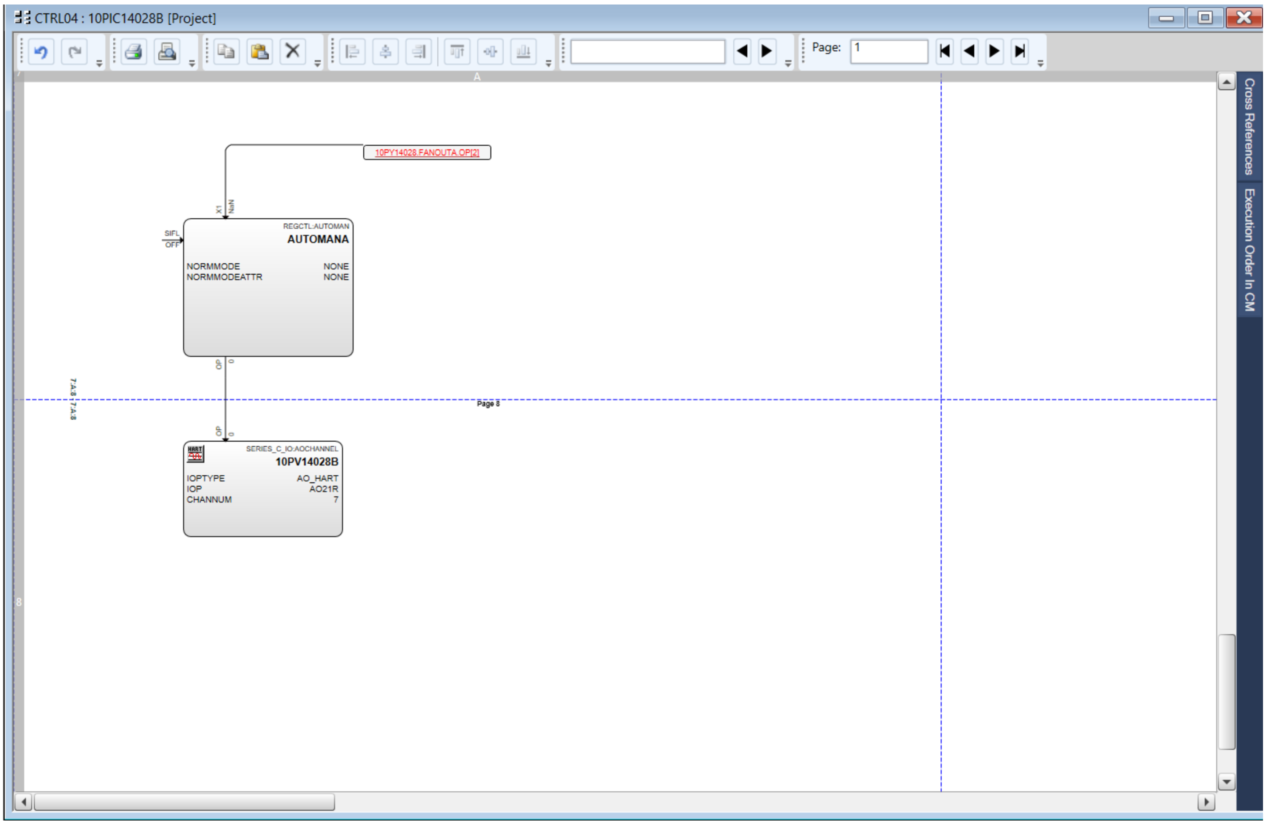
Example of Control loop With SD

Safety Interlock Option (SIOPT) selected ShedSafe and Safety Interlock Flag (SIFL) connected to ESD shut down. when SIFL set then result MODE goes shed to Manual, MODEATTR goes to Operator, external mode switching is disabled, and OP goes to SAFEOP.

9.11.5 Typical Split Rang Control loop







Example of Split Rang Control loop

For each split range using FANOUT and AUTOMAN blocks and separate each AO in different control module.

In control loop , out of PID block will enter in FANOUT block and the FANOUT have this ability that read one input and release 8 output separately and each output have separate gain and bias .

OUT 1 = (gain*input) + bias

OUT 2 = (gain*input) + bias

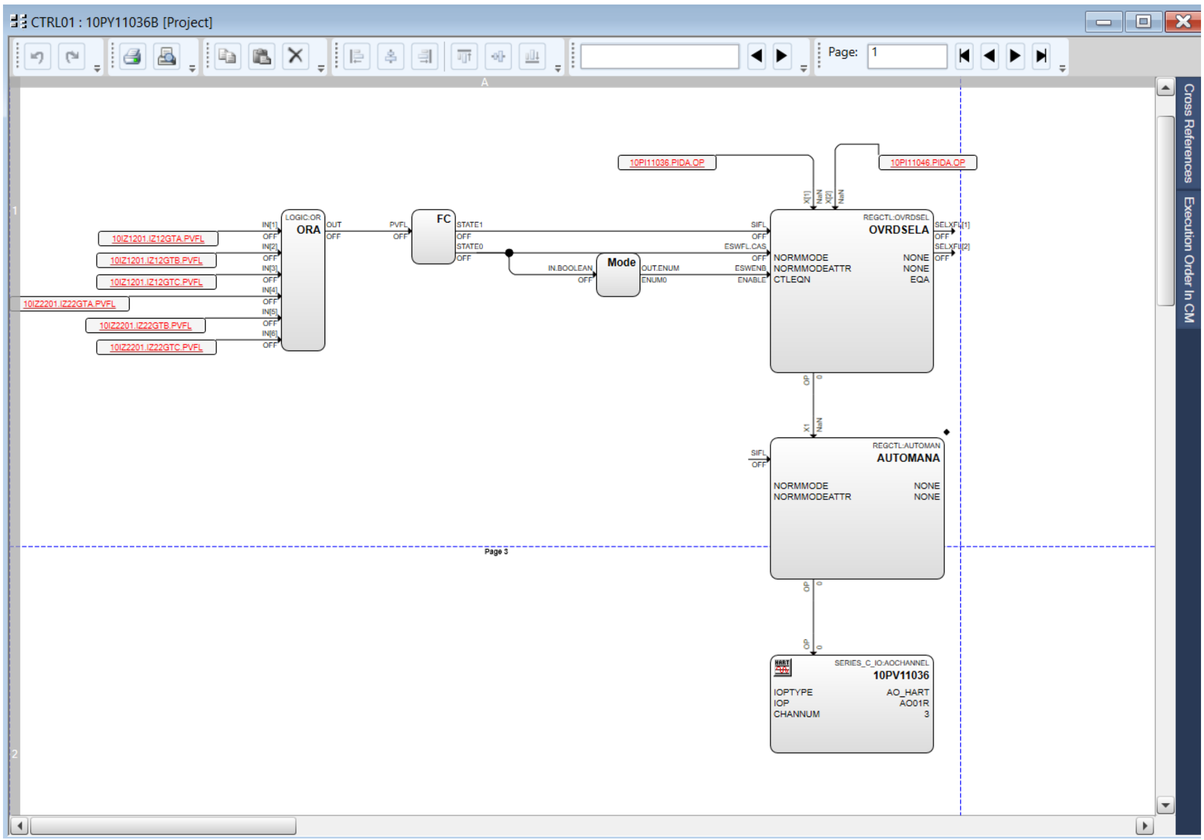
.

.

OUT 8 = (gain*input) + bias

Now each output can operate AO channel but for case of that operator be able to operate the output manually , its consider AUTOMAN block that its can change between auto and manual .

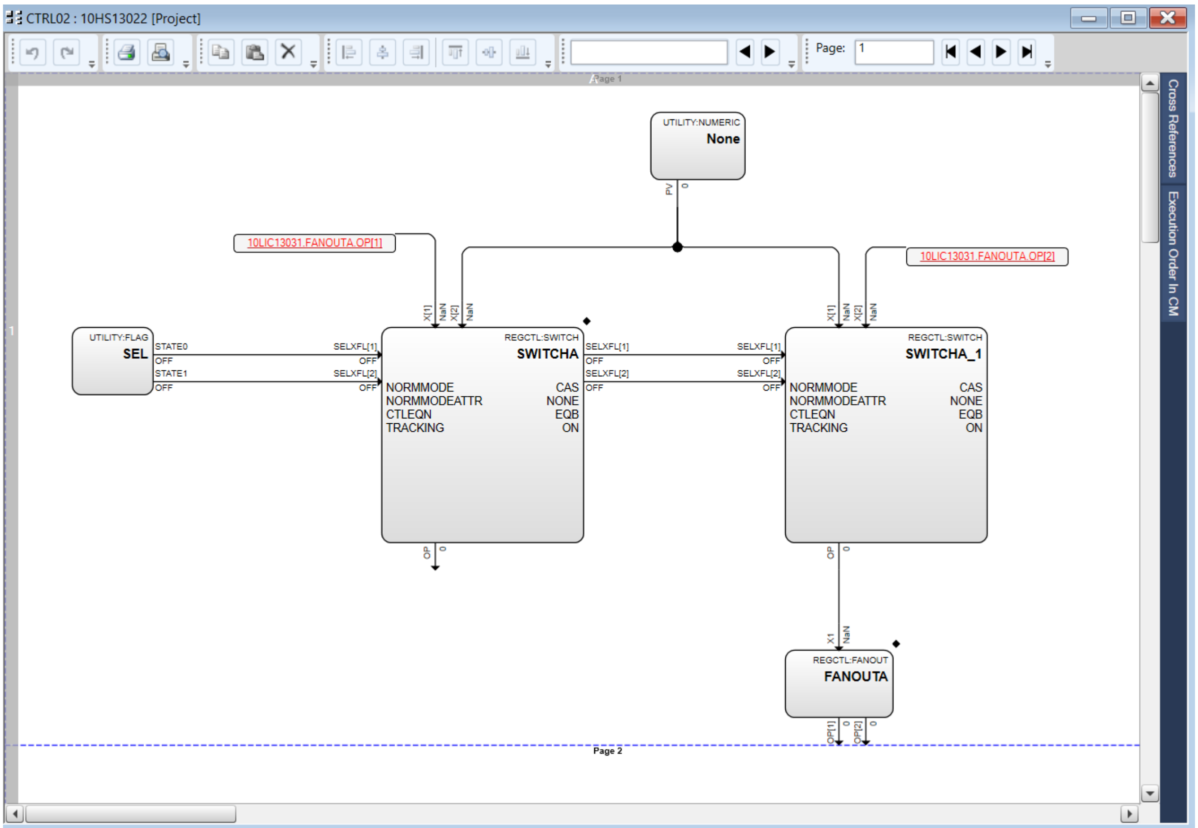
9.11.6 Typical OVRDSEL Loop



Typical OVRDSEL Module

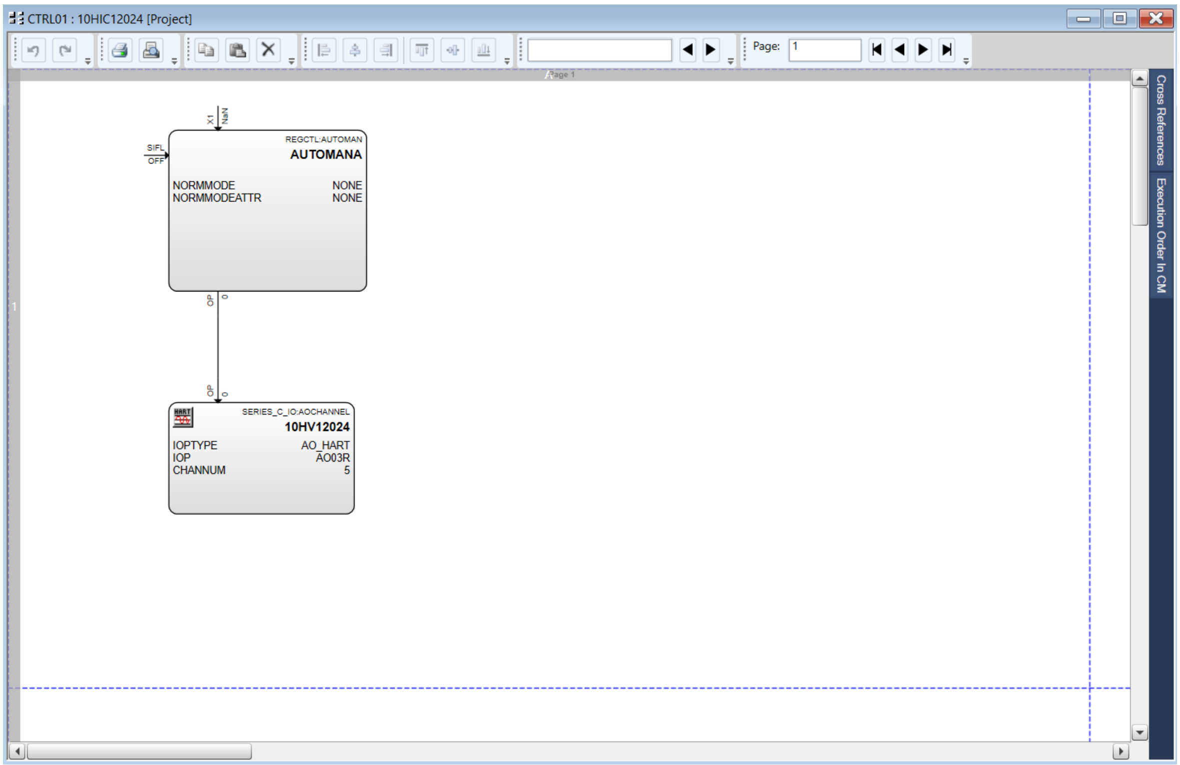
Different AI with different PID and control module will be connect to one OVRDSEL Module to operate one AO.All safety interlock will be seen in OVRDSEL Module.

9.11.7 Typical SWITCH Loop



Typical SWITCH Module

9.11.8 Typical Hand Controller loop



Typical Hand Controller loop

It’s a simple control module of valve that’s use AUTOMAN block.

9.11.9 Typical Device Control Loop

The Pump Logic is implemented by using Device Control. Figure A2.7 below presents a Device Control Module as configured in a Control Builder for Control Logic.

Basic Components

Device Control Module basically consists of Device Control Function Block (DEVCTL) and it's associated input(s)/output(s). Device Control Module can be configured up to 4 inputs with a combination of up to 3 outputs allows the process operator to focus on the device states rather than the software configuration.

Figure A2.6, a typical Device Control Module consists of the following function blocks.

- DEVICE CONTROL
- Device Control Function Block DEVCTLA
- Digital Input-1 (DI 1) as a Parameter Connection from the other Digital Input Channel Block for the Pump-260 Running Feedback
- Digital Output Channel (DO Channel) for the Start and Stop Output of the Pump
- General Logic Operation (AND / NOT Gate)
- Safety Interlock (SI)
- Override Interlock (OI 0 and OI 1)
- Permissive interlock
- Local Man

To form the control strategy, all connections are "soft-wired" between a connection of function block.

Override Interlock is a special parameter in Device Control FB to make an interlock function. When active, it forces a command output of Device Control FB into a respective state.

The process operator can change the state of the Pump Manually by Start and Stop Pulse from the faceplate of the Pump, provided that the pump is selected as a duty Pump and the Interlock is healthy.

Safety Interlock (SI) is provided as for a safety interlock purpose. SI is usually configured to prevent any damages to people or equipment. Therefore, SI has a higher priority than Override Interlock. When active, the command output will change to a pre-defined "Safe State" (Stop) and no one would be able to change the command output of the Device Control Module during SI is in active even though the Bypass parameter has been set.

Functional Scenario

Input (DI[1])

DI is a Boolean value, which represents the state of other block output or a field DICHANNEL (Digital Input Channel) block for the State of the Device- Running / Stop.

Output (DO[1..2])

Output (DO [1]) is on pulse for Start command of the Pump.

Output (DO [2]) is a Boolean value for Stop command of the Pump.

Each output is connected to DOCHANNEL (Digital Output Channel) block.

Control Action on Communication Lost or Control Failure

Each DO Channel can be separately configured the action when control failure or loss of communication between processor and I/O Module. These configurations are available in Channel Configuration Tab under the I/O Module configuration. Control Builder provides two options listed below:

- Shed to Hold - Check the box for each designated channel to specify how to control the output applied upon controller failure or loss of communication between controller and IOM.
- If enabled (checked), the output holds the last state processed.
- If disabled (unchecked), the output goes to the Safe State (see below).
- Safe State – Check the box (to toggle On) to indicate that the Safe State output is to be applied upon controller failure or loss of communication between controller and IOM if the Shed to Hold option is not checked for the designated channel.

State

A "state" represents the present condition of a device- Run and Stop based on the DI[1] state.

Mode and Mode Attribute

Mode (MODE) is fixed at MANUAL. The Normal Mode (NORMMODE) parameter is also fixed at MANUAL.

Mode Attribute (MODEATTR) - determines where state commands to the DEVCTL block may originate - that is, who may set the commanded output state (OP), as follows:

- OPERATOR = only the operator may command the output state.
- PROGRAM = only other function blocks (such as Logic blocks, SCM programs) may command the output state by setting OPREQ.
- NORMAL = the setting specified by the Normal Mode Attribute (NORMMODEATTR) is assumed.

Safe Output State

The Safe Output State (SAFEOP- State 0 / Stop) parameter defines the default state for certain actions of the DEVCTL block. SAFEOP can be assigned to any of the settable states of the block (that is, those states to which parameter OP may be assigned). The default for SAFEOP is State 0.

SAFEOP may not be assigned to a state, which is already configured as momentary.

Safety Override Interlock (SI)

The Safety Override Interlock (SI) forces the commanded output state (OP) to the Safe Output State (SAFEOP) when active. No one may command OP to a different state while SI is active. It may be connected to the other blocks or by an operator if MODATTR parameter is set to Operator.

- SI is defaulted to OFF, it must be set to ON to force OP to go to SAFEOP.
- When SI turns OFF, OP = SAFEOP is maintained.

Override Interlocks (OI[0..2])

OI[0..2] are Override Interlocks which, when active, force the commanded output (OP) to a respective state regardless of the condition of the Permissive Interlocks. OP cannot be commanded to a different state when an Override interlock is active. Override Interlocks may be connected to other block outputs or may be directly set by an operator if MODEATTR is set to OPERATOR.

OI has lower priority than that of SI. Therefore if SI and OI are occurred at the same time, SI will take action.

Permissive interlocks

Permissive Interlocks which are inputs that may be connected to an external function block to determine whether the operator and/or user program are allowed to change the commanded output (OP) of the DEVCTL block to a specific state. Permissive Interlocks themselves never cause OP to change.

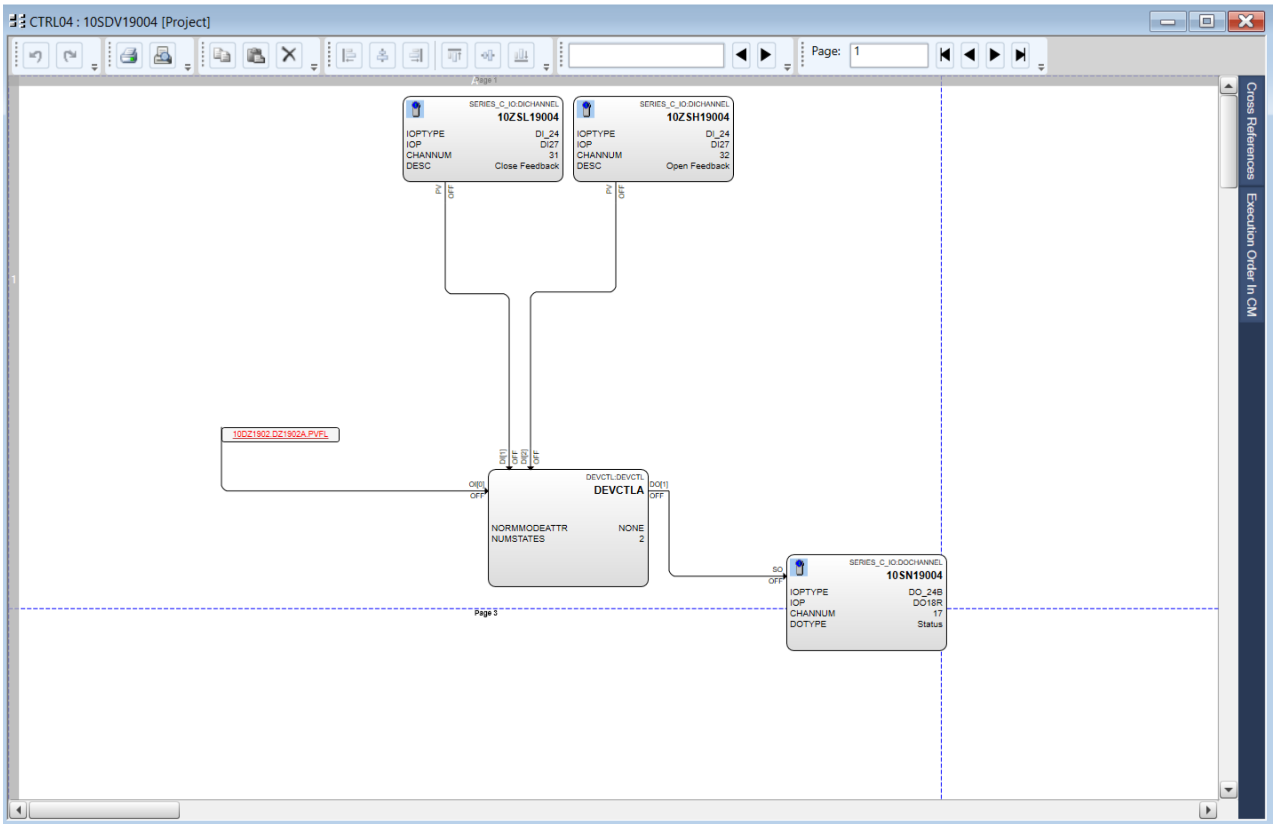
LocalMan

if LocalMan is ON, OP tracks PV, if PV is in a settable state (State0, State1, or State2). If PV is in an unstable state (Null or In Between), or PV does not exist, OP is set to Safe OP.

Override/Permissive Interlock Bypass

To grant an operator the ability to bypass the Permissive and Override Interlocks for a DEVCTL block, the parameter BYPPERM must be set to ON. The operator can then set or reset the parameter BYPASS.

BYPASS does not affect the Safety Override Interlock (SI).



Device Control Module

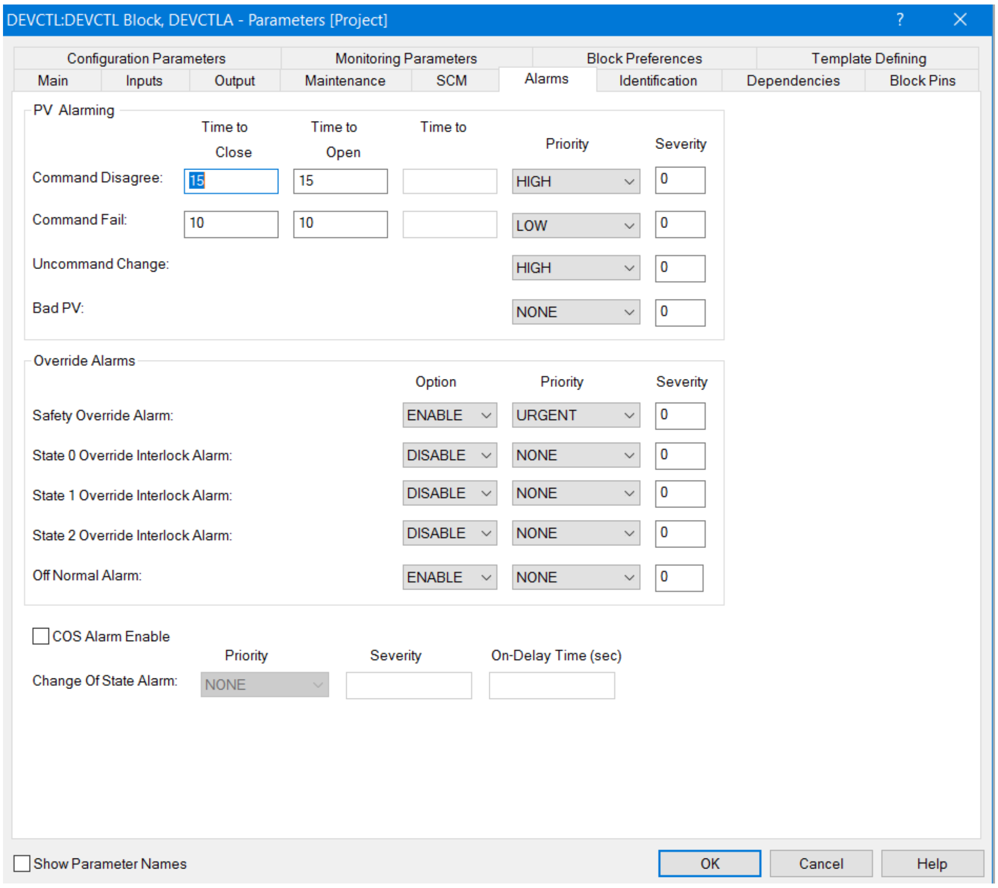
Alarms

- Command Disagree (CMDDISALM.FL):

  This alarm is generated when the commanded output state (OP) changes and the feedback state (PV) does not change to the same state within the specified feedback time. This alarm returns to normal when the PV state becomes the same as the OP state.
- Command Fail (CMDFALALM.FL): This alarm checks to see if the PV state changed from its original state to any other state within a specified feedback time after the OP state is commanded. For slow responding devices, absence of this alarm indicates that the device responded to the command, even if it has not yet moved to its commanded position.
- Uncommanded Change (UNCMDALM.FL):

This alarm is configured in conjunction with the Command Disagree alarm function. This alarm is generated, if an OP state has not been commanded and the PV state changes for any reason except BADPV.

- Bad PV (BADPVALM.FL): This alarm is generated whenever PV is detected in the Null state. The Null state can result from a Bad PV condition for an input provided by a source block, or because input combinations represent a Null state as defined by the DIPVMAP[0..15] parameter.



9.11.10 Typical P1 (Single Motor With ESD Interlock)

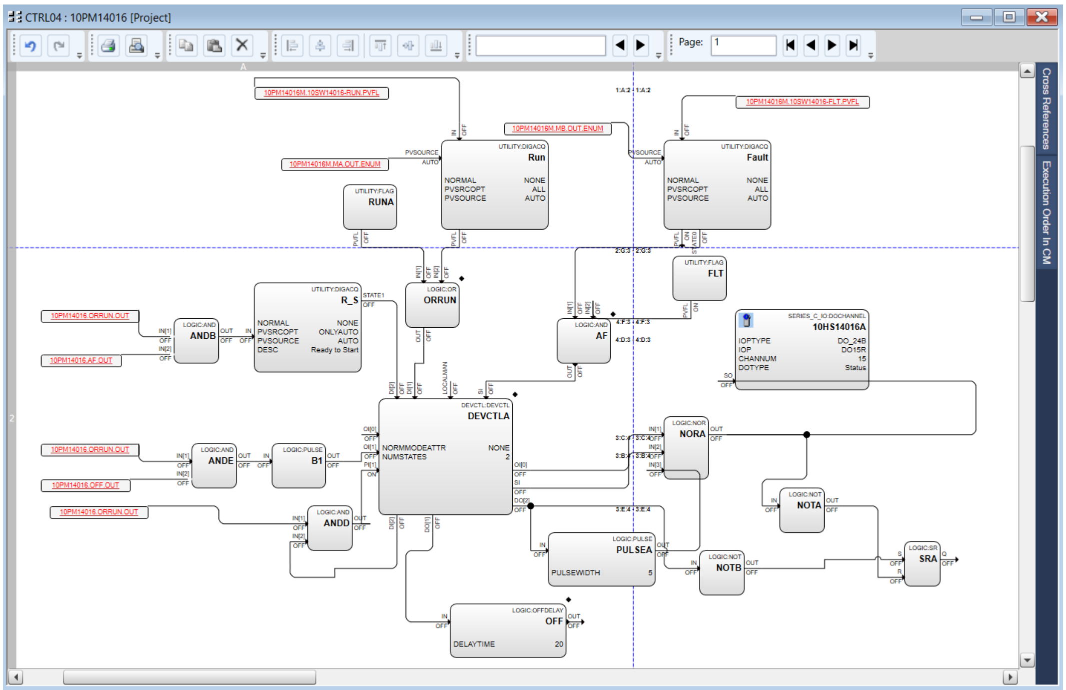
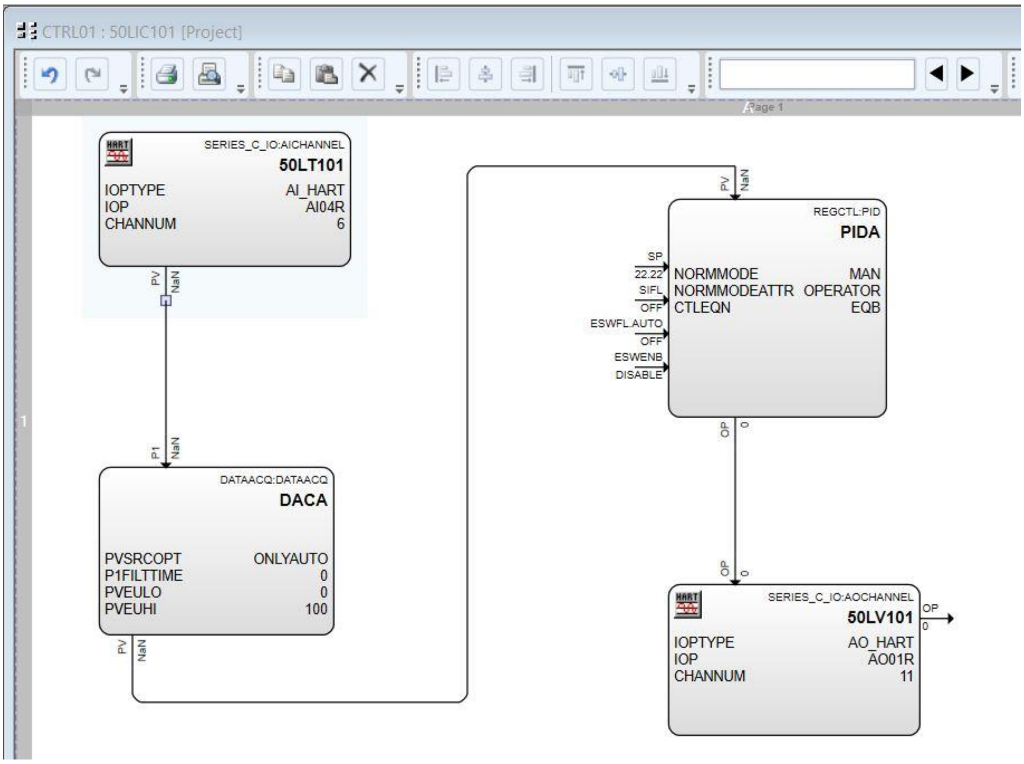


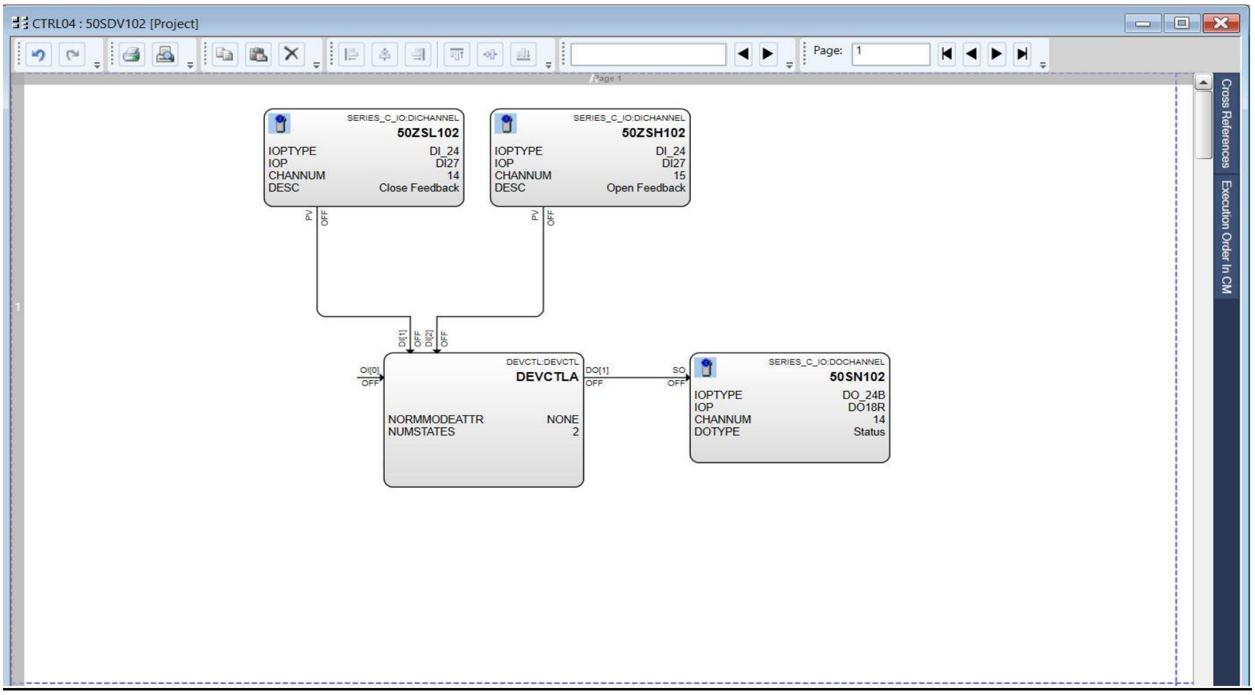
Figure Typical P1 (Single Motor With ESD Interlock)

9.11.11 Control valve



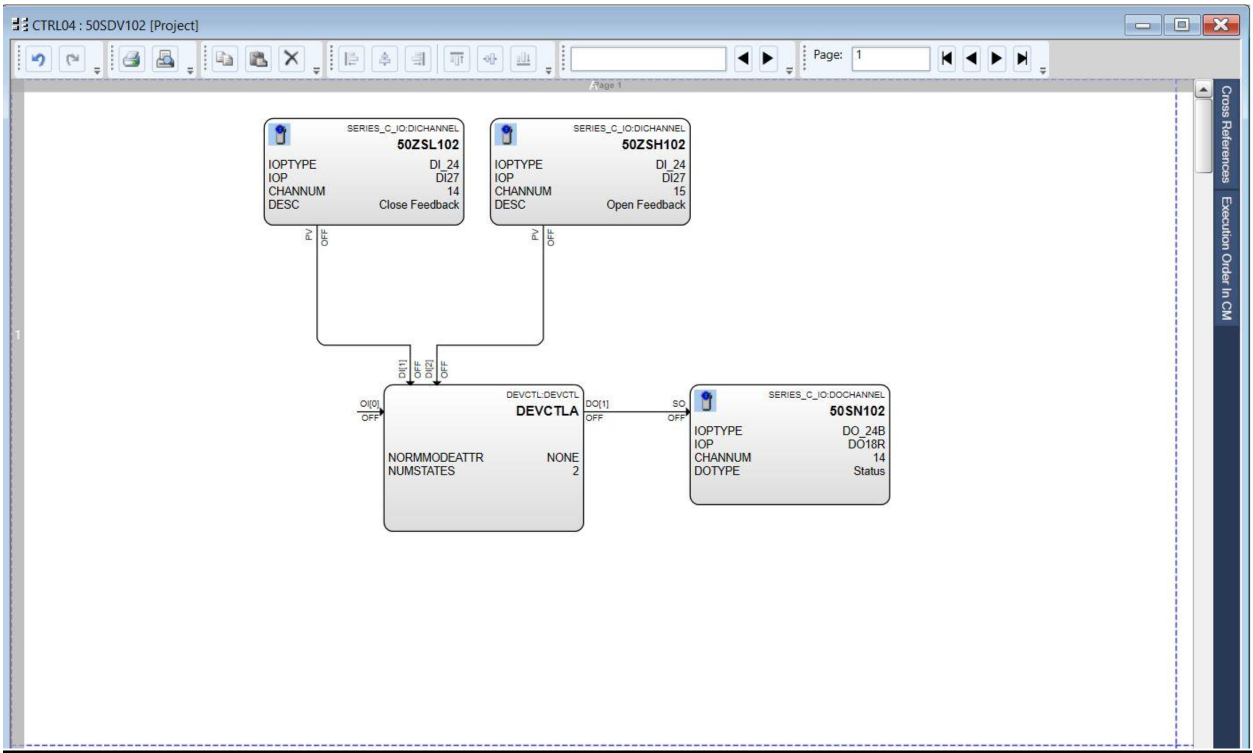
50LIC101

9.11.12 MOV valve



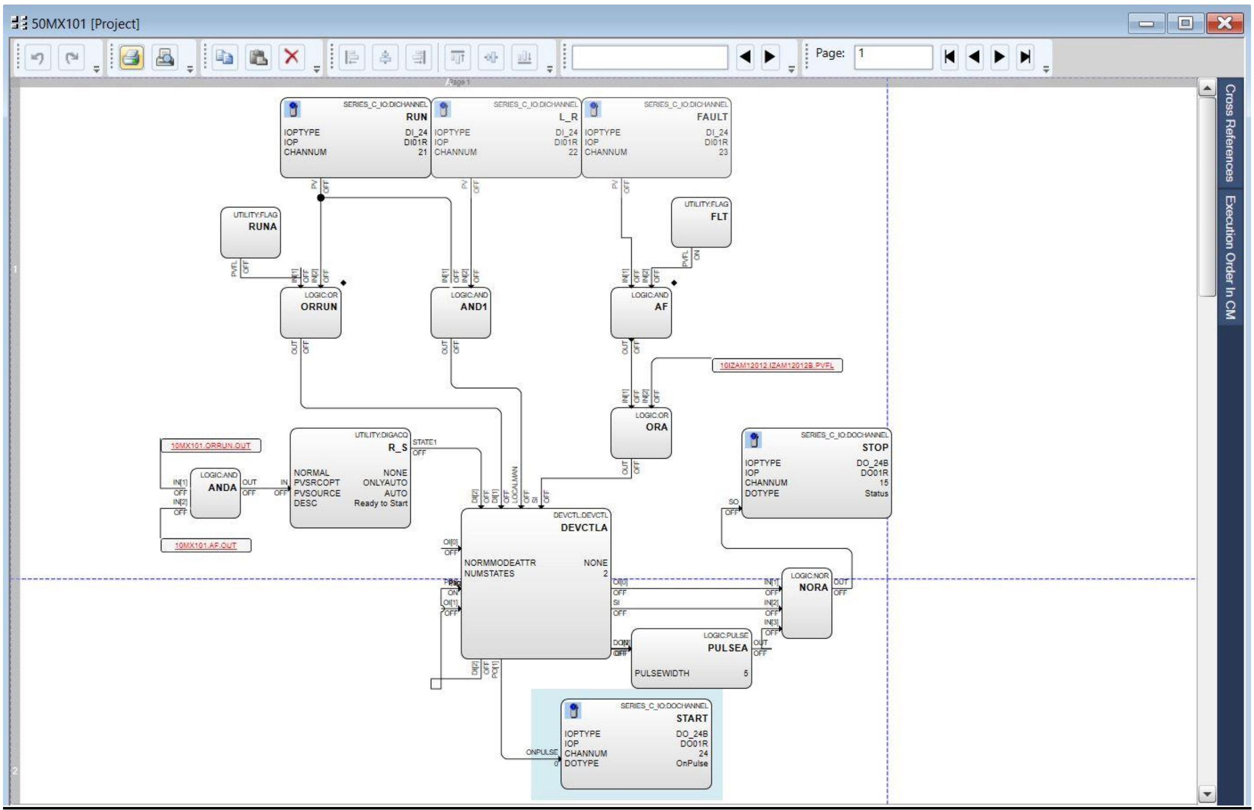
50MOV101

9.11.13 On/off valve



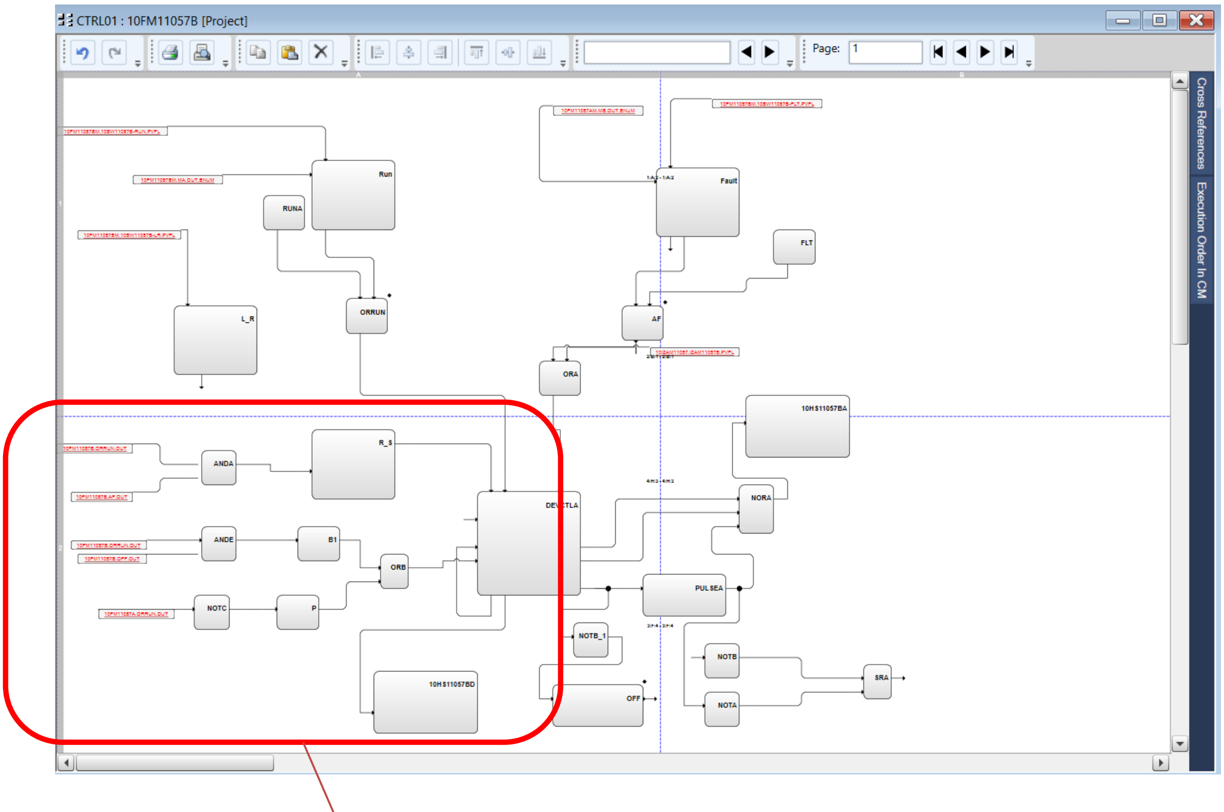
50SDV102

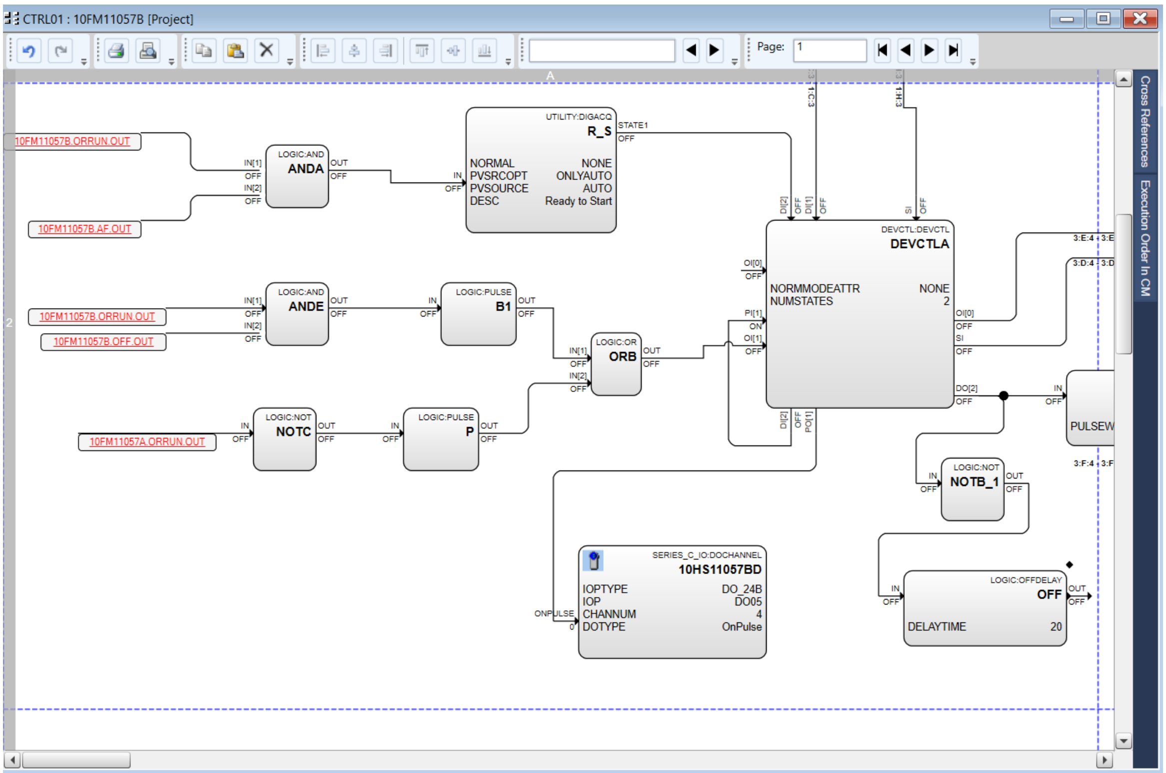
9.11.14 Agitators



10MX10

9.11.15 Duty Standby





The only different between duty standby (PUMP,MOTOR,FAN,...) and the single one is in DEVCTL pin(OI[1]) that its linked to the other stand by device in case of that if the ON device going OFF , the standby device going ON.

10 C300 restart

- RAM Retention Restart in Embedded Controllers

The C300 controller is to support a higher level of functionality for RAM Retention Restart (RRR)

General principles for RRR in CEE / C300 are as follows.

- Handling of CEESTATE following RRR is subject to a configuration option
- Configuration can force CEESTATE to Idle
- Configuration can allow CEESTATE to return to its last state before power down
- For repower to CEESTATE = Run, configuration can force CEE to execute either cold or warm restart

- Alternatively, configuration can cause cold or warm restart to be selected based upon the length of a configured power down interval

- Upon RRR individual blocks execute either a cold or warm restart based upon the behavior of CEE and the configured behavior of their parent control module

Further details for C300 support of RRR are described in the following sections.

The following behaviors are observed by all blocks upon execution of cold or warm restart. The same applies to RRR.

- PV handling on warm and cold restart

PVs are initialized to their fail safe values. For PV algorithms which involve historical computation state (filters, delays, etc.) all history is re-initialized.

- Alarm handling on warm and cold restart

When a CEE or IOLIM is restarted, any alarms reported by that node acquire the time stamp that goes with the time of restart. The timestamp of original report is not retained.

There are general principles which apply to restart behaviors for wide classes of blocks. These behaviors apply to RRR as well as to cold and warm restart.

Parameter RRRCEESTATE of the CEE block allows the application engineer to configure the CEE start up behavior to be performed upon RRR. It has an access lock of Engineer. Possible values for RRRCEESTATE are as follows.

• IDLE

CEESTATE goes to Idle upon RRR regardless of the value it had before power down. Operators may command a transition to 'Run' thereafter. The transition to Run may be commanded as either a cold or warm restart.

• LASTCOLD

CEESTATE goes to the last state it had at time of power down. If CEESTATE returns to Run then the transition is treated as a cold restart.

- LASTWARM

CEESTATE goes to the last state it had at time of power down. If CEESTATE returns to 'Run' then the transition is treated as a warm restart.

- LASTTIMEOUT

CEESTATE goes to the last state it had at time of power down. If CEESTATE returns to run then the transition is treated as either a cold or warm restart depending upon the duration of the preceding power down. The threshold for selecting warm or cold start is configured in parameter WARMTIMEOUT.

Parameter WARMTIMEOUT has the following characteristics.

- WARMTIMEOUT

This is a configurable floating point parameter which indicates a time value in units of minutes. If RRRCEESTATE = LASTTIMEOUT and the last CEESTATE was Run then upon RAM retention restart CEE examines the duration of the preceding power down. If power down duration was less than WARMTIMEOUT then CEE executes a warm restart. Otherwise CEE executes a cold restart.

The access lock of parameter WARMTIMEOUT is Engineer.

CM impact

A CM always executes either a cold or warm restart when RRR occurs. If the CM's CEESTARTOPT is configured to ALWAYSCOLD then the CM executes a cold restart upon RRR. If CEESTARTOPT is configured to ALWAYS WARM then the CM executes a warm restart upon RRR. If CEESTARTOPT is configured to FOLLOWCEE then the CM executes whatever type of restart results from the configuration of CEE parameters RRRCEESTATE and WARMTIMEOUT.

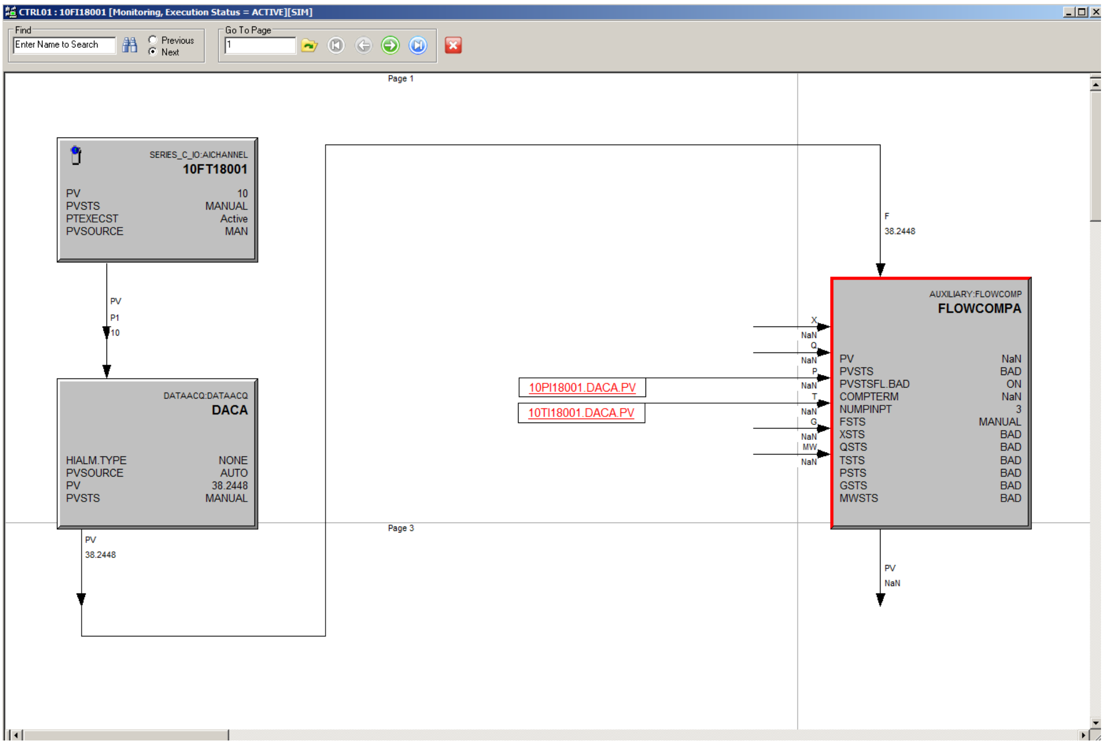
SCM impact

Like CM, an SCM always executes either a cold or warm restart when RRR occurs. The value of CEESTARTOPT determines whether RRR will always result in SCM cold restart, always result in SCM warm restart or whether SCM will follow the restart commanded for CEE as a whole.

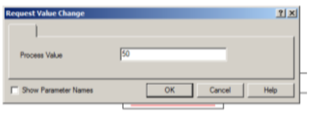
11 Simulation procedure

After preparing the controller logic its should be test to be reliable this process is applicable by simulating in Experion software.

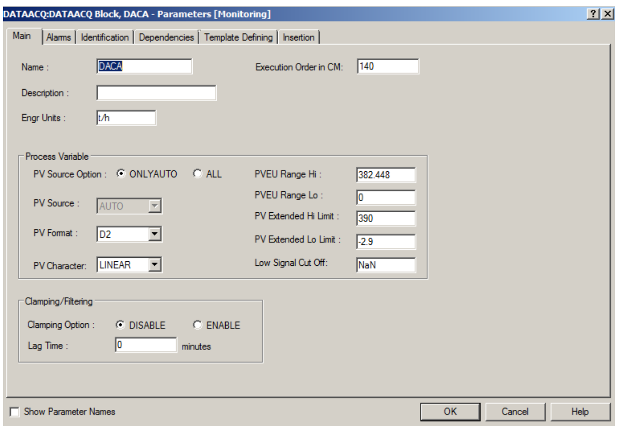
At first choose a control module , download controller and control module , after that open the control module . For example its shown a flow comp control module in this picture .



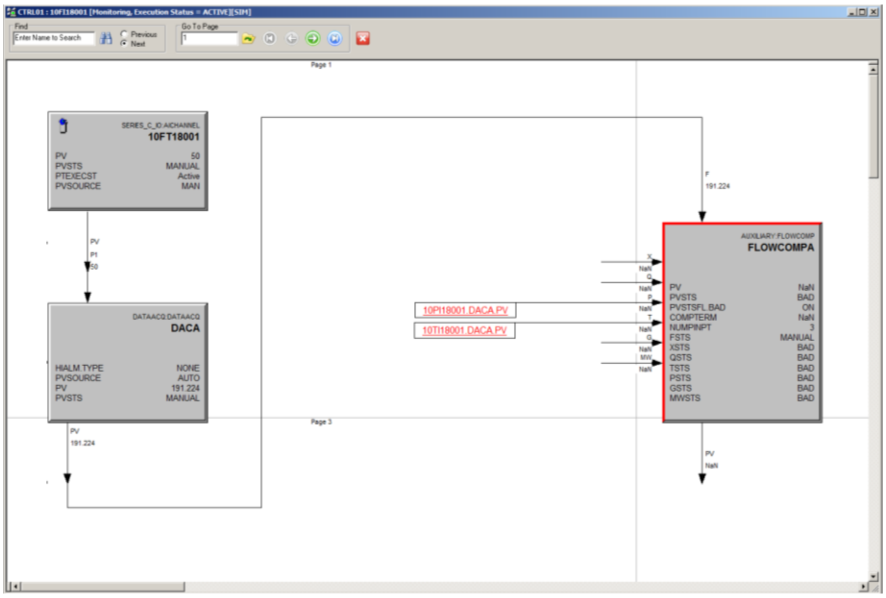
By clicking on out put of transmitter you can change the OP amount.



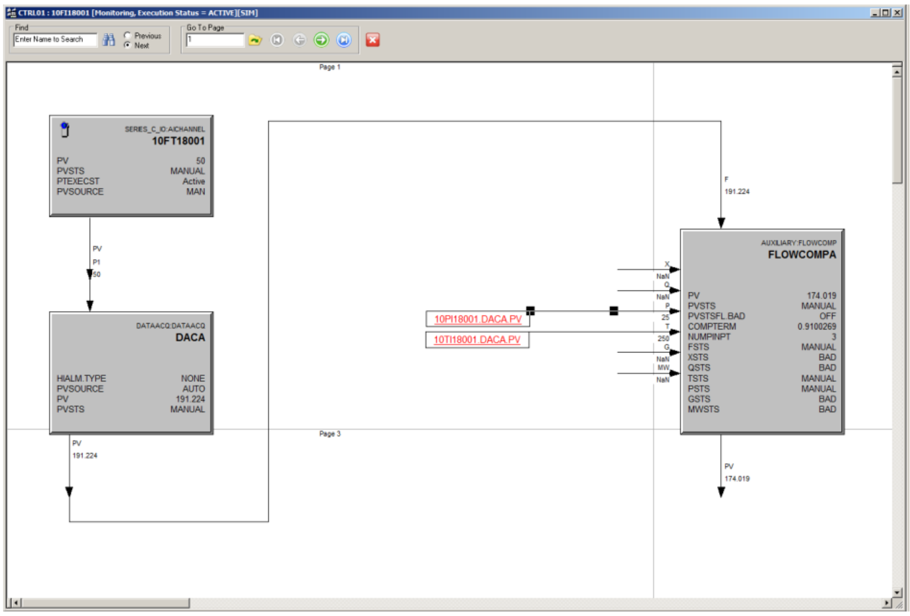
For example we set the OP on (50%) , so its will be read by DACA and DACA block will calculate the real amount by the define range in this block.



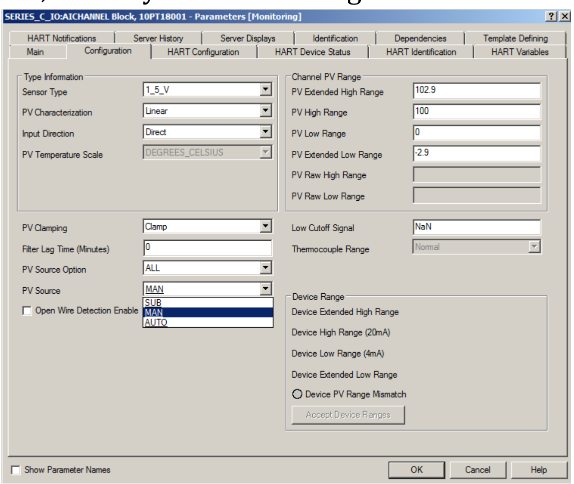
In this case range is between 0 to 382.448 , so the output will be 191.224 as see in below picture.



By changing related (PI & TI) the FLOWCOMP block will be shown the result.



Be noted the for changing OP , at first you should change the block source from AUTO to MANUAL.



12 PKS Software Standard Functionality

12.1 Server

12.1.1 Data Global Ownership

Global means that there is one-and-only-one owner of any particular data element across the entire automation system.

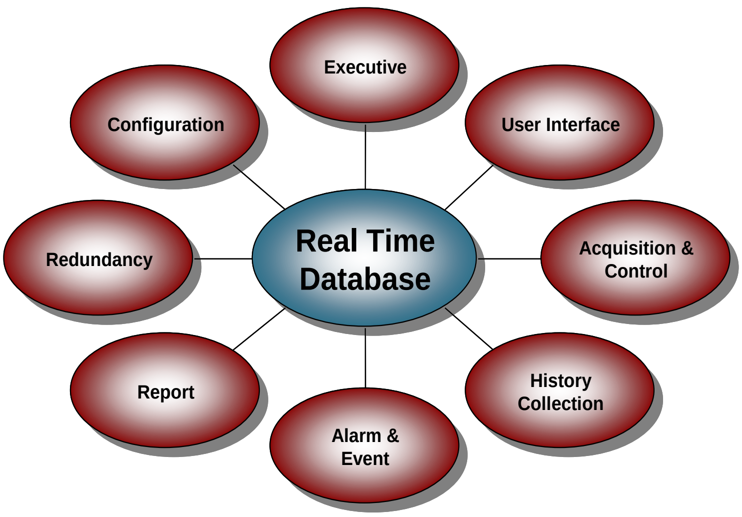
Within the Experion PKS system, some data is owned by a controller while other data is owned by a server-based database known as the System Repository. Each data element is owned by one-and-only-one of these entities. This provides robustness because all users throughout the Experion PKSTM system, whether in the control layer or the supervisory layer, are dealing with the same value for that data at any given point in time.

12.1.2 Real Time Database

1) At the heart of the Experion PKS server is the Real Time Database. The following information is stored in the Real Time Database:

- Acquired Data – data read from or related to controllers
- Process History – historical store of acquired data
- Alarms and Events
- System Status
- Configuration Data – details how Server subsystems are configured to operate
- User Defined Data – structures to store application specific information

To maintain data integrity, memory resident portions of the Real Time Database are periodically written to the hard disk in a process known as checkpointing.



12.1.3 Alarm/Event Management

All on line Alarms are stored in the server's real time database.

For , ESD and 3rd party packages, the controllers will generate alarms but the time stamping will be done at the server.

All alarm limits are specified in the controller's configuration and transferred as alarm points to the database in the server. All the alarms will be displayed together in the alarm summary page and stored in the same format.

Following priority levels will be used for segregating alarms for display and reporting:

- Journal (alarm not reported in alarm summary page but stored in events)
- Low
- High
- Urgent (Emergency alarms)

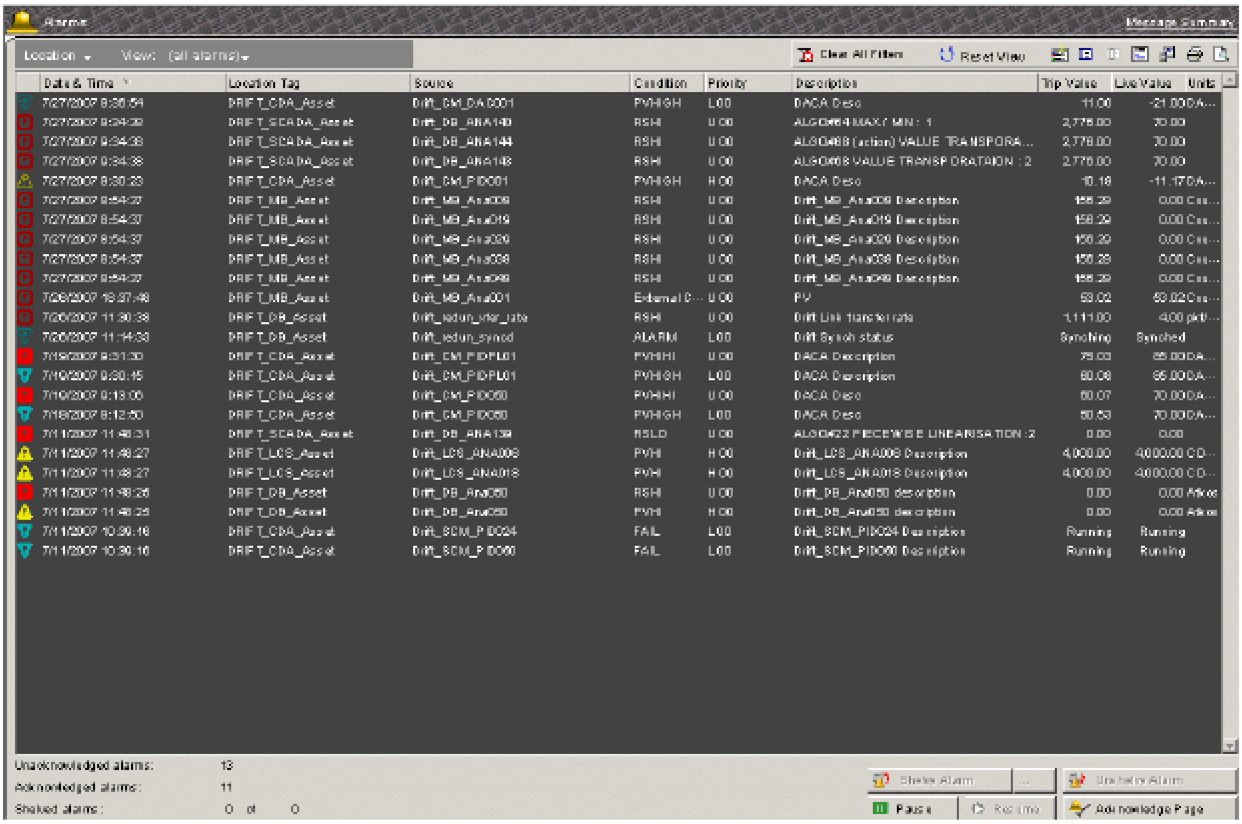
Low alarm is having the least priority and the urgent alarm is having the topmost priority. Low, high and urgent alarms are displayed on the alarm summary pages and also indicated in the events log.

Sub-priority / Severity number, between 0 (lowest) and 15 (highest), may also be specified for each alarm priority. Alarms having the same priority can be distinguished by assigning sub priority values. The high sub priority alarm will be displayed first, then other alarms which have the same priority but lower sub priority will be displayed.

Priority levels will be defined by HXXXX within the data base. Severity number and area, group allocation will be defined during the functional analysis and also inserted in the data base.

An alarm line at the bottom of the operator station will display the latest highest priority alarm.

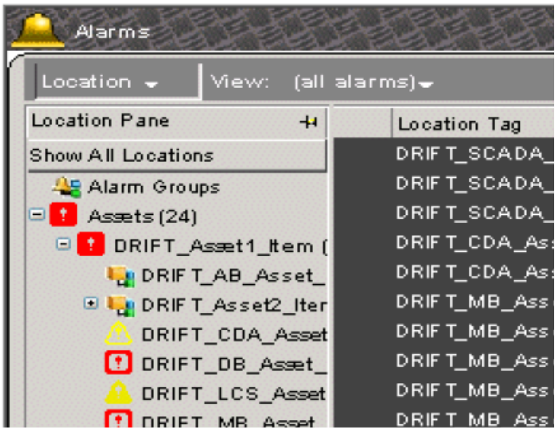
Alarm summary page can store up to 2000 alarms. If more then 2000 alarms exist at the same time then the alarms will be displayed on the first in first out basis.



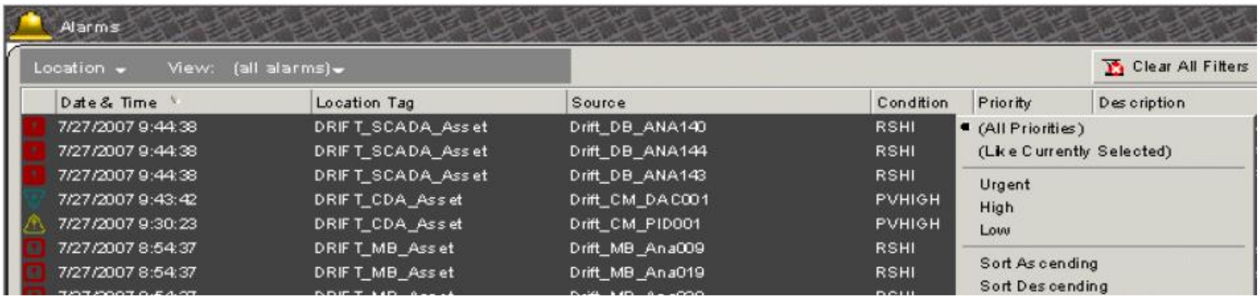
12.1.4 Alarm Reporting /features

Many options available to operators for quickly targeting the process problems include:

- Navigation to Associated display and point detail display - Operator can navigate to associated page or point detail page of the tag from the alarm summary page.
  - Double clicking on the alarm opens the point detail page
  - Navigation to the associated page (process overview page) by selecting the alarm and pressing the associated page button or press F2 button on the keyboard.
- Asset assignment – Asset will be assigned to the operator stations and the alarms for those assets will only be displayed on that operator station. Refer to the security section below.



- Operator log – Operator actions such as acknowledgement will be recorded in events so that all the actions taken can be stored.
- Alarm filtering – Alarms can be filtered by any column that is displayed on the alarm summary page. For example the bellow filter will be applied only for urgent alarm.



- Alarms related with the Experion PKS system and alarms related to the process are displayed on different pages. The system alarms are notified on the system page. A separate link to the system alarm page is provided beside of the alarm summary page link, on the status line of each operator station.
- The alarm display will show the alarm only once but it will show the number of times the point went in the alarm state. If the alarm is acknowledged and returned to normal the alarm will be removed from the alarm summary page and line will be generated again the next time alarm state occurs.

12.1.5 Events

The Event Summary will list events that occur in the system, including:

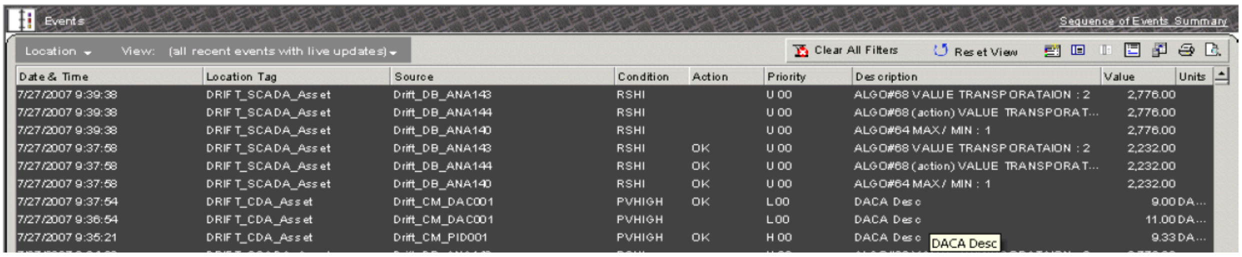
- Alarms
- Alarm Acknowledgments
- Return to Normal
- Operator Control Actions

- Operator Login & Security Level Changes
- On-line Database Modifications
- Communications Alarms
- System Restart Messages

Up to 30,000 events may be stored in the Event Summary. After that the events will be retained on the first in first out basis. Events are collected in an Experion system database file. Periodically events are copied from the Experion system database to the Microsoft SQL Server event database for queries and reporting. Events are archived from the SQL Server database according to a schedule specified. The event archive will be stored on different partition of server for maximum efficiency.

Event Achieve

The event achieve will be generated on the D partition under the directory of "event".



12.1.6 Historization

History Collect

The points defined in CCRSVR01 and CCRSVR02 PKS Server will be configured to use standard history, as noted below:

1-minute snapshots

6-minute averages (of the 1-minute snapshots)

1-hour averages (of the 1-minute snapshots)

8-hour averages (of the 1-minute snapshots)

24-hour averages (of the 1-minute snapshots)

History archive

The history archive will be generated on the D partition under the directory of "history". After each 7 days, the archive file will be moved to a network storage automatically or be moved to a portable storage manually.

History Archive setting: Station-> Configuration->History->History Archive

NOTE:

The PKS system default history archive path is :

"C:\Program Files\Honeywell\Experion PKS \Server\data\Archive" ;

When the number of samples stored equals the maximum number that can be stored, the oldest history

Sample is discarded in order to store a new sample. The oldest history sample may need be saved for future analysis. The history archive will be set such that the history files for the nominated history intervals are copied to a default folder on the hard disk of the server, permitting off line viewing of archived data.
The folder of "history" on the D disk partition will be chosen to archive historic data. The free disk space limitation of 5000MB will be set such that once the free disk space on the archive partition is lower than the limit, an alarm will be raised and archive function will stopped until disk space is returned to below the limit.
If longer history storage is required, the history archive can be moved to a network or portable storage .
The archived folder's name contains the date and time at which the history was archived and the type of history that is archived for easy reference after archiving.

Once collected, historical data is available for use by:

- Trend displays
- Custom displays
- Reports
- Application programs
- Spreadsheets

History Restore

The PKS system default history restore path is :

"C:\Program Files\Honeywell\Experion PKS \Server\data\Restored Archives

Periodic history:

Periodic history collects and stores numerical data at predefined regular intervals. Periodic history data is generally used for operational purposes such as trend monitoring but is also collected for historical analysis.

Experion has a comprehensive range of collection rates for periodic history. These collection rates provide a high degree of flexibility in moderating the load on your control network.

- Fast history stores snapshots of a point parameter at short regular intervals. You can choose from 8 different collection rates. Fast history has a base collection rate of 1 or 5s. By default, the fastest collection rate is 5 seconds but this can be changed to 1 second if necessary. Total number of point parameters assigned to fast History should not exceed the system limit.

- Standard history stores snapshots at slightly longer intervals, ranging from 1 minute to 30 minutes. The fastest standard history collection rate of 1 minute can be changed to 30 seconds if necessary. In addition to storing snapshots, standard history also calculates and stores average values, based on the standard history snapshot rates. The default averages are: 6-minutes, 1-hour, 8-hours, and 24-hours.

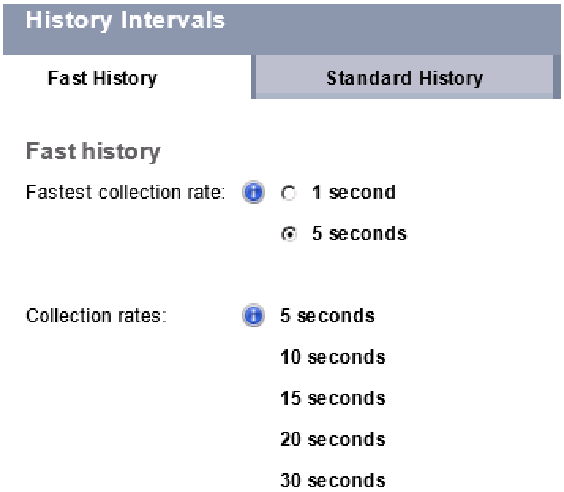
- Extended history stores 1, 8, and 24-hour snapshots.

  Note that history collection is synchronized to the system time. For example, the 6-second collection occurs at 6, 12, 18 and so on seconds after the minute boundary on the system.

For some specific process points, fast history could be required.

Fast history will be defined for the project with a 5 Sec collection fastest rate (default).

Caution: changing the default setting to 1 sec induce system warning message:”Increasing history collection will impact system performance” (server load)



System trend interval can be selected as 1 sec (or 5 sec*), 1min, 6 min avg, 1 hr avg, 8hr avg, 24 hr avg, 1 hr, 8 hr, 24 hr.

* The 1 sec interval is fast history. A project can be setup to have either 1 sec OR 5 sec fast history collection.

12.1.7 Trending

Trend Configuration allows trends to be configured on line as necessary by simply selecting the point and the parameter from the database.

Trend displays are standard Station displays that provide a way of viewing historical data for points. Trend displays complement other types of displays that can be used to view historical data such as point detail trends, group trends, and custom display trends.

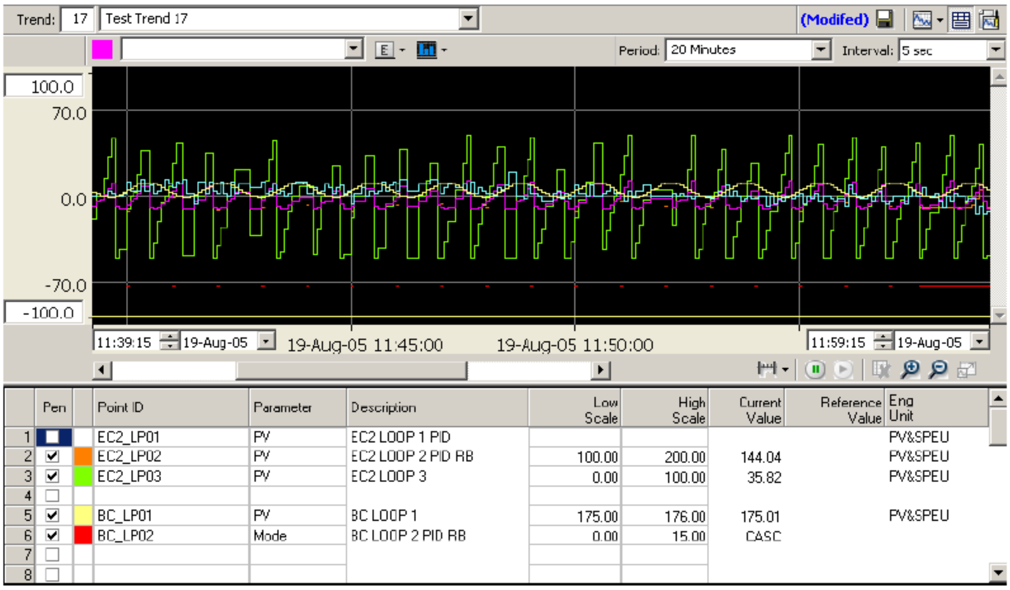
The history archive files are automatically searched to display archived history data for specified times and dates on a Trend display.

According to choose fast , standard and extended mode in history , different trend will be shown ; if fast mode being choose sampling will be done more standard so its support short

Standard trend types include:

- Single bar graphs
- Dual bar graphs
- Triple bar graphs
- Multi-plot trends – Can display up to 32 parameters simultaneously.

- Multi-range trends – This trend is similar to the multi-plot trend except that the range can be defined separately for each parameter. This will be useful if we want to enlarge some portion of the trend for only one parameter.
- X-Y scatter plots,
- Numeric tables
- Group trends



The trends will be configured on the Multi-plot Trend standard base except specified otherwise in the functional specification.

Trends reference from 1 to 500 will be reserved for standard trends and operators will not be able to modify their parameters allocation. Predefined standard trends will be provided. Refer functional analysis. Trends referenced above 500 will have parameters configurable by Operators.

Following Functions will be provided:

- Combination real-time/historical trending – in the same trend the history values as well as the current values will be displayed and will be continue..
- Trend zooming, panning, and scrolling – the trend can zoomed by dragging cursor or it can be scrolled in backward or forward direction
- Hairline readout - Fort the tags that are histories , by clicking on the trend the value at that instance can be seen in the trend
- Declutter – Declutter feature allows to temporarily enable or disable the particular “Point. Parameter” value in the trend without removing the entry of the particular point.

- Configurable trend density – The number of samples displayed for the particular type of history can be specified for the trend.
- Simple recall of archived history - Archived history may be accessed automatically by simply scrolling to, or directly entering, the appropriate time and date.
- Trend protection – Trend can be protected so that operators will not have access to modify the configuration of the trend like "Point Parameter" information.
- Smart clipboard support for copy/paste of data – Data from the trend display can be copied and pasted to excel sheet for further analyze.

Note : About short trends, long trends and related time resolution and time period of them will be discussed during prototype test meeting.

- **Limitation**

- The number of trends available in the system depends on the server database sizing. By default the system will be sized with maximum 3000 trend display page per server.
- Each trend display page can have maximum 32 parameters assigned.

Trends on custom displays performance specifications	Limitations
On a single Display	
Maximum number of full trends[1, 15.16]	1
Maximum number of basic trends[2, 15,16]	4
On a single Station[3]	
Maximum number of trends[4]	8
Maximum number of standard history trends[5]	8
Maximum number of fast history trends[6]	4
Maximum number of pens across all trends[8]	64
Across all Stations[9]	
Maximum number of trends[10]	28
Maximum number of standard history trends	28
Maximum number of fast history trends[11]	14

Maximum number of pens across all trends[8,17]			224
Trend call up time[12]			
Sample interval	**Call up time per Trend object**[13]	**Number of pens**	**Trend period**[14]
1 second fast history	3 seconds	1-8	1 hour
	3 seconds	9-32	20 minutes
5 second fast history	3 seconds	1-8	4 hours
	3 seconds	9-32	1 hour
1 minute standard	10 seconds	1-8	2 days
	10 seconds	9-32	1 day

Note 1 - A full trend is a complex, flexible object with multiple views such as Trend with Legend, Trend with Events or Tabular history.

Note 2 - A basic trend is a much simpler object that only includes the plot area and axes. It is recommended to use the basic trend when adding more than one trend object to a single display

Note 3 - Either single of multi-window Station.

Note 4 - From here on trend refers to both full trends and basic trends

Note 5 - Standard history default is one minute collection rate

Note 6 - Fast history default is five second collection rate

Note 7 - It is recommended to have as few trends but with many pens per trend on a custom display

Note 8 - Placing a history parameter alphanumeric on a custom display induces the same load as a trend with one pen on it

Note 9 - Maximum number of Stations includes DSA connected Stations calling up trends with pens that reference data from this server

Note 10 - It is recommended that the history files do not have large amounts unnecessary space. For example, if historizing 1500 point parameters in standard history, the standard history files should be sized for 2000 point parameters rather than10000 point parameters.

Note 11 - For performance reasons it is recommended to either not have fast history archiving or to minimize the amount of archived fast history for current process analysis only.

Note 12 - Trend call up time is for the largest system (i.e. maximum number of fast and standard history assignments). Fast call up time is expected on smaller systems.

Note 13 - This is the call up time on a single Station. It is expected that call up time would be longer if trends were called up on two or more Stations simultaneously.

Note 14 - This is the recommended maximum trend period for the specific sample interval and number of pens

Note 15 - A subscription is setup to the process control network at the Station update rate for each pen on a trend.

Note 16 - A Limit Trend object with default view settings corresponds to a Basic Trend for the purpose of these capacity constraints. If any additional view options are enabled a Limit Trend object corresponds to a full trend.

Note 17 - For a Limit Trend (either system or custom), the number of pens includes the trend plots configured by the engineer but does not include the limit plots automatically shown by Experion.

According to what was discussed in this section and the history section regarding the different times that can be selected for sampling, from this issue, we can also enter the topic of short trend and long trend, in such a way that the sampling time being, the number of samples increases, as a result, a shorter period of time can be covered, so short trend is available, and the opposite of this is applicable for long trend.

12.1.8 Reporting

Reports will be configured by operators from the supervisory system which provides many built-in reporting functions. These functions can be segregated in two types. The report outputs can be directed to screen, printer or file.

- Standard Reports

- Custom Reports

Standard reports – The output format of standard reports is fixed Honeywell forms and the contents will be configured by Operators. Contents would be like reports for some tags or area for specified time period will be reported in the standard format. For defining contents in all the system reports wild characters like “*” and “?” can be specified. These reports will be configured to execute on operator demand.

- Alarm/Event report - Alarm and event details from the event file. This report enables analysis of alarms and events that occurred during a specified time span on specific points. The report will collect the information for the duration specified in the configuration. For these reports, the reporting period shall be specified and may be:
  - Specific time period - By specifying start and end date and time.
  - Relative time period - Specified by the time span backwards from the current time.
- Alarm Duration report - Reports the time of occurrence and elapsed time before return-to normal for specific alarms in a specified time period. The time period is specified as for above alarm/Event report. It is possible to use filters to limit the report to a specific area or specific point
- Point Cross-Reference – This report determines database references for the points specified to enable easier system maintenance when points are decommissioned or renamed.

Custom reports – Each report is fully custom built for format and content and each have a definition file. These reports can be generated on time basis or invoked directly by operator command.

- Integrated Excel Report – In this type of report the definition file is created using Microsoft Excel. The report output will be as defined by operator in the definition file. The dynamic data will be filled in the Excel file to generate the report. Microsoft Excel can access the Real time database of the SCADA Sever using the Open Data Access option provided in the system.
- Free Format Report – This type generates reports in text format, which may include math and statistical functions such as Max/Min and standard deviation.

12.1.9 Security

2) Security in both system integrity and access to control information is of a primary concern with open systems design. To maintain system security, Experion PKS provides configurable security levels, control levels and asset

assignments. These may be configured for each individual operator or alternatively for each operator station. Therefore, Experion PKS™ offers two types of security:

- Station-based security (Password for going to each level above operator level.)
- Operator-based security (Password for each operator and a fixed level assigned to user.)

3) In Station-based security, there are security levels that are used to restrict operator authority within Station.

4) There are two aspects to operator-based security: authentication and authorization. Authentication is the process of verifying that a user is known to the system, while authorization controls what a known user can do within the system. Accounts are used to restrict access and authority within Station. It is possible to use traditional operator accounts or integrated accounts can be created.

5) For traditional operator accounts, the Experion PKS server authenticates the user against credentials stored in its database. Authorization is also controlled by Experion PKS using security levels and, if applicable, assets.

6) For integrated accounts, Windows authenticates the user on the Server computer against the Windows user account. Authorization is then controlled by the Experion PKS server using security levels and, if applicable, assets. Using integrated accounts enables the plant to be configured by the:

- Use existing enterprise-wide security policies
- Use single sign on

7) Minimize the number of accounts required for operators, Use Sign on Manager, Use Windows auditing to track user activities.

8) Up to six security levels govern operator access to Experion PKS functions as shown in the table below.

Security Level	Functionality
Level 1	Signed-off mode
Level 2	View only mode with alarm acknowledgment
Level 3	Level 2 plus control of field parameters
Level 4	Level 3 plus field parameters of level 4, configure standard system infrastructure such as reports
Level 5	Level 4 plus user configured field parameters
Level 6	Unlimited access

9) Operator-based security provides up to 255 control levels to further refine operator control access to individual items of plant and equipment. Any actions initiated by an operator are logged in the Event database against an operator identifier. In addition any control actions to a given point is only allowed if the control level configured in the operator profile exceeds the level assigned to the point.

10) Individual operator profiles, including security levels, control levels and area assignments, are activated when operators sign on to the system. In addition, area profiles can be created enabling plant areas to be enabled or disabled for control, between certain time and date criteria.

For example:

The operator will be assigned an account belonging to Level 2. The engineer will be assigned an account belonging to level 4.

Proper assets will be assigned to operators with the proper restrictions. The operator has no access to assets to which he has not been assigned.

Following restrictions can be given to the operators for accessing each asset

- No access – If an asset is not assigned to the operator then he cannot even view the points or alarms from the asset.
- View only permission – The operator can only view the points from this asset
- Acknowledge only permission – The operator can view the alarms and acknowledge the alarms but cannot change any parameters value from the asset.
- Full Control – Operator can view, acknowledge and change parameter values for the points under assigned assets.

	Station-Based Security	Operator-Based Security		
		Traditional Accounts	Integrated Accounts	EPKS Windows Group Accounts
Login	No Login	Server Login	Windows Login	Windows Login
Asset Assignment	Assigned to Station	Assigned to Operator ; NOT to Station		
Access Level	Oper Supr Engr Mngr	Viewonly Ackonly Oper Supr Engr		

		Mngr

12.1.10 Account and password

The following windows users have been specified and are configured with varying level of system access:

- Administrator
- mngr
- oper

The Administrator account (System Administrator) will be assigned the password of “Honeywell” for all the PC nodes. The account will be used to maintain the windows system but not used for daily operation.

The mngr account (PKS Manager) will be assign password of “mngr123” for PKS server. The account will be used to start various PKS system services, but is not used for daily operations.

The oper account (PKS Operator) will be assigned the password of “oper” for all the operator station PCs. The account of oper will be limited by windows security and used to let the operator start Station software.

12.2 Communication with Other Systems:

12.2.1 To Third Party Package- MODBUS RTU

Communication with the third party systems will be using Modbus RTU protocol over serial Link:

Modbus RTU Protocol over serial link

Where using Modbus Protocol over serial link, connection will be across the terminal server and then with Ethernet to the switch. The terminal servers will have serial links to communicate with the 3rd party package devices and the Ethernet link for communication with the PKS server.

The PKS server will act as Modbus master for communication with 3rd party devices.

Modbus protocol can be Modbus RTU

The vendor for the package has to ensure that the proper communication can be established with the PKS server.

12.3 Time Synchronization

The time synchronization utility will be provided with Honeywell PKS software, to execute the NTP setup in PKS server or station.

In PKS server, appoint the NTP server as an external time source to the GPS server, but in PKS station appoint the NTP source as its PKS server.

The following topics are important for understanding and planning the time synchronization requirements of your Experion PKS system.

12.3.1 Experion time requirements

Reliable and coordinated time is an important element in an Experion system. It is used in many aspects of the system including:

• control functions

• redundancy options

• intersystem communications of supervisory systems

Supported time protocols

When deploying a topology, the time protocols supported should be considered.

Device	CDA client	SNTP client	NTP client	PTP client[7]	CDA server	SNTP server	NTP server	PTP server
C300	X	X		X				
Series C FIM4 and FIM8	X	X		X				
PGM	X	X		X				
Safety Manager			X	X				X[8]
Wireless Device Manager			X				X[9]	
IEC 61850 Interface Module	X	X		X				
PMD		X	X					
C200	X							
RTU[10]		X						
CISCO switches		X	X					
Moxa switch (IEC 61850 compliant)		X	X	X			X[11]	X[11]
Rugged com switch (IEC compliant)		X	X	X			X[11]	X[11]
Windows client			X				X	
Windows server[12]			X		X[13]		X	
Windows domain controller			X				X	

7 PTP IEEE-1588 version 1 and PTP IEEE-1588 version 2 are supported for Safety Manager devices. For all other devices, only PTP IEEE-1588 version 2 is supported.

8 Downlink only

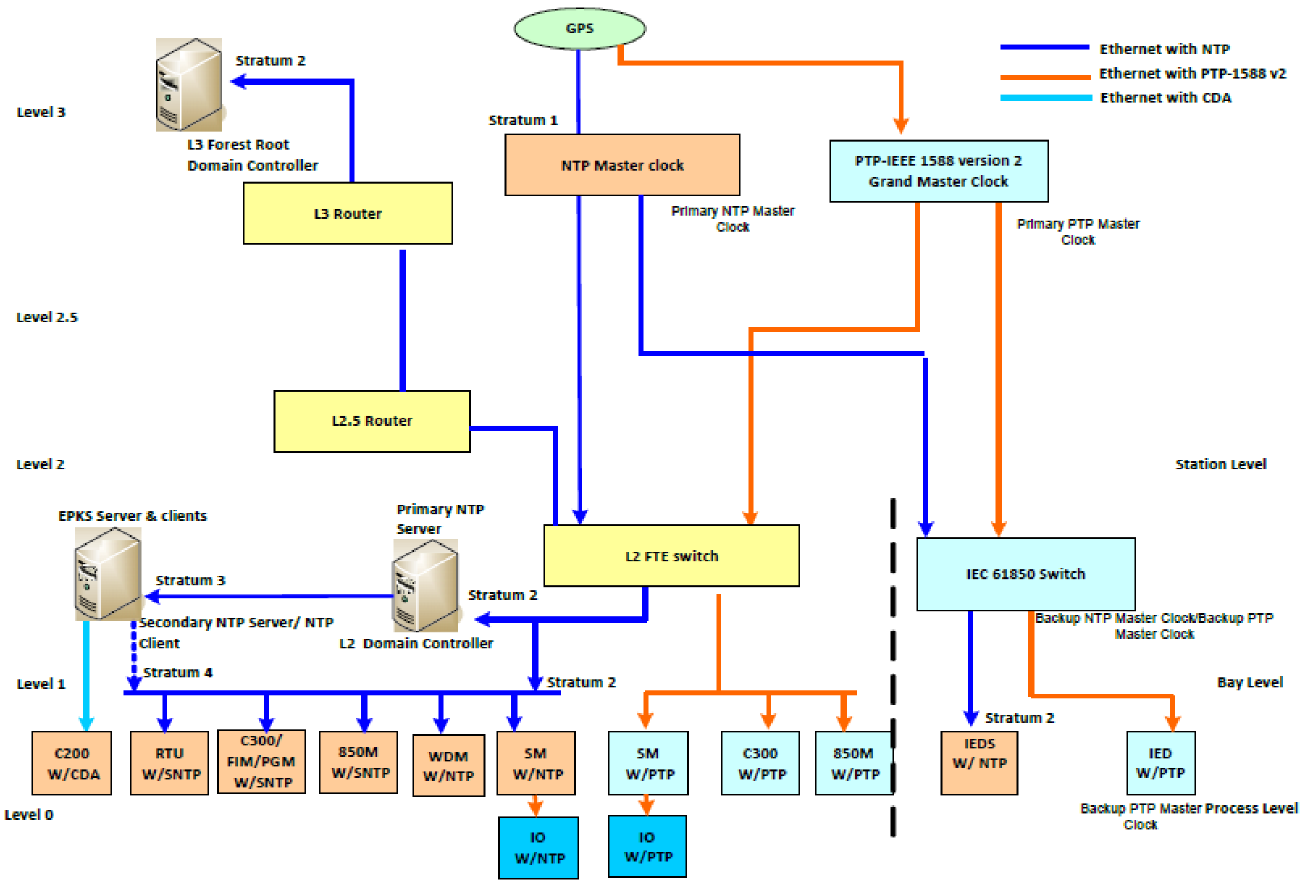
9 Limited to servicing Field Device Network(FDN) only

10 Also supports DNP3 for time synchronization

11 Limited to IEC 61850 networks only

12 Including ESXi hosts

13 With Experion PKS Server install



Typical time distribution topology that is supported by Experion connected devices.

12.3.2 Time synchronization in workgroups and Windows domains

By default, workgroups synchronize once a week and Windows domains synchronize during logon or authentication, or once every 8 hours or so if not logged in. Within a Windows domain, computers autoconfigure themselves for time distribution.

For more information, see http://technet.microsoft.com/en-us/library/cc749145(WS.10).aspx.

The default time synchronization mechanisms for workgroups and Windows domains do not meet control system specifications, which require localized, high-quality time synchronization.

Level 1 network

The C300 Controller, Series C FIM4, FIM8, PGM, IEC 61850 Interface Module, and Safety Manager require high-quality time served from a Network Time Protocol (NTP). C200 controllers receive time through CDA from the server.

The quality of time determines how tightly events from different devices can be correlated, especially with regard to digital Sequence of Events (DI SOE). For DI SOE, therefore, it is recommended that you use a high quality external time source such as an IEEE-1588 (Precision Time Protocol)-based time server.

The most stringent requirement for time synchronization at level 1 is from controllers that need high-resolution, digital sequence of events. An example of this requirement is in the electric utilities where thousands of digital input points are monitored. Each state change is time stamped to a minimum resolution of 1 ms. These state changes are logged. After certain faults, such as a generator going offline, the log is frozen for later analysis to see which condition led to the fault. Examples are turbine imbalance, drop in frequency of the grid, circuit breaker trips, and so on. Critical physical and electrical events follow each other over a short period of time and the log is used to identify which event happened first and caused the rest. Typically the cause is used to backcharge the responsible party.

Level 2 network

At level 2, time is a gating factor used in many supervisory functions such as:

- redundancy
- event processing
- system monitoring

Time stamping is also used in diagnostic messages. For example, Safety Manager notifies Experion of all abnormal changes in system behavior by means of time-stamped diagnostic messages.

12.3.3 Summary

Both levels provide important and critical functions to an overall system that is being used to control a plant's processes. System events can be generated from either the hardware side (level 1) or from the supervisory side (level 2). Therefore, time synchronization between the two network levels also needs to be coordinated, ideally, using the same source. This mitigates against messages and events being mishandled, regardless of where they are sourced from.

12.3.4 About time protocols

- **Time protocols and Microsoft Windows**

All supported Microsoft Windows operating systems use NTP. Each domain controller in a Windows Server 2003 or 2008 domain, by default, is an NTP server. The Active Directory provides a hierarchical time infrastructure. Each system added into the Active Directory/domain synchronizes time with a time source in the domains hierarchy.

- **About setting up time synchronization in your Experion PKS control system**

Because the default Windows NTP implementation is not set up for the tolerances needed for control systems, Honeywell recommends that you use the NTPConfig tool to configure time synchronization on your Windows nodes. The NTPConfig tool corrects the tolerance deficiencies and converts the Active Directory default settings to be compatible with control system requirements. To overcome the tolerances, parameters are updated to maintain more stringent time synchronization.

If your control system is integrated with a Windows domain, it is recommended that you use the domain controller as the time source for all the clients within the domain (this is the default setting). As domain controllers are typically not on a network that is accessible to the control system itself, the controllers within the process control should be configured to get their time from an Experion server that has been set up as an NTP server acting as a secondary NTP server, which gets its time from the domain controller.

- **Time protocols and time servers for Experion PKS controllers**

The C300 uses PTP to attain the time, but includes its own proprietary ability to adjust the clock increment rate to that of the time source so that fewer actual adjustments (bumps) are necessary (similar to the,functionality of NTP).

When plant-wide correlation of C300 DI-SOE is required, PTP must be provided.

As a client, Safety Manager supports the NTP protocol but also supports the PTP time protocol, which is used for synchronization over its own SafeNet network.

- **Precision time protocol (PTP)**

The IEEE-1588 Precision Time Protocol provides high-precision time with low overhead. Unlike NTP, it is only a Local Area Network protocol, requiring a local time server, known as an IEEE-1588 Grandmaster. PTP supports multiple Grandmasters for availability. Typically, Grandmasters get their reference time from GPS, but can also get time from other GPS devices using one of the standard coaxial cable protocol connections. Multiple units from the same manufacturer may even share a GPS antenna. PTP is a UDP Multicast protocol, so it adds some network traffic.

Time source hierarchy

The Series C controllers and interfaces have a time source hierarchy. When the better time source becomes unavailable, they degrade to a less-accurate source. From highest to lowest precision:

• PTP (if enabled)

• CDA Server Protocol

Time source configuration

PTP is used when individual devices are configured in Control Builder to use it. Once enabled, PTP is self-configuring.

PTP Grandmaster configuration notes

PTP defines synchronization profiles for various applications. We use the default profile. The basic settings are synchronization every two seconds, and using multicast for round-trip-time determination.

12.3.5 Planning your time hierarchy

Implementing a time strategy that is appropriate for a process control system involves using a time hierarchy where the root of the hierarchy is the most reliable time source.

Windows domains, by default, implement a hierarchy of time distribution across the domain. The domain controller that is the PDC Emulator is the node that synchronizes with a reliable external source.

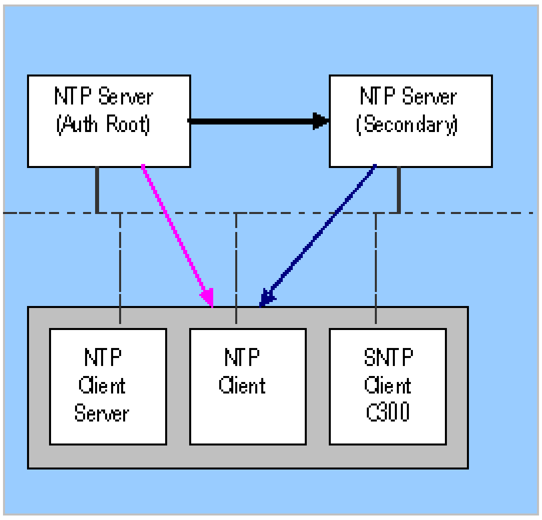
Ideally, all the control systems nodes are linked together in a hierarchy of time critical usage. In most cases, it may be better to set up one or two secondary network time protocol (NTP) servers that service all the control systems NTP clients on the local network. This provides mitigation against requirements on the network being available to keep the time consistent and attain accurate time.

Your organization may already have a time hierarchy in place that can be used as a source from which you can then branch off for the control system. Regardless of the source of time, all topologies should include local NTP servers that serve time in the control system.

Workgroup topology with no external source

In the following diagram, redundant Experion servers are set up as redundant NTP servers. All time clients, such as Flex Stations, Console Stations, and C300 controllers, receive time from the primary Experion server. If this server fails over, time is served by the backup Experion server.

This topology uses the internal CMOS clock of the authoritative server as the time source. This time source is not as accurate and will create deviations in time seen throughout the hierarchy. Systems with strict requirements on sequence of events should not implement this topology.

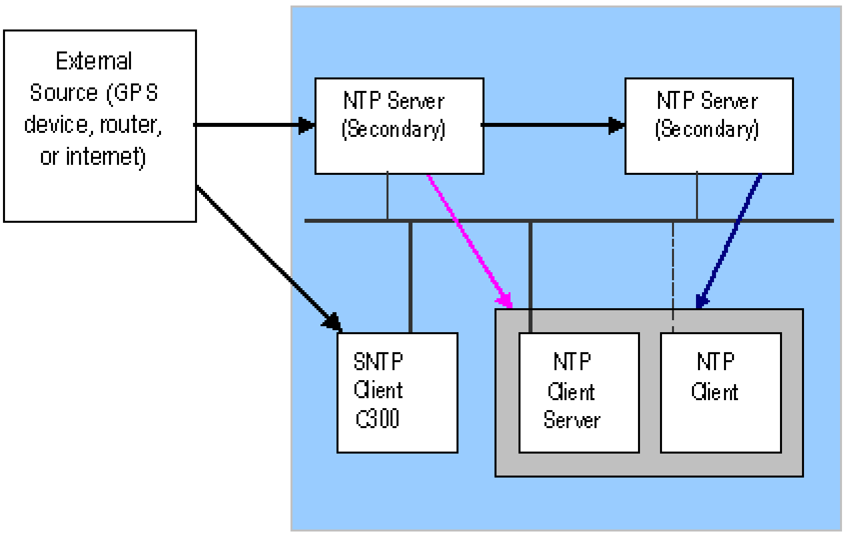


Workgroup topology with external source

In the following diagram, the primary Experion server receives time from an external source. The redundant Experion servers are set up as NTP servers which serve time to all time clients, such as Flex Stations, Console Stations, and Safety Manager. If the primary Experion server fails over, time is served by the backup Experion server. An external source serves time to the C300 controller.

This topology creates a dependency on the network infrastructure between the control system and the external source. The reliability of this network should be taken into consideration when planning this time hierarchy.

If the network between the external sources is reliable or local, then direct access to the external (or local) device is recommended. This means that you only need one secondary server. This is highly recommended for SOE configurations.

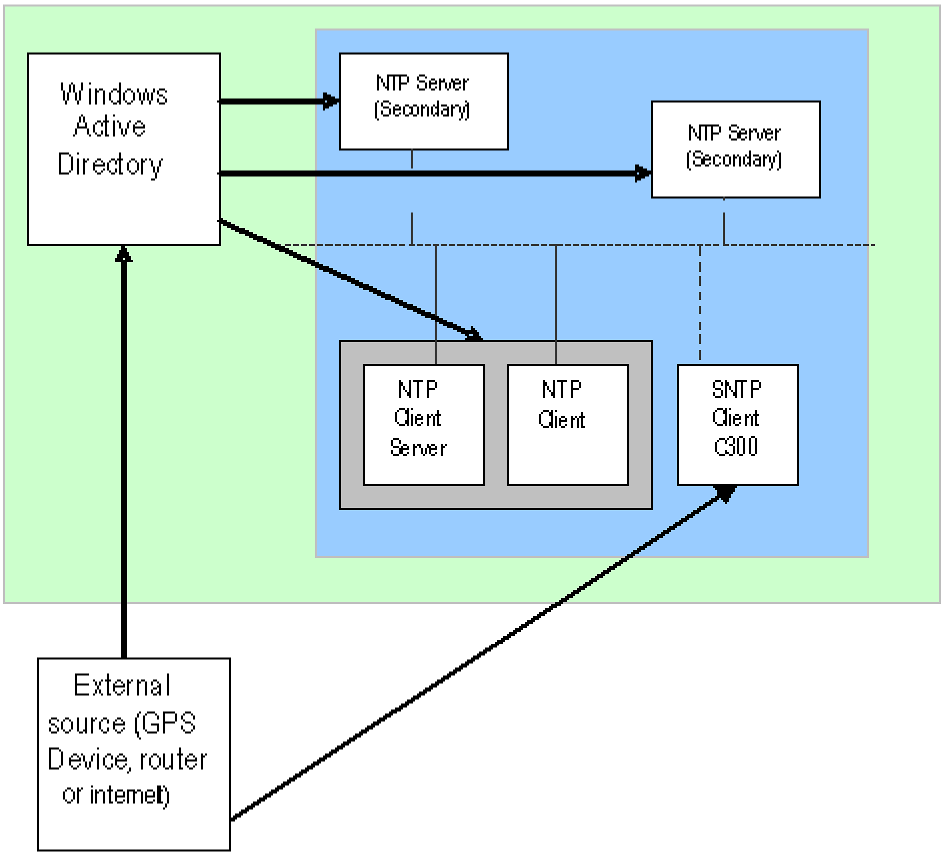


Domain topology

In the following diagram, the domain controller receives time from an external source. The domain controller serves time to the redundant Experion servers, Flex Stations, and Console Stations. An external source serves time to the C300 controller.

You should verify that the Active Directory/domain you link to uses a reliable external time source. If an external source is not used, the internal CMOS of the domain controller (or PDC Emulator) is used as the time source. This time source is not as accurate and will create deviations in time seen throughout the hierarchy.

Systems with strict requirements on sequence of events should not implement this topology.



12.3.6 Planning considerations for time synchronization

• Controllers and CEEs obtain the start and end of Daylight Saving Time (DST) from Experion servers. For controllers and CEEs to automatically change to DST, the Experion server's clock must be selected to automatically adjust clock for daylight savings changes. DST changeover in the Experion server's clock triggers the controllers and CEEs to change to DST, regardless of whether it's done automatically by the operating system or enabled manually by the user within the DST period.

• For controllers, CEEs, and Experion servers in the same time zone, whenever DST starts or ends, the controllers and CEEs clocks are adjusted either forward or backward by 1 hour.

• The start and end of DST is always identified from the Experion server. In the case where controllers, CEEs, and Experion servers are not in the same time zone; controllers and CEEs will shift by one hour (irrespective of any time zone the controller or CEE is in and also irrespective of if DST is applicable for controller or CEE time zone) at the start and end of DST (applicable as per server time zone).

• For controllers and CEEs that are in different time zones to the Experion servers, Honeywell recommends not to use Automatically Apply DST option and to manually change the 'Daylight Savings Time' on these controllers and CEEs.

For controllers and CEEs that are in the same time zone as the Experion servers, you can use the Automatically Apply DST option. Therefore, the 'Automatically Apply DST' option can be applied to specific controllers and CEEs within an Experion cluster.

12.4 Scan time

The processing time from program step 0 to the last step is called the Scan Time.

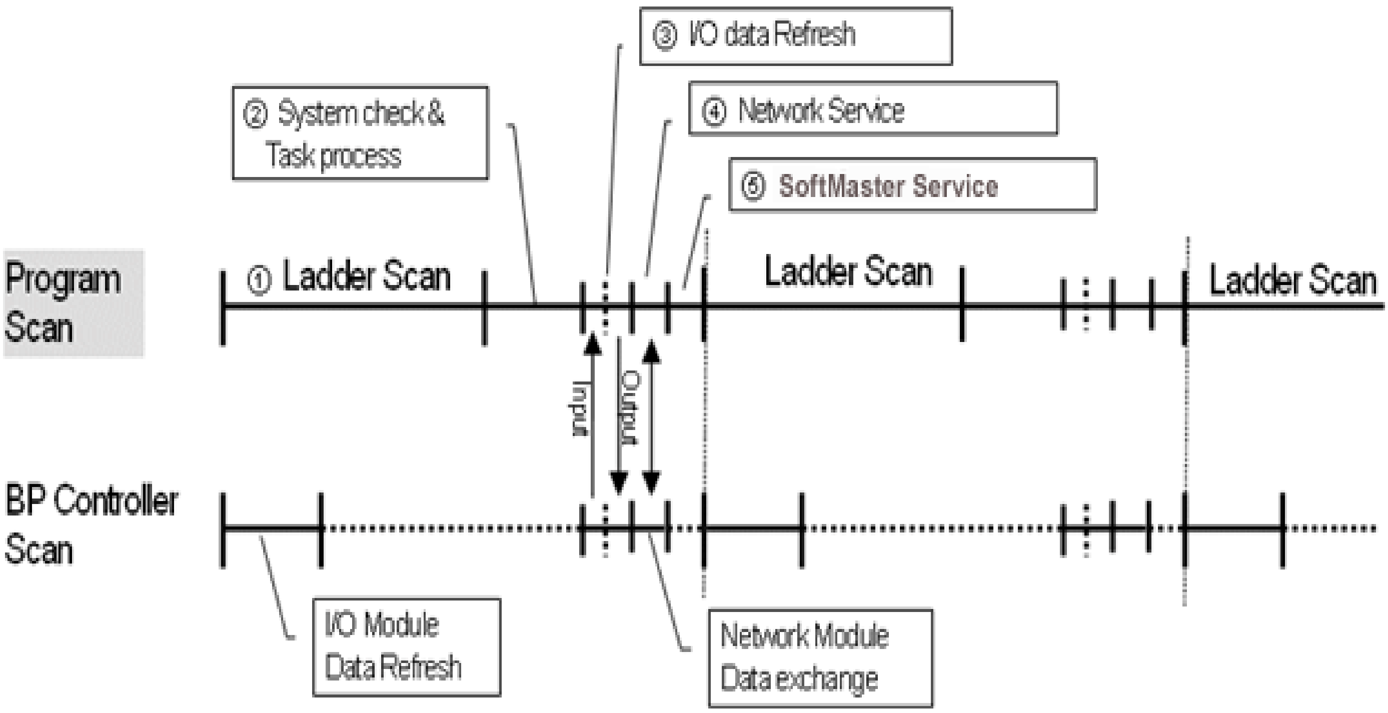
1. Operation and performance of ML-200R.and ML-200 IEC The program execution time, I/O data process time, and communication service time, and the synchronization time between the master and standby CPU are important factors affecting the scan time.

- Using the hardware relay method, the data exchange between expansion drive modules and the data communication performance of the CPU is improved.

- The ML CPU reduces scan time through improved data reception performance through ladder program execution and backplane, ladder program execution by MPU, and parallel execution of I/O data scan, and so on.

Type	MPU Processing Time		BP Controller Processing Time		
	Ladder Execution (32 KStep)	System Task	Digital I/O Module (32 points, 1module)	Analog Module (8 ch, 1module)	Communication Module (basic/extension) (200 byte, 1 block)
2MLR-CPUH	2.752 ms	1.0 ms	20ms	75 ms	170 + 44ms
2MLI-CPUU	0.896 ms	0.6 ms	20 ms	75 ms	185 ms

2. Calculation of Scan Time (Single CPU operation). The CPU module executes the scan program in the following sequence. You can estimate the scan time performance of a system using the following calculation.



- Scan time (µs) = Scan program process + system check and task process + I/O data refresh + network service + SoftMaster Service + user task program process + CPU

a) Scan program process = number of program steps created x 2 x 0.042(µs) *Precise scan time requires in addition of processing time for used instructions. *x 2 refers to the number of average execution code per instruction step.

b) System check and task process: 800 µs ~ 1.0ms [parameter depending on the usage of auxiliary functions].

c) SoftMaster service process time: 100 µs at the maximum data monitor. *10% of maximum scan time, there is a setting in the connection settings to improve the process time.

d) Task program process time: Sum of task processing time that occurs within a scan; the time calculation by task programs are the same as of scan program.

13. PKS Software Backup

13.1 PKS Software installation

Following software will be installed in the C partition, with the minimum size of 80 GB:

At	*System of operation*	*Supplier*	*Operation software*	*Supplier*
Operator Stations & Engineering Station	Windows10,Professional	Microsoft	PKS Operator Station HMI Web Display Builder Knowledge Builder Control builder Quick builder	(Honeywell)
PKS Server	Windows10,Professional	Microsoft	PKS Server Quick Builder HMI Web Display Builder PKS Operator Station Knowledge Builder, Etc. Modbus RTU interface	(Honeywell)

13.2 Database and System Files Backup

13.2.1 DBAdmin tool

- DBAdmintool:
  Initialize/Expand/Backup/Restore database ect
- EMDB : Configuration Studio->[System Name]->Administer the system Database
- ERDB : Configuration Studio->[System Name]->[Server Name]->Administer the Control Strategy Database

13.2.2 System Files Backup

The following system files will be backup :

- **PKS Server**

.. = C:\Program Files\Honeywell\Experion PKS

- ..\Engineering Tools\system\ER\epks_emdb.mdf ERDB
- ..\Engineering Tools\system\ER\ps_emdb.mdf EMDB
- ..\Engineering Tools\system\ER\VersionControl.mdf QVCS Database
- ..\Server\Data*.* : Experion server database ;
- ..\Server\Run*.exe Executable application ;
- ..\Server\Archive*.* : History archive files (default folder) ;
- ..\Server\User*.* : Custom files ;
- ..\server\user\applic\src Custom applications
- ..\Server\Data\Report*.* : Report output Files ;
- ..\Server\Evtarch\EMSEvents.mdf : Event Database ;
- ..\Server\Data\Qdb*.qdb : Quick Builder Database ;
- ..\Server\Data\OPCIntegrator\OPCIntegrator.mdf OPCIntegrator
- ..\Client\Abstract*.* : Custom displays ;
- ..\Client\Station*.Stn and etc. : Menu files/Startup scripts/Alarm wav files ;
- ..\Client\System\R501*.Stb : Menu and keyboard configurations ;
- ..\Client\System\R501\Sys[nnn].dsp/htm : System standard display ;
- C:\Windows\System32\Drivers\etc\hosts : hosts file

- **PKS Station**

.. = C:\Program Files\Honeywell\Experion PKS

- C:\Windows\Station.ini Station.ini
- ..\server\data*.* ES-C
- ..\server\user*.* Custom files
- ..\Client\Abstract*.* : Custom displays ;
- ..\Client\Station*.Stn and etc. : Menu files/Startup scripts/Alarm wav files ;
- ..\Client\System\R501*.Stb : Menu and keyboard configurations ;
- ..\Client\System\R501\Sys[nnn].dsp/htm : System standard display ;
- ..\Engineering Tools\system\bin bootpdata.txt

- **EMDB Files**

- .. = C:\Program Files\Honeywell\Experion PKS
- ..\Engineering Tools\System\Er\<Systemname>.Bak : EMDB backup manually

- **C300 Files**

- ..\Engineering Tools\System\Er\Ps_erdb.Bak : ERDB backup manually

- **Checkpoint Files**

- C:\Documents and Settings\All Users\Application
- Data\Honeywell\Experion\PKS\Checkpoint

14. Configuration STRUCTURE

The platform SIS modules hold the logic for performing the actions required in a Safety Instrumented System and can be downloaded only to Logic Solvers. Each Safety Logic Solver (SLS) can contain a maximum of four SIS modules.

Logic Unit/ Areas

In the SIS configuration so called “Areas” are used to create logical groups of control / SIS modules. Two other functions related to the Areas are;

- Per Area can be decided which operator (= user) has privilege for operation of that area.
- Per operator station can be decided which AREA(s) will alarm on that operator station.

The naming of the areas shall be meaningful, typically related to the associated equipment, i.e. named after the actual equipment/unit (e.g. R101, T202, etc.) or after the functional use of the equipment/unit (e.g. Storage, Blending, Separation, etc.).

More details of the area names and module assignments within those areas will be defined during the detail design phase of the project in the Detail Design documents.

Function Blocks and Parameters

DeltaV SIS has a set of standard function blocks (e.g. AND, OR, XOR etc.), timers (OND, OFFD, etc.) And a set of advanced function blocks. All DeltaV SIS function blocks are TÜV TP approved.

Math Palette Blocks		
	Calculation/Logic (LSCALC)	Allows you to specify an expression that determines the block's output. Mathematical functions, logical operators, constants and parameter references can be used in the expression.
	Comparator (LSCMP)	Compares two values and sets a Boolean output based on that comparison.
	Limit (LSLIM)	Limits an input value between two reference values. The block has options that set the output to a default value or the last value if the input becomes out of range.
	Middle Signal Select (LSMID)	Selects between multiple analog signals. The block selects the mid-valued input from the inputs that are are not disabled and do not have Bad status. If there is an even number of inputs, the average of the two middle valued inputs is used as the middle value.

Timer Palette Blocks		
	Off-delay Timer (LSOFFD)	Delays the transfer of a False (0) discrete input value to the output by a specified time period. The block supports signal status propagation.
	On-delay Timer (LSOND)	Delays the transfer of a True (1) discrete input value to the output by a specified time period. The block supports signal status propagation.
	Retentive Timer (LSRET)	Generates a True (1) discrete output after the input has been True for a specified time period. The elapsed time the input has been True and the output value are reset when the reset input is set True.
	Timed Pulse (LSTP)	Generates a True (1) discrete output for a specified time duration when the input makes a positive (False-to-True) transition. The output remains True even when the input returns to its initial discrete value and returns to its original False value only when the output is True longer than the specified time duration.

Icon	Function Block	Description
IO (Input/Output) Palette Blocks		
	Analog Input (LSAI)	Accesses a single analog measurement value and status from an I/O channel. The input value is a transmitter's 4 to 20 mA signal. The block supports linearization, signal scaling, signal filtering, signal status propagation, and simulation.
	Digital Valve Controller (LSDVC)	Drives a HART Two-state Output channel connected to a digital valve controller. The block supports partial stroke testing, fault state detection, and field device confirmation.
	Discrete Input (LSDI)	Accesses a single discrete measurement value and status from a two-state field device and makes the processed physical input available to other function blocks. The block supports signal inversion, signal filtering, signal status propagation, and simulation.
	Discrete Output (LSDO)	Takes a discrete input value representing the commanded output state and writes it to a specified Discrete Output channel. The block supports fault state detection and field device confirmation.

Logical Palette Blocks		
	Alarm (LSALM)	Performs alarm detection on a user-specified input. The parameters generated can then be used to generate alarm events at the user interface.
	Bi-directional Edge Trigger (LSBDE)	Generates a True (1) discrete pulse output when the discrete input makes a positive (False-to-True) or negative (True-to-False) transition since the last execution of the block. The block supports signal status propagation.
	Boolean Fan In (LSBFI)	Generates a discrete output based on the weighted binary sum, binary coded decimal (BCD) representation, or logical OR of one to sixteen discrete inputs. The block supports signal status propagation.
	Boolean Fan Out (LSBFO)	Decodes a binary weighted input to individual bits and generates a discrete output value for each bit (as many as sixteen outputs). The block supports signal status propagation.
	And (LSAND)	Generates a discrete output value based on the logical AND of two to sixteen discrete inputs. The block supports signal status propagation.
	Not And (LSNAND)	Generates a discrete output value based on the logical AND of two to sixteen discrete inputs, then performs a NOT on the result. The block supports signal status propagation.
	Not Or (LSNOR)	Generates a discrete output value based on the logical OR of two to sixteen discrete inputs, then performs a NOT on the result. The block supports signal status propagation.
	Not (LSNOT)	Logically inverts a discrete input signal and generates a discrete output value. The block supports signal status propagation.
	Or (LSOR)	Generates a discrete output value based on the logical OR of two to sixteen discrete inputs. The block supports signal status propagation.
	Not Exclusive OR (LSXNOR)	Performs a NOT on the exclusive OR of two inputs.
	Exclusive OR (LSXOR)	Performs an exclusive OR of two inputs to produce an output.
	Negative Edge Trigger (LSNDE)	Generates a True (1) discrete pulse output when the discrete input makes a negative (True-to-False) transition since the last execution of the block. The block supports signal status propagation.
	Positive Edge Trigger (LSPDE)	Generates a True (1) discrete pulse output when the discrete input makes a positive (False-to-True) transition since the last execution of the block. The block supports signal status propagation.
S R	Reset/Set Flip-flop (LSRS)	Generates a discrete output value based on NOR logic of reset and set inputs.
S R	Set/Reset Flip-flop (LSSR)	Generates a discrete output value based on NAND logic of set and reset inputs.

Advanced Functions Palette Blocks		
	Analog Voter (LSAVTR)	Monitors a number of input values and determines if there are enough votes to trip. If a configured number of the inputs vote to trip, the block trips and sets the output of the block to 0 (zero).
	Cause & Effect Matrix (LSCEM)	Defines interlock and permissive logic that associates as many as 16 inputs and 16 outputs. Configure one or more inputs to trip each output. When an input becomes active, all outputs associated with that input trip.
	Discrete Voter (LSDVTR)	Monitors a number of input values and determines if there are enough votes to trip. If a configured number of the inputs vote to trip, the block trips and sets the output of the block to 0 (zero).
	State Transition Diagram (LSSTD)	Implements a user-defined state machine. The state machine describes the possible states, and the transitions between those states, that can occur.
	Step Sequencer (LSSEQ)	Associates system states with actions to drive outputs based on the current state.

15. Performance

15.1. SIS Module Cycle Time

The default cycle time for SIS Module execution in DeltaV SIS Logic Solver is 50ms considering standard logic. The SLS cycle time is directly related to the input to output response time of the system. Other possible cycle times are 100ms, 150ms & 200ms.

15.2. Logic Solve response Time

The response time for a Safety Instrumented Function (SIF) must be less than the process safety time. The SIF has a response time associated with the sensor, logic solver, and final element subsystems. The response time of the logic solver subsystem is the time between any change on a SIF input channel that should result in a trip and the time that the output channel or channels change to the tripped state. The time is measured from screw terminal to screw terminal. The Scan Rate of Logic Solver is 50ms.

16. Naming Convention

Tags and names in DeltaV SIS are limited to a maximum of 16 characters. For better visibility to the operator the names are kept as small as possible by optimizing the usage of "_" and "-"s. In some cases, Emerson recommends limiting the tags to 12 characters so that a suffix may be added for alarm identification.

17. Node Naming

The DeltaV controller that connects the DeltaV SIS SLS to the DeltaV control network is a called a

Node. The naming of the SIS node is normally a logical name that includes a reference to its location or application. The following table lists the node names for this project. The naming of the areas shall be meaningful, typically related to its location (e.g. SIS-CCR, SISTANKS, etc.) or application (e.g. SIS-FG, SIS-ESD, etc.). More details of the node names and SLS assignments within those nodes will be defined during the detail design phase of the project in the Detail Design documents.
The below arrangement will be used for the AWP EPC3 - CCTMU incl. BPS, CAL, AGMU, Test line, HPPS and Measuring System on Altosonic-5 Flowmeter.

ESx_IRyy_zzz
FGx_IRyy_zzz

ES: Emergency Shutdown System

FG: Fire and Gas System

X: Domain Number

YY: Instrument and Technical Room Number

ZZZ: Controller Number

17.1. SLS Naming

The SLS name is normally a logical extension of the node name as follows (max. 16 characters):

Czz_Snn
C: Controller

ZZ: Controller Number

nn: Card Number

S: Safety System

- Example: C04_S22

Note: Redundant logic solvers occupy 4 slots; hence this number is always an odd nr. e.g. 01, 05, 09M

17.2. SIS Module Naming

The SIS module name shall start with the logical group name, typically in line with SIF name identified or another logical name. If multiple modules are required for the same group of logic, a numeric suffix will be added to the name. More details of the module names will be defined during the detail design phase of the project in the Detail Design documents.

17.2.1. Alarm Naming and Priorities

17.2.2. Alarm Names

Alarm names shall be based on the client assigned tag number and the function of the individual alarms. Alarm names are limited to a maximum 16 characters. Alarms will be configured as below.

❖ **Analogue Input Alarm**

- Each LSAI function block will have a PVBAD alarm

- Each LSAI will be connected to an LSAVTR function block. Each LSAVTR will have a confirmed trip alarm. The number of alarms will depend upon the number of inputs to the AVTR function block (i.e. an AVTR with 3 Inputs will have four alarms – one for each input and one confirmed trip Alarm). Refer to the Analogue section for detail of AVTR function block.

- Each LSAVTR function block which has a bypass requirement will have a bypass enabled alarm. This alarm shall be purely used for data entry into the event history (the bypass was applied by the operator so there is no need to alarm him). As such this alarm shall be configured as a LOG alarm.

❖ **Digital Input Alarm**

- Each DI will be connected to a discrete voter (LSDVTR) function block. Each LSDVTR will have a confirmed trip alarm. The number of alarms depends upon the number of inputs to the LSDVTR function block (i.e. a LSDVTR with 3 Input will have four alarms – one for each input and one confirmed trip alarm). Refer to the discrete section for detail of LSDVTR function block.

- Each LSDVTR function block which has a bypass requirement will have a bypass enabled alarm. This alarm shall be purely used for data entry into the event history (the bypass was applied by the operator so there is no need to alarm him). As such this alarm shall be configured as a LOG alarm.

17.2.3. Alarm Priorities

Alarm priorities indicate to the operator the importance of an alarm. The priority affects the order in which alarms appear in the alarm banner and the alarm list in the operator interface.

All alarms that are configured with the same alarm priority are displayed the same way throughout the system. If the definition of a particular alarm priority is modified, all alarms configured using that alarm priority will use the new definition.

There are 12 possible alarm priority levels: numeric values 4 through 15 plus a special "log only" priority level (value 3). The highest priority value is 15 (it is used for the most important alarms). The lowest priority value is 4.

Events with Log priority (level value 3) are not considered alarms and are not displayed in the alarm banner or the alarm list. The Log priority can be used to designate an event that is important enough to be recorded in the event chronicle, but not something the operator needs to be aware of.

The alarm priorities to be utilized for alarms on the project are summarized in below Table:

Priority Value	Alarm Priority	Auto Acknowledge	Auto Acknowledge when Inactive	Alarm banner show
15	Critical	NO	NO	Not Hidden
11	Warning	NO	NO	Not Hidden
7	Advisory	YES	NO	Not Hidden
4	Active Bypass	YES	NO	Hidden
3	Log	YES	NO	N/A

17.3. Signal Handling

17.3.1. Fail Safe Logic – de-energize to Trip

The configuration of the SIF functions will be based on De-Energize to Trip operation. A logical “1” indicates healthy or no trip action and a logical “0” indicates abnormal or trip action. Field devices are assumed to be configured for fail safe action accordingly (by others). Input signals are assumed to be failsafe e.g. contact normally closed and open to trip. Output signals are assumed to be Fail safe e.g. de-energized to trip and valves will enter the safe position if energy or instrument air is lost (fail open / fail close).

17.3.2. Energized to Trip Logic

In certain applications Energize to Trip logic is required. If input signals are not fail safe “In the field” the signal requires inversion in the Input function block before using it in the logic. If Non - fail safe outputs are used (Energies to Trip) a “Not” function block will be added in the wire to the DO function block “CAS_IN_D” parameter.

17.3.3. Line Fault Detection Requirements

- **Discrete Inputs**

For DI signals line fault detection requires that resistors are installed in the field or that NAMUR proximity switches (for inputs) are used.

- **Discrete Output**

All the discrete output channels (de-energized to trip) will have their LINEFAULT_DETECT parameter set to TRUE/FALSE. This is the default setting recommended by the DeltaV SIS

User's Guide and enables periodic pulse testing of the outputs.

- **Analog signal conversion**

All analogue inputs will be linearly converted from percent to the specified engineering range and Units.

- **Input filter time**

By default, the filter time value will be set to zero. As this is a configurable parameter, it can be changed as required during commissioning.

- **Over-range/under-range detection**

All configured analogue input channels will be configured with an under range and over range detection functions. As soon as the AI channel under range / over range is detected, the AI Channel is marked as "BAD" which results in a system diagnostics alarm.

By default, the under-range and over-range parameter values are set at - 3% and +103% of range respectively.

17.4. HART Communication

The HART communication option is enabled for AI channels as per information available in IO list.

17.4.1. Historical Data Collection

The DCS be capable of displaying process parameters in time-based trends. The time base can be operator selectable from a set presented on the Operator double Workstation.

Up to 8 variables be selectable for trending on a selectable time base on a single display panel. The variables be capable of being panned across the time axis. The traces be displayed in different colors.

Every analogue value in the system be capable of being trended, including process variables, set points, control outputs, internal calculated values and variables captured from sub-systems. Historical data storage include data from ESD, F&G and other connected sub-systems. It include events, measurements or calculations.

The DCS can provide short term tuning trends (with scan resolution of 1 second) to enable the operator/engineer to tune the closed loop controls by adjusting the settings on the control loop. The tuning trend display time base be operator selectable in order to assist in tuning loops with different time responses.

Longer term historically stored trends be configurable in the system for online storage for a minimum of 1 year. The trend display time bases be operator selectable for display purposes, with the minimum of 1 minute for each sample to be displayed.

The system be capable of transferring the data to removable data storage media (DVD, CDROM) for archiving purposes and retrieval at a later stage. It be possible to enable or disable archiving of individual trends to reduce data storage requirements. The vertical axis of both the tuning and historical trends be capable of being selectively magnified and compressed, by a factor of 10 in order to provide a flexible viewing window for the users, for the entire recording period. This apply to real time trends as well as archived data.

The historical storage devices and network interfaces be redundant.

All operator double workstations be capable of accessing the common trending system, using a common tag numbering system. The DCS be capable of printing trends and screen dumps on a dedicated color printer.

17.4.2. Trending

Real time trend of the above two data points can be accessed from the operator Faceplate and appear as a 'pop up' window over the current display. Trends based on the historical data collected for the above data points can be similarly accessed via the operator interface displays and viewed on the workstation.

17.5. Operational functions

17.5.1. Maintenance override (MOS)

MOS's are implemented in the module logic for all the individual inputs except on Flame and axial displacement type sensors for security reasons.

A common MOS permit key-switch is fitted on the ESD / FGS Auxiliary Consoles. Individual MOS's are activated / removed from the maintenance workstation in relevant ITR in the following manner:

- **MOS activation**

At any given moment there can only be one MOS active per interlock.

If no other MOS is active on the unit and when the MOS permit key-switch is activated a timer, the duration of which will be initially configured as 60 seconds, is started. During this MOS permitted time window an operator (with sufficient privilege) can then activate a MOS from the operator interface via a check box in the associated voter block faceplate.

Once the MOS permit timer has expired no further maintenance overrides can be applied until the maintenance bypass enable key switch is re-activated.

The activation of a MOS is logged in the event history using a LOG alarm.

- **MOS removal**

An active MOS can be removed individually at any time by an operator, logged into the

engineering workstation with proper privilege, through the associated voter block faceplate.
It is not necessary to activate the MOS enable key switch in order to remove an active maintenance bypass.
At no circumstance will a MOS be removed automatically by the system.

- **MOS reminder alarm**

A MOS reminder alarm will be provided for each individual MOS. If the input remains MOS' ed for longer than the configured time then a reminder alarm is generated. These reminder alarms can reoccur at the defined time intervals so long as the input remains bypassed. The reminder time period will initially be set to 12 hours = 43200 seconds.

17.5.2. Start-up Bypass (Overrides)

Start-up bypasses will be provided as per customer definitions.
These will be configured, utilizing functionality built into the voter blocks, as timed start-up bypasses.
These timed start-up bypasses will be automatically initiated if the cause concerned is in the tripped condition and the corresponding reset is activated. Whilst the start-up timer is active the output of the voter function block is forced to its normal state value allowing the corresponding outputs to be reset.
Once the start-up time has elapsed if the input has not become healthy in the intervening period a trip will be initiated. The duration of these start-up timers (voter function block STARTUP_TIME parameter) will be configured as 30 seconds, as an initial project default setting. Actual required start-up times to be provided by customer for detailed design.
A start-up bypass will also be removed before the time elapses if the associated process input reaches its normal condition and remains stable for a period of 5 seconds. The value of this voter function block STABLE_TIME parameter can be changed either during configuration or subsequently as a runtime writeable parameter.

17.6. Reset Functions

An operator reset will be required before a tripped function can return to normal after a trip has occurred. Reset buttons will be provided on the operator graphic(s) in order to facilitate this function.
Some effects require individual resets whilst others are reset in groups as defined in the client supplied specifications.
Individual reset buttons will be located immediately below the effect dynamo concerned. The group reset buttons will be located below the legends on the graphic screen.
The reset button(s) will only be visible when the associated outputs (effects) are tripped and all associated causes other than those with defined start-up overrides have returned to their normal condition, i.e. a reset action is necessary

17.7. Fault detection and logic propagation

The DeltaV SIS is capable of detecting signal faults that indicate a possible problem with the instrument, wiring or input channel reading the signal.
SIS function blocks have a predetermined way of propagating the status of input parameters to output parameters. Faults detected on input channels cause Bad status to reach output function blocks

in SIS modules depending on the configuration of other function blocks in the SIS module.
The analogue voter (LSAVTR) and discrete voter (LSDVTR) blocks propagate Bad status on input parameters selectively. For example, if a single input of a 1oo2 or 2oo3 voter block has Bad status, its output [OUT_D] continues to have Good status because there are enough good inputs for a real process demand to cause a trip. However, if a single input of a 1oo1 or 2oo2 voter block has Bad status; its output [OUT_D] has Bad status. If a Cause input of a Cause Effect Matrix (LSCEM) block has Bad status, all Effect outputs associated with that input have Bad status.
This ensures that if Bad status is capable of preventing a process demand from causing a trip, Bad status propagates to the output function block(s).
It is then a configurable option at the output logic function block whether or not to trip the output channels a consequence of this Bad status. If configured to trip there is an additional parameter FSTATE_TIME in the output block which determines how long the status can be BAD before the output block initiates a trip. The default value is 300 seconds.
The AWP EPC3 - CCTMU incl. BPS, CAL, AGMU, Test line, HPPS and Measuring System on Altosonic-5 Flowmeter DeltaV SIS is configured in such a way that the fault detection on an input signal propagated through the logic to an output module causes an immediate trip.

18. I/O channel default parameters

18.1. Analog Input Channel

Below Table shows the configurable parameters of the AI channels and their project defaults.

Parameter	Default value	Project value	Remarks
NAMUR_ENA	TRUE	FALSE	Controls whether NAMUR alarming is performed on the channel. If enabled and if the transmitter supports it, an analog value that is outside the NAMUR limits (106.25% and -2.5%) for four seconds has its status marked as Bad: Sensor Failure.
OVERRANGE_PCT	103	103	The field value above which the Channel declares an Over-range
UNDERRANGE_PCT	-3	-3	The field value below which the Channel declares an under-range

18.2. Digital Input Channel

Below Table shows the configurable parameters for a DI channel and their project defaults.

Parameter	Default Value	Remarks
LINEFAULT_DETECT	FALSE	

18.3. Digital Output Channel

Below Table shows the configurable parameters for a Digital Output channel and their project defaults.

Parameter	Default Value	Project Value	Remarks
LINEFAULT_DETECT	FALSE	TRUE	

19. Logic implementation

DeltaV SIS software configuration is based on function block programming. All function blocks, which may be used in SIS modules, are approved by TUV.

19.1. I/O function blocks

Different I/O types use different I/O function blocks. Depending on the I/O type, the I/O function block parameters allow I/O specific parameter configuration.

19.2. Analogue voter

The AVTR function block provides an analogue voter function for use within safety instrumented functions. A voter block monitors a number of input values and determines if there are enough votes to trip. The AVTR block can monitor as many as 16 analogue inputs. If the configured number of inputs votes to trip (MooN), the block trips and sets its output to 0 (zero).
This function block provides additional functionality including different type of bypasses, trip delays, base stabilizing delays etc. In order to utilize the additional functionalities provided by the AVTR function block, all analogue input signals will be passed through an AVTR block. (When used in conjunction with a single signal the AVTR block will be configured for 1oo1 voting).

19.3. Discrete voter

The DVTR function block provides a discrete voter function for use within safety instrumented functions. A DVTR block monitors a number of input values and determines if there are enough votes to trip. The DVTR block can monitor as many as 16 discrete inputs. If the configured number of inputs votes to trip (MooN), the block trips and sets its output to 0 (zero). This function block provides additional functionality of different type of bypasses, trip delays, base stabilizing delays etc.
In order to utilize the additional functionalities provided by the DVTR function block, all discrete input signals will be passed through a DVTR block. (When used in conjunction with a signal the DVTR block will be configured for 1oo1 voting).

19.3.1. State Transition Diagram (LSSTD) function block

The LSSTD function block implements a state machine in the Logic Solver that transitions the block to a new state based on the current state and the currently active transitions (inputs). The block's state can also be set from another source. This function block supports as many as 16 transitions and 16 states.

19.3.2. Sequence (LSSEQ) function block

The LSSEQ function block defines as many as 16 states. In each state, the block sets the value of as many as 16 discrete outputs. The block can step through the states in sequence using internal increment and decrement parameters, or the block can be set to specific states (and the corresponding outputs set) from logic external to the block.

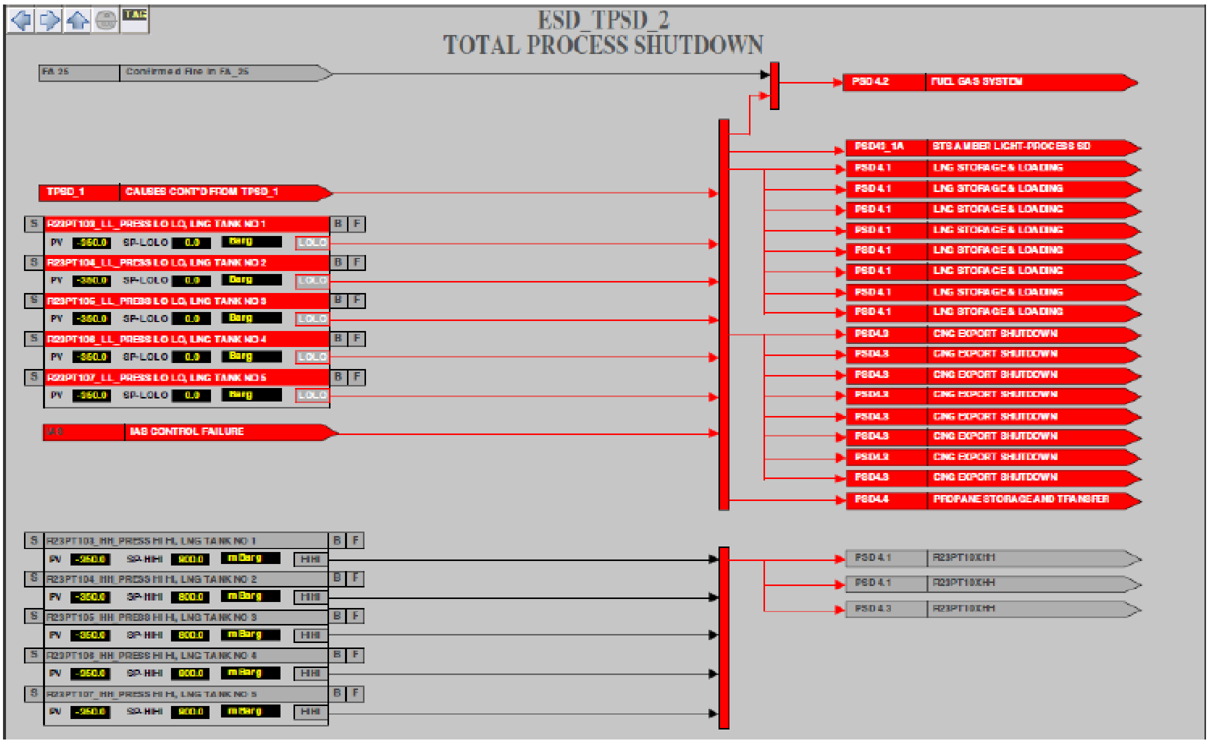
20. Control Modules

Input bypassed reminder Control modules are configured within SIS area for providing RESET, Alarm Auto Acknowledge an alarms functionality. Below table show list of control Modules:

Sr.	Control Modules	Description	Remarks
	ESD_ACT_BYP1	Control Module for ESD BYPASS Reminder alarm	
	ESD_RESET	Control Module for ESD P0/P1/P4/MEOH/THI Resets	
	FGS_ACT_BYP1	Control Module for FGS BYPASS Reminder alarm	
	FGS_RESET	Control Module for FGS BYPASS Reminder alarm	
	ALARM-ACK	Control Module for Alarm Acknowledgment in Remote	

21. Displays

A typical project specific C&E display is shown below. The causes are shown on the left side with the bypass status and the effects are shown on the right side with the description.



The dynamic elements on the Cause and Effect display will utilize the Emerson standard dynamos which are modified to suit need.

21.1. PCSD SIS analogue link dynamo

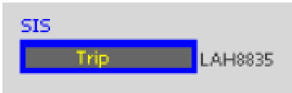
Below Table, shows the dynamos which will be used for analogue inputs shown on the process displays:

Dynamo	State		Description
PCSDSIS Analogue Link Dynamo	Normal	SIS -10.0 kg/hr FT8814HH	The process variable (PV) value and associated engineering units are shown in bright yellow on a small coloured rectangle. The colour of the rectangle indicates the PV status; dark grey = normal; dark red = PVBad. The module name is aligned to the right of dynamo; text black in normal state, magenta when simulated. The SIS text indicates an SIS dynamo. This is used to differentiate it from the DeltaV BPCS dynamo.
	PV Bad and simulated	SIS 25.3 kg/hr FT8834LL	
	Tripped	SIS -10.0 kg/hr FT8834LL	Active alarm in function block depicted as coloured surround. Whether or not this coloured surround flashes and the alarm requires acknowledgement is dependent upon the configured alarm priority Blue = Event [PCSD - LOG]. Yellow = Low [PCSD - Advisory] Orange = High [PCSD - Warning] Red = Emergency [PCSD - Critical]
	Bypassed	SIS B -10.0 kg/hr FT8814HH	Maintenance bypass active shown with the letter 'B' displayed to the left of the analog value. A start-up override active is similarly shown with the letters 'SO' displayed to the left of the analogue value.

When hovering over the dynamo with the mouse a bounding box and a tool-tip comprising the module/function block name appears. Clicking within this bounding box opens the associated AVTR function block faceplate.

21.2. PCSD SIS discrete voter (DVTR) dynamo

Below Table, shows the dynamos which will be used for discrete inputs shown on the process displays:

Dynamo	State		Description
PCSDSIS Discrete Link Dynamo	Normal		The process variable (PV) status either 'Normal' or 'Trip' is shown in Bright Yellow text on a small coloured rectangle. The module name is aligned to the right of dynamo; text black in normal state, magenta when simulated.
	Trip		Active alarm in function block depicted as coloured surround. Whether or not this coloured surround flashes and the alarm requires acknowledgement is dependent upon the configured alarm priority  Blue = Event [PCSD - LOG]. Yellow = Low [PCSD - Advisory] Orange = High [PCSD - Warning] Red = Emergency [PCSD - Critical]
	Bypassed		Maintenance bypass active shown with the letter 'B' displayed to the left of the analog value. A start-up override active is similarly shown with the letters 'SO' displayed to the left of the analogue value.

When hovering over the dynamo with the mouse a bounding box and a tool-tip comprising the module/function block name appears. Clicking within this bounding box opens the associated DVTR function block faceplate.

AVTR and DVTR dynamos on process graphics are similar to those used on Cause and Effect graphics except they are different for below things:

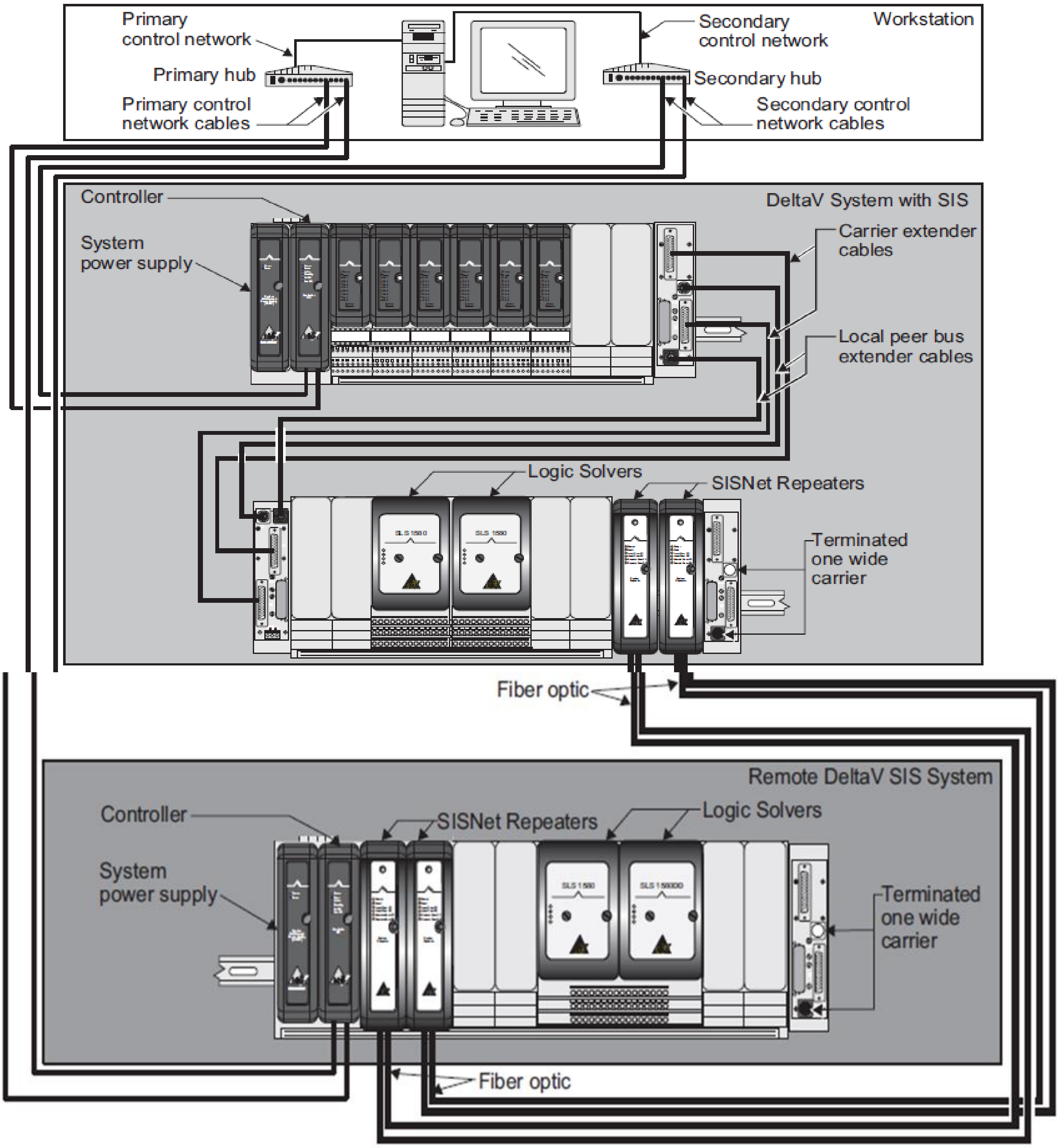
- SIS text appearing on each dynamo is animated such that SIS text will be RED if particular
- voter is Trip and it is Grey 32 if particular voter is Healthy.
- Clicking on SIS text will replace process graphics by respective Cause-Effect graphic.

22. Safety Network

22.1. Control Network

The DeltaV Control Safety Network provides communication between the nodes in the SIS network. Logic Solvers communicate with other Logic Solvers and with local SIS Net Repeaters through the carriers over a 2-channel local peer bus. The same message is broadcast over both channels. The local peer bus must be terminated at both ends. The local peer bus is terminated at the left end through the 2-wide power/controller carrier and at the right end through a terminated one wide carrier. SISNet Repeaters hosted by one DeltaV controller communicate with SIS Net Repeaters hosted by a different DeltaV controller over a fiber-optic remote peer ring.

A local SIS Net Repeater collects locally generated global messages into a single message and sends it to the next SIS Net Repeater in the ring. Upon receipt of a message, the receiving SISNet Repeater broadcasts it to its local peer bus and forwards the message to the next SIS Net Repeater in the ring. A global message is forwarded around the ring once. The primary SIS Net Repeaters form one fiber-optic ring and the secondary form a separate, independent ring. SISNet Distance Extenders that convert multimode fiber-optic signals to single mode fiber-optic signals can be used to extend the remote peer ring. Necessary data, like trip commands can be transferred in safety Network for using in other Logic solvers. Please see section 11.2 in this manual.



22.2. DeltaV SIS Nodes

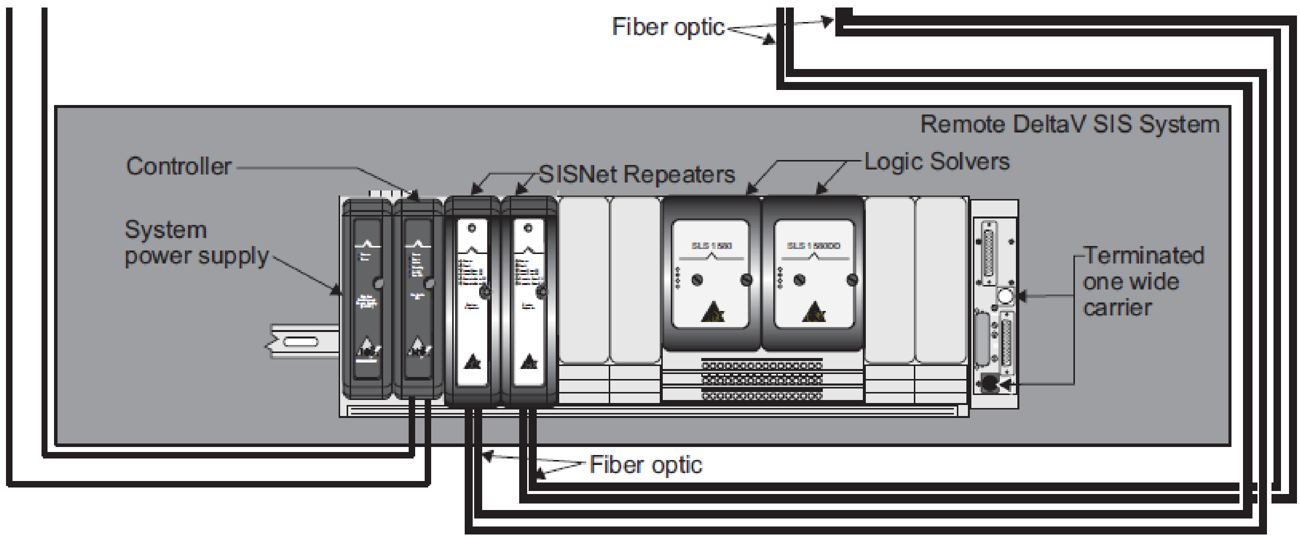
A DeltaV SIS node is a modular logic solving unit. Logic Solvers communicate with each other through

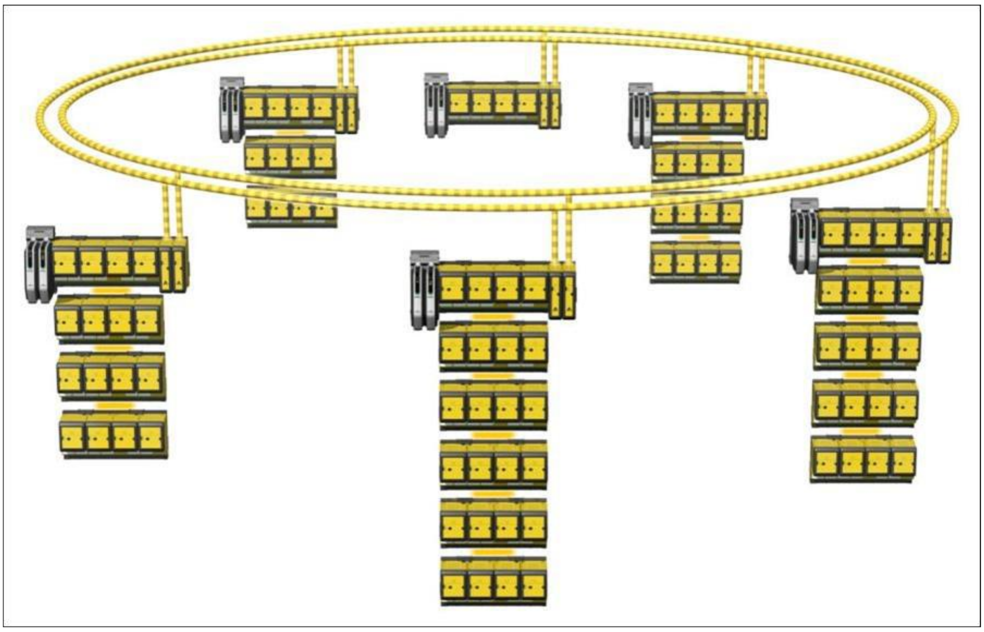
the eight-wide carriers over a two-channel, local peer-to-peer bus. The SIS secure communications are physically separate from the DeltaV control network and I/O bus, providing the complete physical separation and independence of control and safety that is required by IEC 61511.

All input data from any logic solver is accessible to all logic solvers on a node, with no effect on response time or safety integrity. This means that inputs can be wired into separate logic solvers and the input data is accessible to all of the logic solvers on the node. Complex logic can be executed in a single module, located on a separate logic solver from where the inputs are wired. This provides the capability to implement larger cause and effects matrices that combine several SIFs together in a single module.

With this kind of capability, the decision for where inputs are wired can be independent of the decision for where logic solving is performed. Combine this flexibility with the concept of configurable I/O and it is clear that DeltaV SIS provides significant flexibility to distribute logic solving across a node, according to your needs.

Multiple DeltaV SIS nodes can be connected together via the SIS Net secure communications network. The SIS Net is a redundant counter-rotating fiber optic ring network that provides SIL 3 certified communications between SIS nodes with very high communications reliability.





22.2.1. SIS Net Repeater

The SIS Net Repeater provides a link for DeltaV SIS nodes to communicate. This network is dedicated to safety information, carrying only safety signals. The network and SIS Net Repeaters are immune to any failure of the basic process control system network.

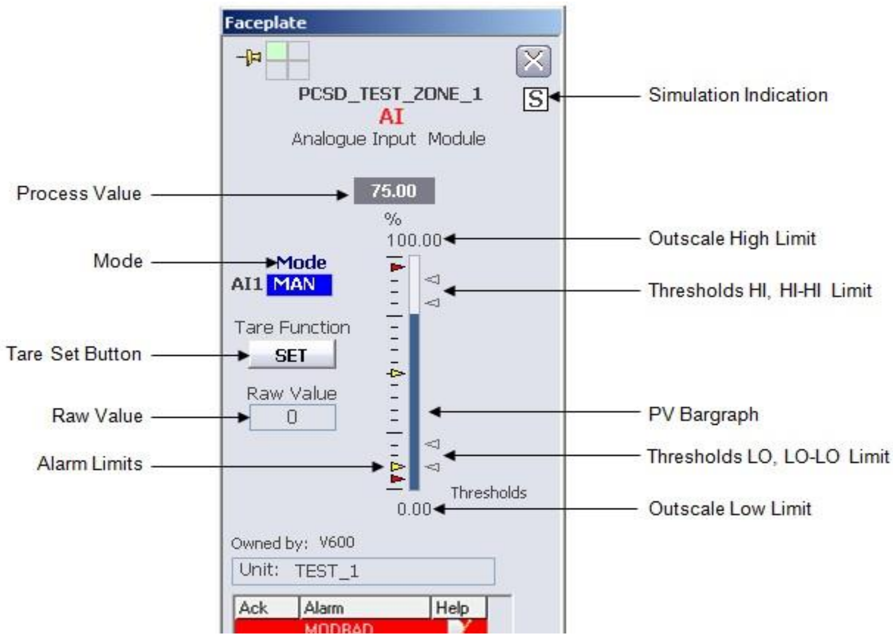
The SIS Net Repeaters are installed as a redundant pair. The primary SIS Net Repeater is connected to one fiber optic ring, and the secondary SIS Net Repeaters form a separate, independent ring.

22.2.2. Operator interface

Graphic elements associated with analog input modules:

❖ Faceplate

Faceplate for Analog Input Family (PCSD_AI_STD_130_FP)



Faceplate for AI Family except _AI_SCADA_130, _AI_SLMG_130, _AI_SOFT_130 and _AI_SOFTMG_130 modules

The faceplate provides following functionalities to the operator:

- **Process Value:** This value field provides indication of the current process value (PV). Background rectangle color of this value field changes to maroon color and orange color when status of PV changes.
- **Mode:** It becomes visible when actual mode of AIx block is other than AUTO. Bright blue background in actual mode field indicates Actual mode is different than Normal mode. (Not applicable for _AI_SCADA_130 and _AI_SLMG_130 modules).
- **Tare Set Button:** This button when clicked, sets the raw value as raw offset value. The Raw offset value is displayed on the detail display. (only applicable for _AI_TARE_130)

- **Raw Value:** Indicates the gross value. (only applicable for _AI_TARE_130)

- **Alarm Limits:** Indicates alarm limits for HI,HI-HI and LO,LO-LO alarms when enabled. Color of the pointers reflect the alarm priority.

- **Simulation Indication:** The simulation 'S' symbol is visible when the process value is simulated.
- **Out scale High Limit:** The higher limit of the PV scale.

- **Threshold HI, HI-HI Limit:** Indicates the HI, HI-HI limits for Thresholds. Color of HI and HI-HI limit indicator changes to grey and black respectively when PV crosses respective limits.

- **PV Bar graph:** A bar graph provides graphical representation of the process value (PV). The upper and lower limits of the PV scale and the engineering units are also indicated.

- **Thresholds LO, LO-LO Limit:** Indicates the LO, LO-LO limit for Thresholds. Color of LO and LO-LO limit indicator changes to grey and black respectively when PV crosses respective limits.

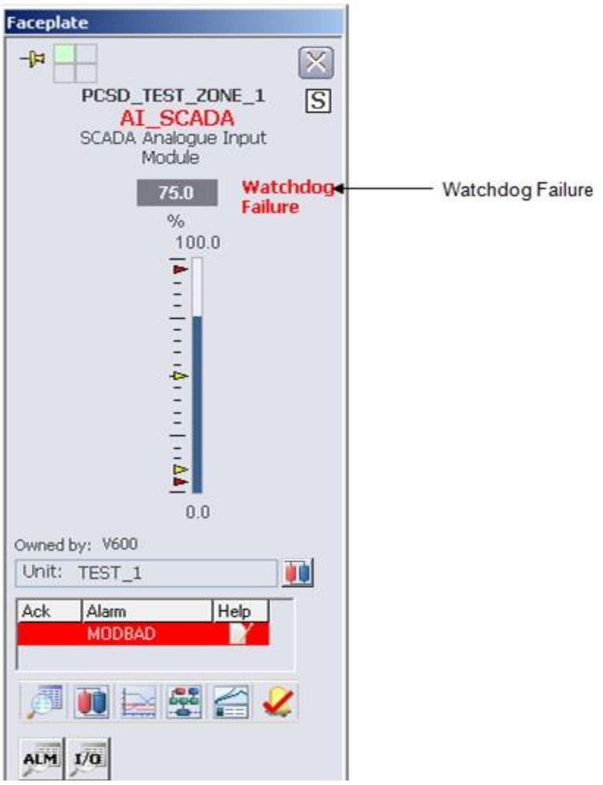
- **Out scale Low Limit:** The lower limit of the PV scale

22.2.3. Detail Display Call up buttons

- ALM (Alarms/Limits) : PCSD_MF_AI_ALM_130_DT
- I/O (Input/Output) : PCSD_AI_IO_STD_130_DT
- TUN (Tuning) : PCSD_AI_TUN_STD_130_DT
- THR (Thresholds) : PCSD_MF_THR_130_DT
- Info (PV Information) :

  - ✓ PCSD_InDiff : Applicable for _AI_DF_130 control module.
  - ✓ PCSD_InSel : Applicable for _AI_SL_130, _AI_SL3_130 and **_AI_SL4_130 control modules.**
  - ✓ PCSD_InSel_MUX : Applicable for _AI_MUX_F_130 control **module.**
  - ✓ PCSD_PTCOMP : Applicable for _AI_PTC_130 and _AI_MG_130 control modules.

Faceplate for AI SCADA Module (PCSD_AI_SCADA_130_FP)



The faceplate provides following additional functionalities to the operator:

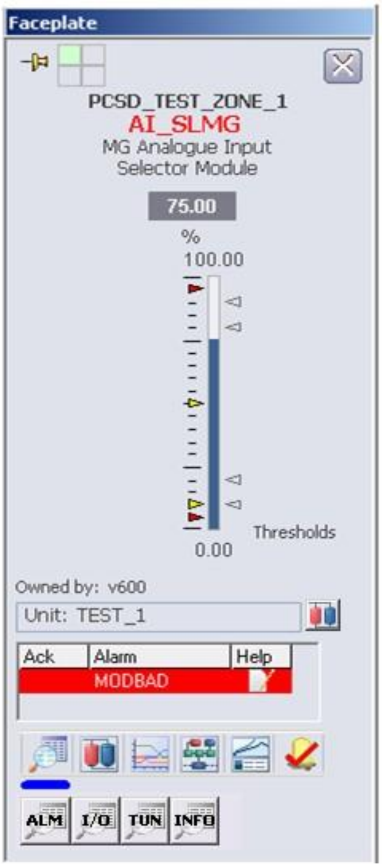
Faceplate for _AI_SCADA_130 module

- **Watchdog failure:** This text is displayed when the status of the link between DeltaV and third party devices is not healthy. Watchdog failure is generated for SCADA Analog control module (_AI_SCADA_130).

22.2.4. Detail Display Call up buttons

- **ALM (Alarms/Limits) :** PCSD_MF_AI_ALM_130_DT
- **I/O (Input/Output) :** PCSD_AI_IO_SER_130_DT

Faceplate for AI Selector Migration Module (PCSD_AI_SLMG_130_FP)

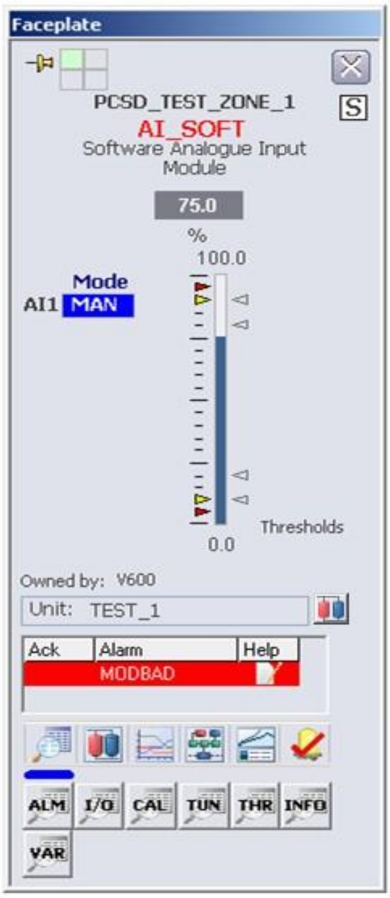


Faceplate for _AI_SLMG_130 module

22.2.5. Detail Display Call up buttons

- **ALM (Alarms/Limits) :** PCSD_MF_ALM_130_DT
- **I/O (Input/Output) :** PCSD_AI_IO_SLMG_130_DT
- **Info (PV Information):** PCSD _Insel _MG

Faceplate for AI Selector Migration Module (PCSD_AI_SLMG_130_FP)



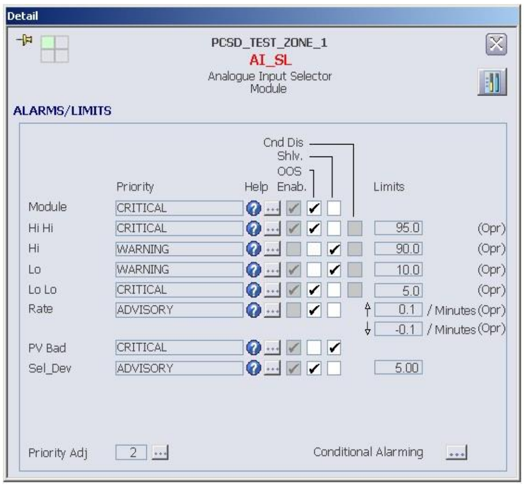
Faceplate for _AI_SOFT_130 module

22.2.6. Detail Display Call up buttons

- **ALM (Alarms/Limits) :** PCSD_MF_AI_ALM_130_DT
- **I/O (Input/Output) :** PCSD_AI_IO_STD_130_DT
- **Info (PV Information):** PCSD_AI_CALC_130_DT
- **TUN (Tuning) :** PCSD_AI_YUN_STD_130-DT
- **VAR (Variables):** PCSD-MF-VAR-130-DT
- **THR (Thresholds):** PCSD-MF-THR-130-DT
- **Info (PV Information):** PCSD-AI-CALC

22.2.7. Detail Display

PCSD Common Detail Display - Alarms/Limits
Detail Display for Alarms and Limits in Analog Input Modules (PCSD_MF_AI_ALM_130_DT)



Detail Display for Alarms and Limits in Analog Input Family (except_AI_SLMG_130) and _O_MLDR_130 control module

Detail Display provides following functionalities to the operator:

- **List of Alarms:** Module Alarm, HIHI Alarm, HI Alarm,

LO Alarm, LOLO Alarm, Rate Alarm, PV Bad Alarm, Sel Dev Alarm. Module Alarm, HIHI Alarm, HI Alarm, LO Alarm, LOLO Alarm and Rate Alarm are applicable to all modules from Analog Input family and _O_MLDR_130 control module. Sel Dev Alarm is applicable to _AI_SL_130. PV Bad Alarm is applicable to _AI_SL_130, _AI_SL3_130 and _AI_SL4_130 modules.

- **Alarm Priority:** Field to indicate the alarms priority.
- **Alarm Help button:** This button launches the Alarm Help available for the corresponding alarm.

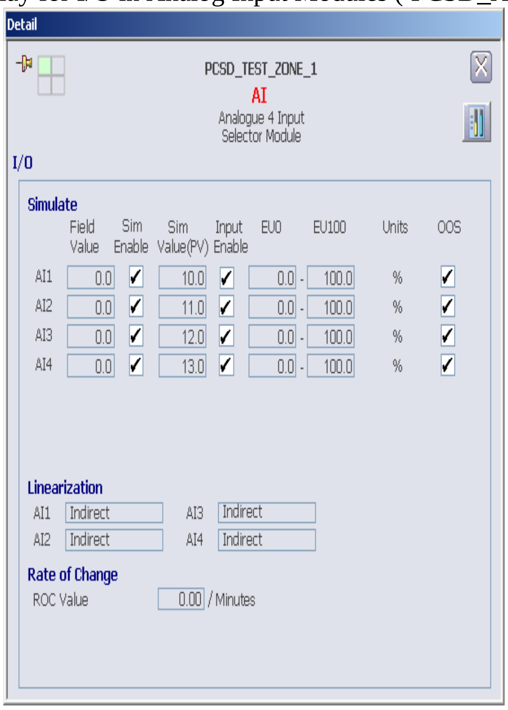
1.  -Alarm Help is Enabled.
2.  -Alarm Help is not enabled but the user has the privilege to edit Alarm Help.
3. Blank- Alarm Help button is not enabled and the user does not have the privilege to edit Alarm Help, or the system does not have a license for Alarm Help.

- **Alarm Priority button:** This button is used to change the alarm priority of the corresponding alarm.
- **Enable:** This checkbox shows ENAB (enabled) state of alarm.
- **Checkboxes to Out of Service:** This check box is used to remove alarm from service or restore alarm to service.
- **Checkboxes to Shelve:** This check box is used to shelve or unshelve the alarm.
- **Conditional Alarming Status:** Indicates that the alarm is conditionally disabled. This indication is available for the High-High, High, Low, Low- Low alarms.

(Not applicable for _AI_SOFT_130 module)

- **Alarm Limit Values:** Indication and entry of the alarm limit values (Applicable to all alarms except PV Bad Alarm)
- **Rate Up and Rate Down Limit Entry Box:** Rate Up and Rate Down Limit Entry boxes are displayed when the Rate Alarm is enabled (Applicable when RATE composite is included)
- **Operator Access to Alarm Limit (text 'Opr'):** Indication that indicates the operator is allowed to change the ala limit setting. This option can be configured separately for each alarm. Applicable to all alarms except PV Bad, Module and Sel Dev Alarm.
- **Priority Adj :** Field to indicate current alarm priority offset. This button adjusts the alarm priority for all alarms as per offset. The alarm priority can be decreased below the configured alarm priority level but cannot be increased above the configured alarm priority.
- **Conditional Alarming Callup Button/ Alarm Hysteresis:** When conditional alarming is applicable, 'Conditional Alarming' button is visible. Calls up the Conditional Alarming popup window when clicked. When conditional alarming is not applicable, Alarm Hysteresis field appears. Entry of alarm hysteresis can be possible from this field.

Detail Display for I/O in Analog Input Modules (PCSD_AI_IO_STD_130_DT)



Detail Display for I/O in Analog Input Family except _AI_SCADA_130, _AI_SLMG_130 control modules

Detail Display provides following functionalities to the operator:

❖ **Simulate:**

✓ **Field value:** Field value indication for AIx.

✓ **Simulate Enable:** A checkbox allows the operator to simulate the process value (PV) of AIx block in engineering units. (e.g. AIx - AI1 or AI2).

Sim Value : A value field provides indication and entry of the simulation value of AIx block in engineering units of OUT_SCALE.

✓ **Input Enable**: Enable/Disable the inputs to input selector block. Applicable for _AI_SL_130, AI_SL3_130 and _AI_SL4_130 modules.

✓ **Simulation Value Range (EU0 and EU100):** Indication of range for the operator to change the simulation value of AIx block.

✓ **Engineering Units:** Field to indicate Engineering units and its range defined in the control module.

✓ **OOS mode Enable for AI:** A checkbox when checked allows the operator to change the mode of AIx block to Out of Service (OOS) mode. When this checkbox is unchecked, mode of AIx block changes to Auto mode. This action requires operator confirmation by confirming the command from the confirmation box that appears when the checkbox is toggled. (Not applicable for _AI_SOFT_130 module)

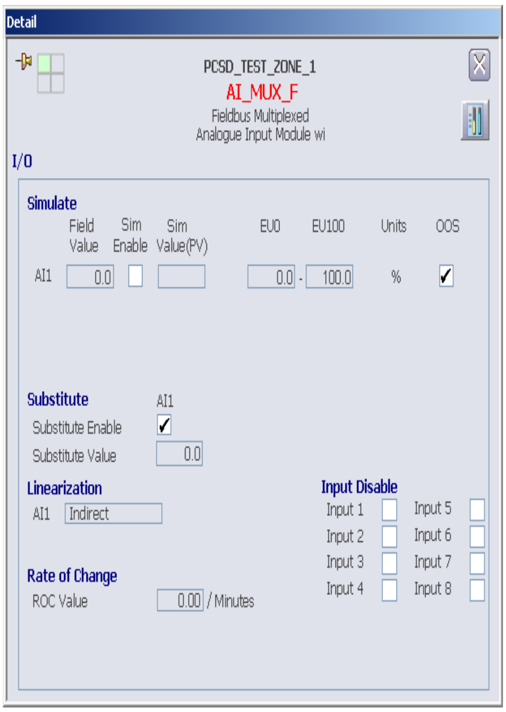
❖ **Linearization:**

- Linearization Type: Determines if the field value of AIx is used directly or converted linearly (Indirect) or is converted with the square root method (Ind Sqr. Root).

❖ **Rate of Change:**

- ROC Value: Indicates rate of change value for rate alarm. (Applicable when RATE composite is included).

Detail Display for I/O in-AI-MUX-F-130 and Fieldbus variant modules in AI Family



Detail Display provides following additional functionalities to the operator:

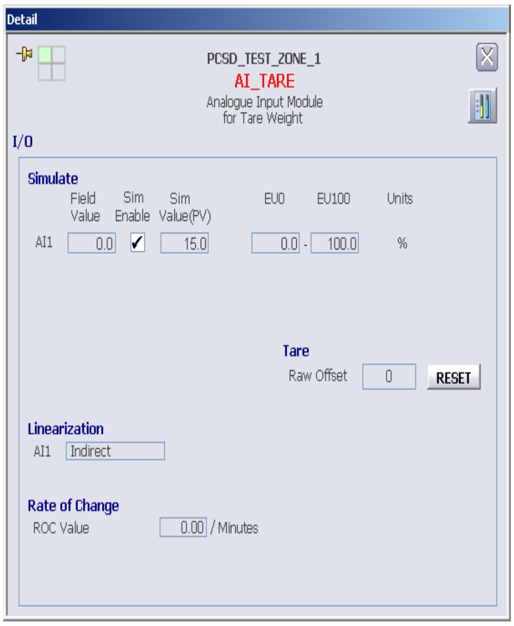
❖ **Substitute:**

✓ **Substitute Enable:** Field to Enable Substitute value entry. Applicable for Fieldbus variant of Analog input modules.

✓ **Substitute Value:** Field to enter the SUBSTITUTE_IN value. Applicable for Fieldbus variant of analog input modules.

✓ **Input Disable:** Checkboxes to disable the corresponding input. The checkbox when checked, disables corresponding input to the input selector block. When checkbox is unchecked, enables the input. (Applicable for_AI_MUX_F_130 module)

Detail Display for I/O in _AI_TARE_130 module



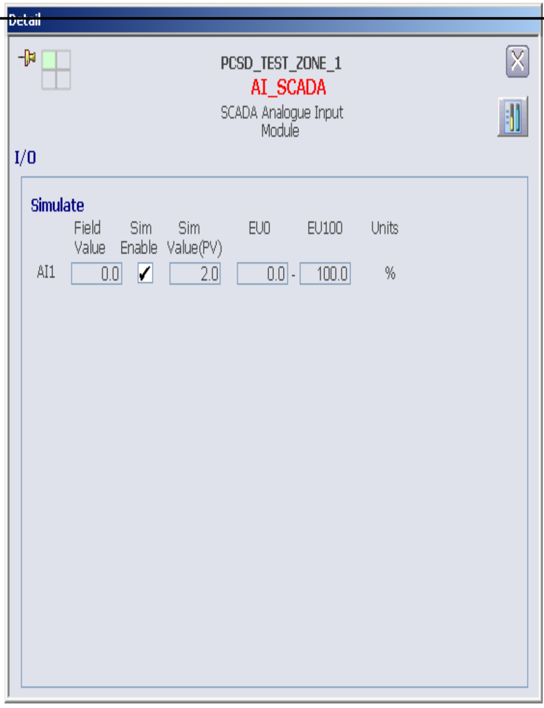
Detail Display provides following additional functionalities to the operator:

❖ **Tare:**

✓ **Raw Offset Value:** Indication of input value for an empty container.

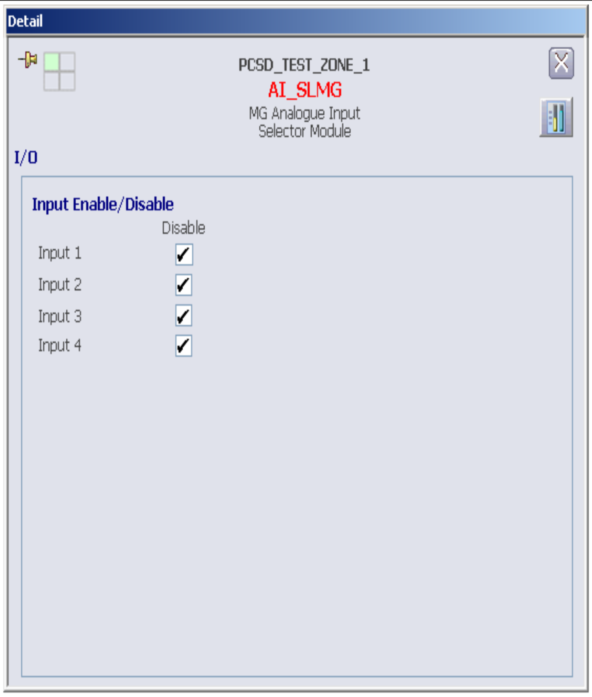
✓ **TARE Reset Button:** Button to reset 'Raw Offset' value. Each time the operator clicks on this button, confirmation box appears. Upon confirmation of the command, the 'Raw Offset' value is reset to 0.

Detail Display for I/O in SCADA Analog Input Module (PCSD_AI_IO_SER_130_DT)



Detail Display for I/O in _AI_SCADA_130 module

Detail Display for I/O of Analog Input Selector Migration Module (PCSD_AI_IO_SLMG_130_DT)



Detail Display for I/O in _AI_SLMG_130 module

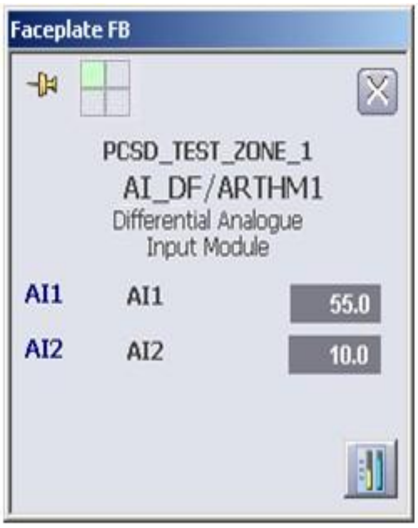
Detail Display provides following functionalities to the operator:

❖ **Input Enable/Disable:**

- **Inputs list:** Number to inputs to appear in this detail display depends on the value of NO_OF_INPUT parameter in the control module.
- **Disable Checkboxes:** Checkboxes to disable the corresponding input. The checkbox when checked, disables corresponding input to the input selector block. When checkbox is unchecked, enables the input.

22.2.8. Supporting Display

Differential Analog Inputs Faceplate FB (PCSD_InDiff)



Differential Analog Inputs Faceplate FB

The purpose of this Faceplate FB is to show the 2 field values used by the differential analog input logic. The differential analog input feature is used by Analog Input Modules (_AI_DF_130) and Control Loop Modules (_CDF_130). The Faceplates of these modules directly show the result of the subtraction. This Faceplate FB allows the operator to see the value and status of each analog input.

- **Input1 Description:** Description for the first analog input.
- **Input2 Description:** Description for the second analog input.
- **Input1 OUT Value:** Field value coming from the first analog input. Background rectangle color of this value field changes to maroon color and orange color when status of input changes.
- **Input2 OUT Value:** Field value coming from the second analog input. Background rectangle color of this value field changes to maroon color and orange color when status of input changes.

This supporting display is applicable to following control module classes:

- Differential Analog Input Control Module (_AI_DF_130)
- PID Differential Analog Control Module (_CDF_130)

22.2.9. Input Selection Faceplate FB (PCSD_InSel)

Input Selection Faceplate FB

The purpose of this Faceplate FB is to show the up to 4 field values used by the input selection logic, to show which input is selected and to give the possibility to the operator to select the input. The input selection feature is used by Analog Input Selector Modules and Control Loop Modules (_CSL_130). The faceplate of these modules directly show the result of the selection. Above snapshot shows four inputs on Faceplate FB and it also supports three (_AI_SL3_130) and two (_AI_SL_130) inputs. Indication of two and three inputs is same as four inputs. Group of AIx Value and AIx button is visible on Faceplate FB when respective inputs are enabled from I/O detail display.

- **Selected Mode:** Value of selection logic mode. Indicates 'Auto Selection' when Auto mode is selected and 'Manual Selection' when Man mode is selected.

- **AI4 Selected:** Button to select the Analog Input when the mode is manual. This example shows the selection of AI4. (Applicable for AI1, AI2, AI3 and AI4). Text on the button changes as per the description configured in INPUTx parameter. (x: 1 to 4)

- **AI1, AI2, AI3, AI4 Values:** Indication of Field values coming from the four analog inputs configured. Background rectangle color of these value fields changes to maroon and orange when status of input changes.

- **Selection Type:** Selection Type depending on configured value of Select Type (SELECT_TYPE) of ISEL1 block (Default-MINIMUM).

- **Mode Selection:** Buttons to select the mode of this function (Auto/Manual)

  ✓ When the mode is Auto, the selection type parameter is used to determine the input to select.

✓ When the mode is Man, the operator can select the input via the buttons AI1,AI2,AI3 and AI4 available on this Faceplate FB.

This supporting display is applicable to following control module classes:

- Analog Input Selector Control Module (_AI_SL_130)
- Analog 3 Inputs Selector Control Module (_AI_SL3_130)
- Analog 4 Inputs Selector Control Module (_AI_SL4_130)
- PID Control Module with Input Selection (_CSL_130)

Pressure-Temperature Compensation Faceplate FB (PCSD_PTCOMP)

pressure-Temperature Compensation Faceplate FB

- **Compensated Flow Value:** A value field provides indication of the calculated Compensated Flow. Background rectangle color of this value field changes to maroon and orange when status of Compensated Flow changes.
- **Pressure Value:** A value field provides indication of the current Process Value Pressure Input. Background rectangle color of this value field changes to maroon and orange when status of Pressure changes.
- **Raw Flow Value:** A value field provides indication of the calculated Raw Flow. Background rectangle color of this value field changes to maroon and orange when status of Raw Flow changes.
- **Temperature Value:** A value field provides indication of the current Process Value Temperature Input. Background rectangle color of this value field changes to maroon and orange when status of Temperature changes.
- **Fluid Type:** A field provides selection of Fluid Type.
- **Compensation Type:** A field provides selection of Compensation Type.
- **Flowmeter Type:** A field provides selection of Flow meter Type.

23. DeltaV Software Licenses

Before you can download your DeltaV software configuration, you must attach the System Identifier to the Professional PLUS workstation, load your licenses on the Professional PLUS workstation from a license disk, and assign licenses to the workstations and controllers in your DeltaV network. Before loading and assigning your licenses, take a few minutes to read the next sections about DeltaV software licenses.

23.1. System Software

Major system software versions require a license. A major version is one in which the first digit in the version number changes from the previous version. A Major Version License is required when you upgrade from one major version to another and for first-time installations.

23.2. Controller Software

Controller software for continuous control is licensed through four I/O-based, system-wide licenses. For batch control, a fifth system-wide license is added; it is the Advanced Unit Management license. The system-wide licenses are assigned to the Professional PLUS workstation and determine both the functionality available in every controller in the system and the potential size of the system. System size is expressed in Device Signal Tags (DSTs).

When purchasing controller licenses, it is necessary to understand the difference between the four I/O-based licenses, which are:

• Discrete Monitor Input
• Discrete Control Output
• Analog Monitor Input
• Analog Control Output

For the controllers in your system, select specific DST sizes of the four I/O-based licenses by counting the number of discrete inputs, discrete outputs, analog inputs, and analog outputs required for your process.

23.3. Workstation Software

When installing DeltaV Software on a PC or server, the installation wizard enables you to define the PC or server as one of three node types: Professional PLUS, Operator, or Application. After the software installation is complete, you determine the functionality and size of the workstation with software licenses. Key licenses are bundled in software suites. Each suite enables the use of specific functions and applications as defined in the product data sheets. The following table defines which software suites can be assigned to the three node types:

Workstation Node Type	Supported Software Licenses
Professional PLUS workstation	Professional PLUS license suite, DeltaV Experience license
Operator Station	Base license suite, Maintenance license suite, Professional license suite, and Operator license suite
Application Station	Application license suite

After you install a license suite, you can install add-on and scale-up licenses. Add-on licenses add new licensed features and functionality to the initial license and scale-up licenses increase the capacity of an existing license feature.

To expand the DST capacity of a workstation that has the DeltaV Experience license assigned, you must first assign a base Professional PLUS license suite. Assigning a base license suite uninstalls the DeltaV Experience license and enables you to assign I/O base and scale-up licenses.

23.4. Redundant Controllers

Each pair of redundant controllers requires a separate redundant controller license

24. Engineering & Operator Workstations

The Engineering Workstation (EWS) is for project development, including configuration of graphics, logic, alarms, security, etc.

The Operator Workstation (OWS) provides the operator interface, including color graphics, faceplates, alarms, logging, trends, diagnostics, etc. Usually, the EWS includes an OWS for testing and troubleshooting.

The DeltaV Engineering Workstation for DCS and SSS systems are separated but Operator Workstations can be common or separate for each DCS and SSS systems.

The specifications of Engineering Workstation are as follow:

Tower, Processor: Intel Core i7 last generation, Memory: 32GB, HDD: 2TB, DELL manufacturer.

System generate an alarm when operations during workstation failure and Operational function and printer output continue its duties.

Operator workstations served as dedicated operator workstations and data servers, and be of redundant application. In case of any one failure, the other replace it to realize full function. Any Operator Console may be used for reconfiguration and reinstallation purpose in case of Engineer workstation failure and the access to the configuration activities protected by means of hardware and software keys.

Engineer workstation served as both engineer and operator workstation, and the functions for all engineer and operator also provided and realized based on user privileges.

Each workstation suitable for use by a seated operator with easy viewing of all monitor displays and associated function keys.

All of these workstations utilized to monitor, control and configure the entire system.

The workstations utilized to allow program development and LCD display changes.

Ability to make changes in the program, database, and LCD displays on-line/off-line are defined.

This combination and associated electronics are defined as "Human-Machine Interface"(HMI). Each workstation allows easy access to internal equipment from the front, top and/or rear.

The engineering workstation used for the software modifications and/or updating activities. The engineering workstation allow on line and off line configuration. The minimum functions to provide by the engineering workstation are the following:

- Configuration editor
- System/configuration bootstrap
- Control algorithm generation and modification
- Data base generation
- HLCL and logic functions builder
- Query and sorting utilities
- Compilation utilities

- Graphics display generation, modification and hardcopy printing
- Modification of operational data
- Tuning parameter modification and printout
- System access configuration
- Process Area/Unit assignment
- Printer assignment
- Utility program accesses (load and save utility)
- Help message definition
- Diagnostics

24.1. Detailed Process displays

These displays provide a schematic representation of the plant, and are the primary operator interface to the plant equipment. Therefore, the majority of plant Control and equipment module manipulation, both manual and automatic, is performed at this level of display. Operator manipulation of plant items will be achieved by opening the appropriate Faceplate, followed by the desired control command, such as open, close, start, stop, etc. These displays typically cover Units within Process Cells and/or Areas, but they may also cover (complex) Equipment Modules.

A range of toolbar icons and onscreen graphic links facilitate easy navigation between and across levels within the hierarchy.

Each PCSD display has a similar format and layout, incorporating a toolbar, a custom (project specific) display-area and an alarm banner. Each display from PCSD is based on a template appropriate to the monitor resolution of 1280x1024 resolution, 1680x1050 resolution (supporting split screen mode, where the right part of the screen is used for alarm list, main display history and module history windows, and full-screen mode), 1920x1080 (supporting split screen mode, where the right part of the screen is used for alarm list, main display history and module history windows, and full-screen mode) resolution.

25. Backup and restore

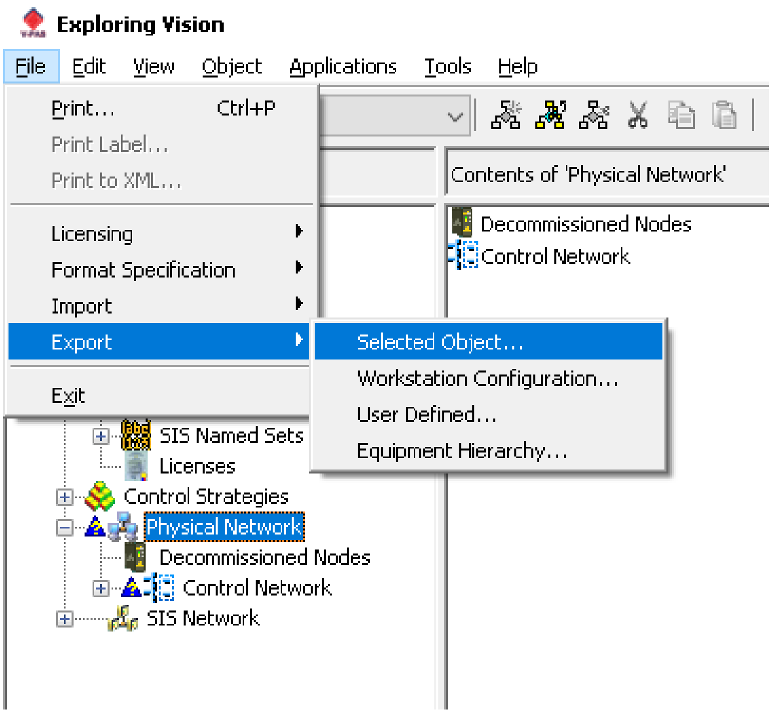
It is vital that you maintain frequent backups of your DeltaV system's configuration For this reason Performing regular backups enables you to restore information that otherwise would have been lost in a catastrophic system failure, such as a hard disk crash, accidental user deletions, operator error, and so on. Include the following items (at the minimum) in your configuration backup:

- the controller's configuration
- operator displays
- historical configuration
- historical data archives
- Batch Historian data

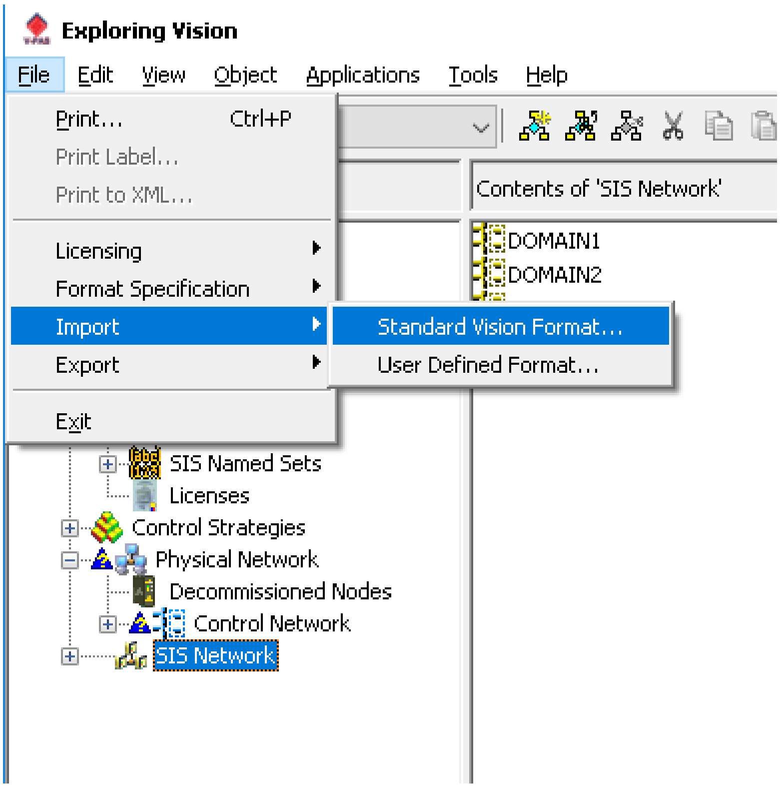
Perform a backup for any data that you change. Back up daily until you are no longer changing the data. Store several of the most recent backups in a safe location free from environmental hazards (such as extreme heat, extreme cold, dust, or magnetic devices, including speaker phones).

25.1. DeltaV System backup and restore

1. Backup: In Exploring DeltaV select desired part to backup then select File → Export → Selected object



2. Restore: In Exploring DeltaV to Restore select File → Import → Standard DeltaV format



25.2. HMI backup and restore

1. Backup: Copy desired files from below directory then paste to backup folder

→ This PC → Local Disk (C:) → DeltaV → DVData → Graphics-iFix → Pic

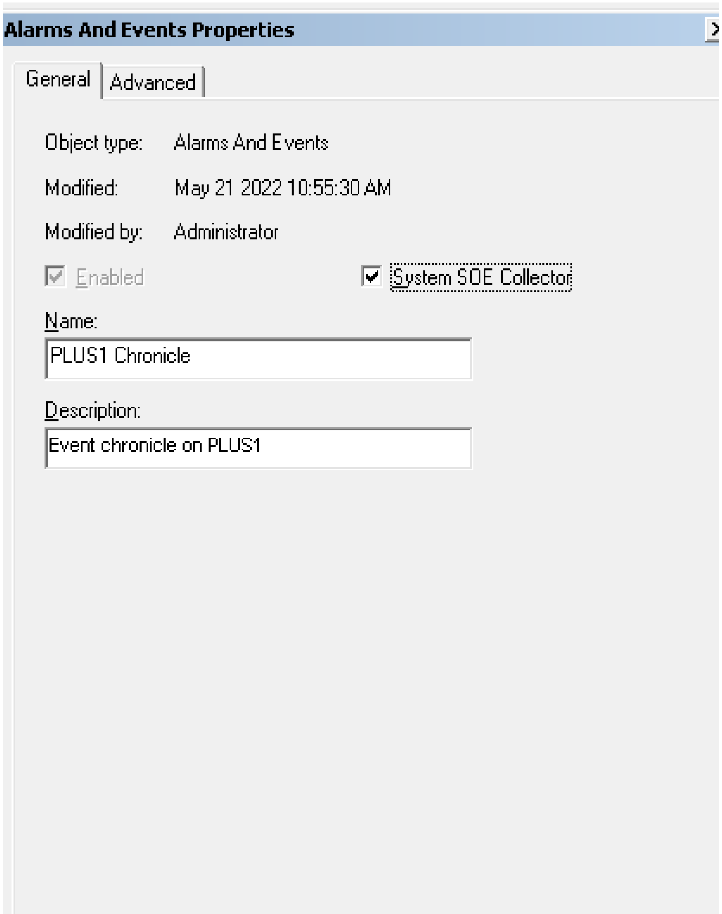
2. Restore: Copy desired files from backup folder then paste to below directory

→ This PC → Local Disk (C:) → DeltaV → DVData → Graphics-iFix → Pic

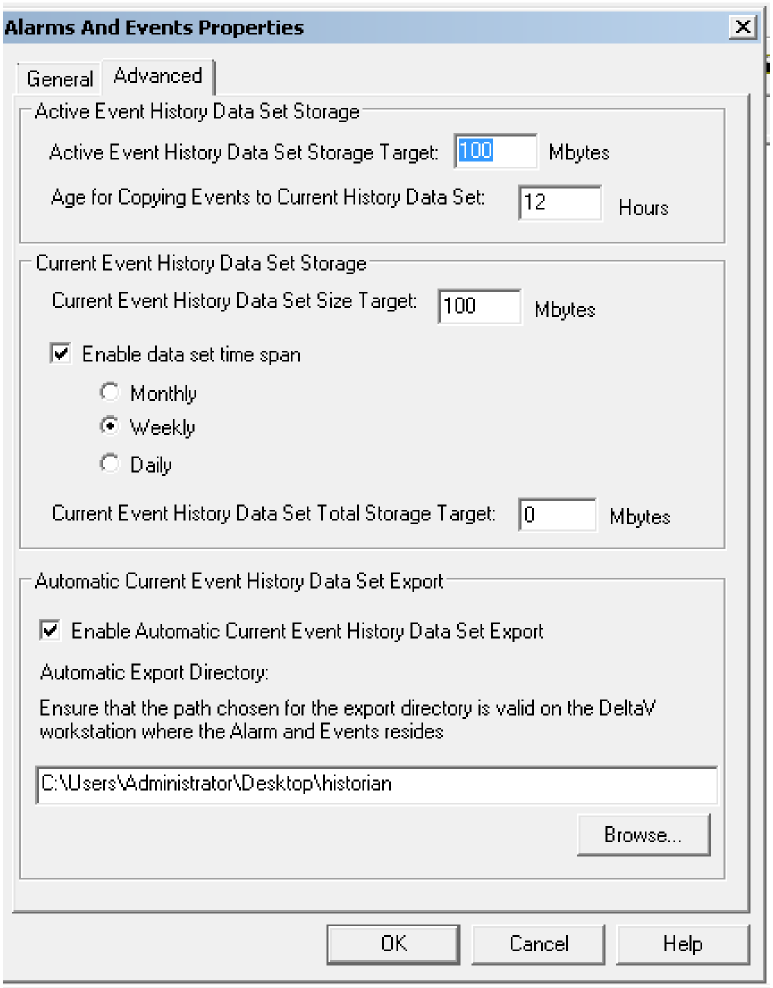
25.3. Alarm and events backup and restore

To restore "alarm and events" we must follow bellow route:

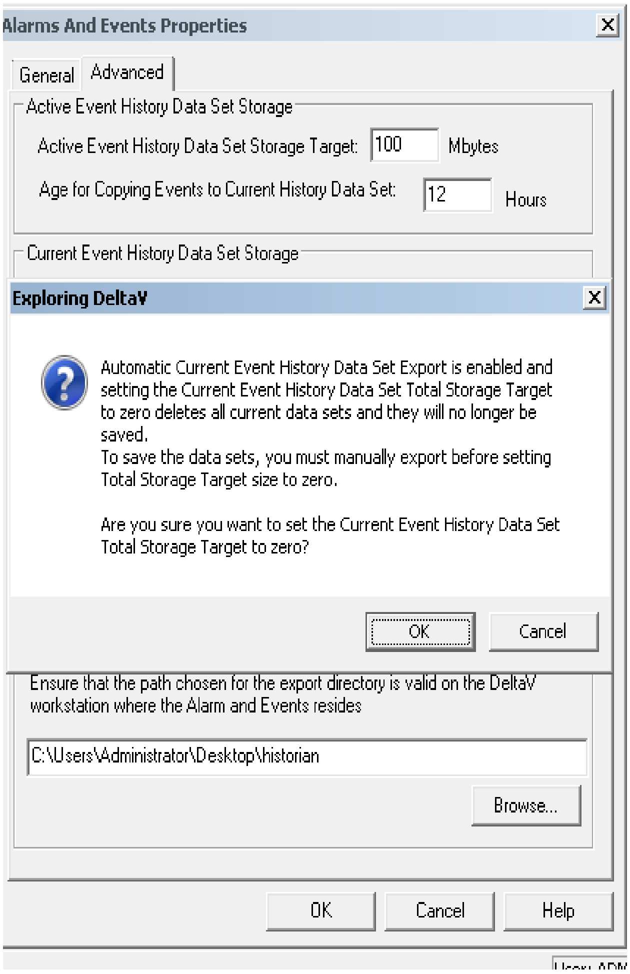
System configuration → Control network → Plus1 → Alarm and events on controller choose Alarm and Events then right click and choose properties.



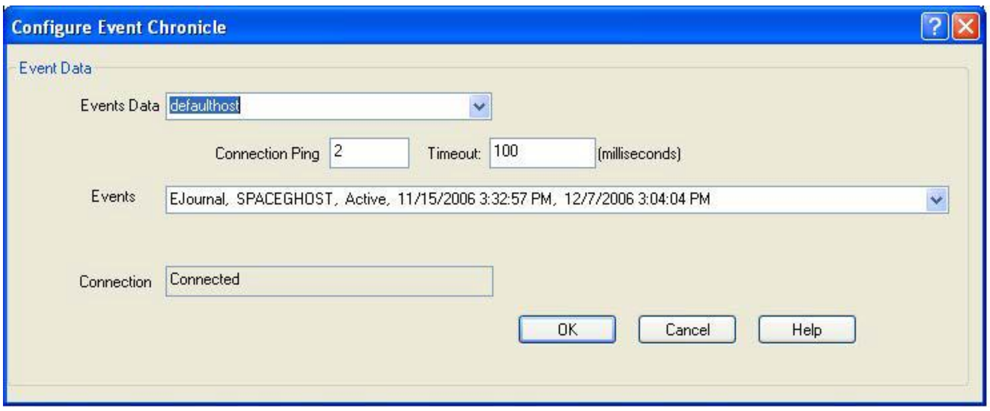
Alarms And Events Properties
General
Advanced
Object type:
Alarms And Events
Modified:
May 21 2022 10:55:30 AM
Modified by:
Administrator
Enabled
System SOE Collector
Name:
PLUS1 Chronicle
Description:
Event chronicle on PLUS1



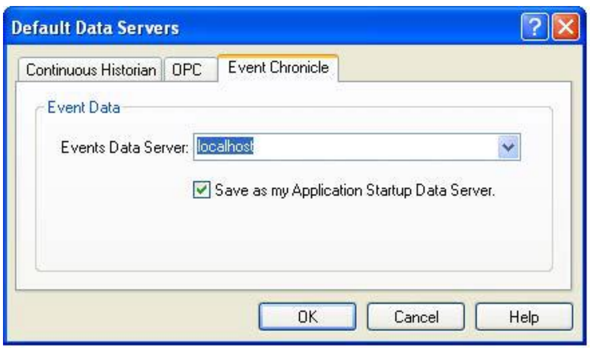
As we can see bellow on advanced tab can define various parameter such as: active event history, data set storage target, as you can see on bellow pictures:



Alarms And Events Properties
General
Advanced
Active Event History Data Set Storage
Active Event History Data Set Storage Target:
100
Mbytes
Age for Copying Events to Current History Data Set:
12
Hours
Current Event History Data Set Storage
Exploring DeltaV
Automatic Current Event History Data Set Export is enabled and setting the Current Event History Data Set Total Storage Target to zero deletes all current data sets and they will no longer be saved.
To save the data sets, you must manually export before setting Total Storage Target size to zero.
Are you sure you want to set the Current Event History Data Set Total Storage Target to zero?
OK
Cancel
Ensure that the path chosen for the export directory is valid on the DeltaV workstation where the Alarm and Events resides
C:\Users\Administrator\Desktop\historian
Browse...
OK
Cancel
Help

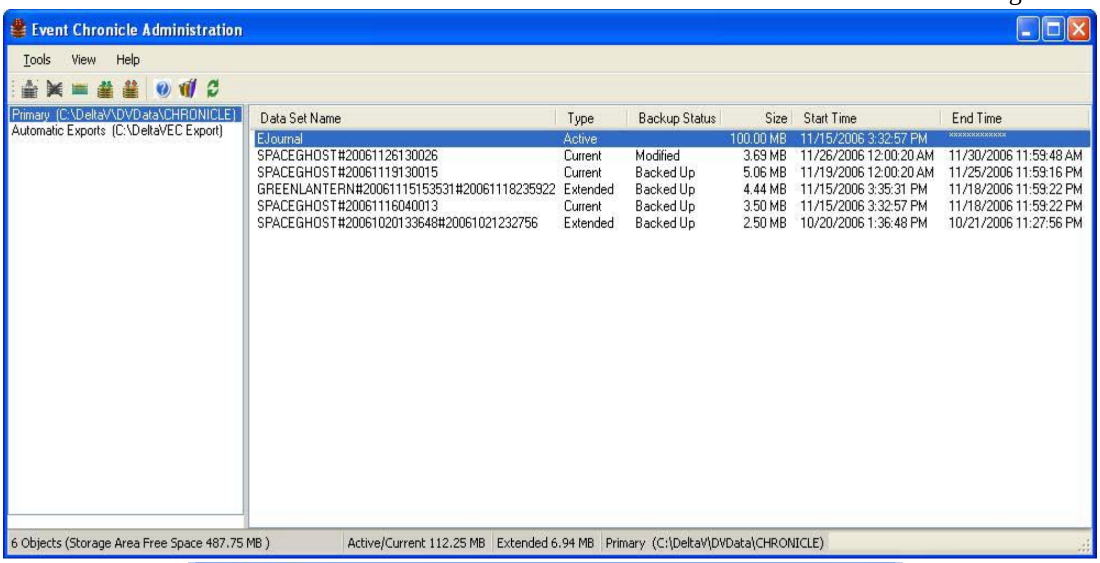


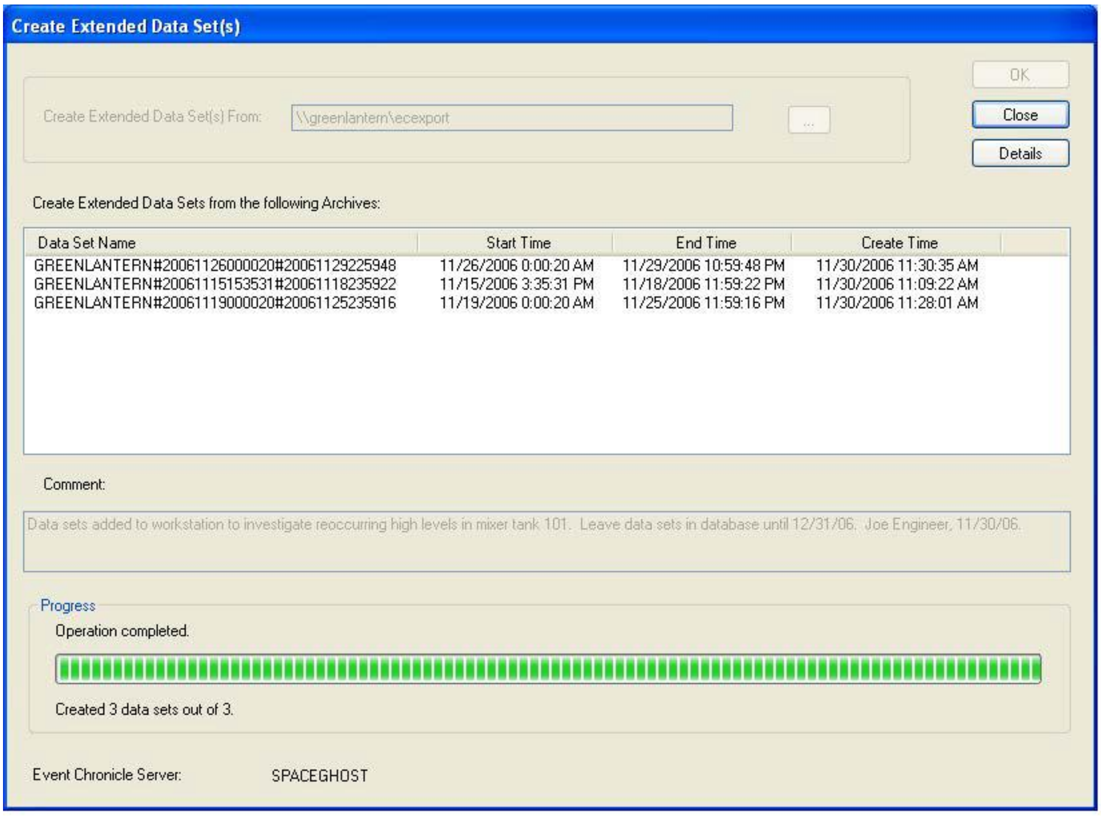
Configure Event Chronicle
Event Data
Events Data
defaulthost
Connection Ping
2
Timeout:
100
(milliseconds)
Events
EJournal, SPACEGHOST, Active, 11/15/2006 3:32:57 PM, 12/7/2006 3:04:04 PM
Connection
Connected
OK
Cancel
Help



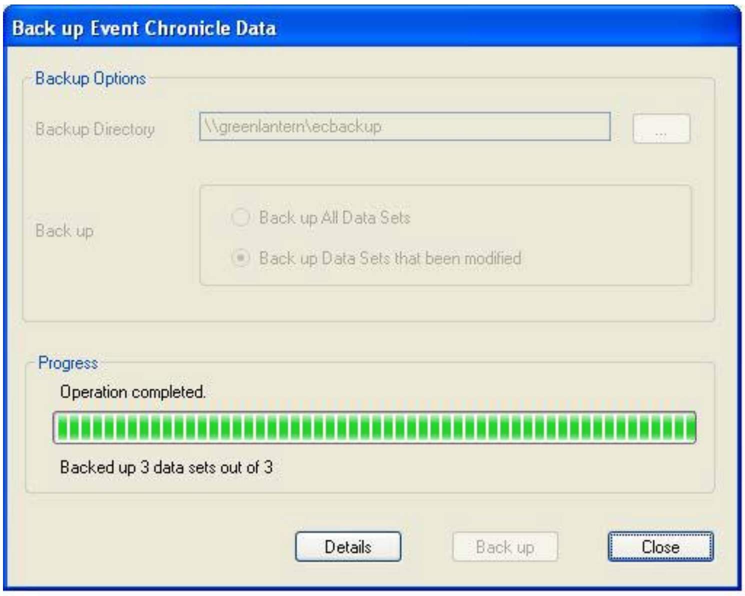
Default Data Servers
Continuous Historian
OPC
Event Chronicle
Event Data
Events Data Server:
localhost
Save as my Application Startup Data Server.
OK
Cancel
Help

Restoring all of control module:

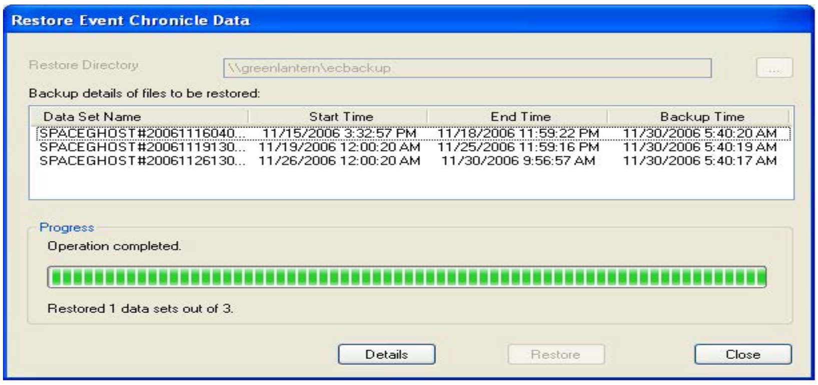




Create Extended Data Set(s)
OK
Close
Details
Create Extended Data Set(s) From:
\\greenlantern\ecexport
...
Create Extended Data Sets from the following Archives:
Data Set Name | Start Time | End Time | Create Time
GREENLANTERN#20061126000020#20061129225948 | 11/26/2006 0:00:20 AM | 11/29/2006 10:59:48 PM | 11/30/2006 11:30:35 AM
GREENLANTERN#20061115153531#20061118235922 | 11/15/2006 3:35:31 PM | 11/18/2006 11:59:22 PM | 11/30/2006 11:09:22 AM
GREENLANTERN#20061119000020#20061125235916 | 11/19/2006 0:00:20 AM | 11/25/2006 11:59:16 PM | 11/30/2006 11:28:01 AM
Comment:
Data sets added to workstation to investigate reoccurring high levels in mixer tank 101. Leave data sets in database until 12/31/06. Joe Engineer, 11/30/06.
Progress
Operation completed.
Created 3 data sets out of 3.
Event Chronicle Server:
SPACEGHOST



Back up Event Chronicle Data
Backup Options
Backup Directory
\\greenlantern\ecbackup
...
Back up
Back up All Data Sets
Back up Data Sets that been modified
Progress
Operation completed.
Backed up 3 data sets out of 3
Details
Back up
Close



26. System Time Synchronize

V-PAS uses the Network Time Protocol (NTP) to synchronize system time for all workstations and controllers on the V- PAS Control Network.
V-PAS systems that require precise system time derive master time from a Global Positioning System (GPS) network time server configured to a specific IP address on the V-PAS Control Network.
Packages include upcs, plcs ,… could synchronized by both hardware (conventional DO) and modbus paluses both of methods are applicable for this purpose.
Execution for time synch is every 24 hour at 00:30 am.
As it shown on below swiches and domain servers are redundant

27. System application capacity

The following tables specify the capacity values of the DeltaV system.

workstation capacity

Description	Limit
Maximum open DELTAV Center applications	1
Maximum open Control Module through Studio applications	4
Alarm List	1
Continuous (Historical data values) Historian	250
History sample rate per second	250
Open faceplates per module type	4
Open detail displays	1
Open pictures (HMI pages)	25

Professional PLUS workstation

Description of	Limit
Plant areas (including remote zone areas)	250
Modules per unit	255
Alarm types For the Professional PLUS Station only	(available) 235
DELTAV user accounts per system	200
Maximum open engineering tasks	60
Number of concurrent Professional Workstations connected	10

Application Stations

Description	Limit
OPC data values	30,000
Maximum assigned modules	3,000
SCADA tags	25,000
Historical data values	30,250
History sample rate per second	3000

Controllers Capacity

Description	Limit
DST Limit	1500
I/O cards per controller	64
SCADA Tags	3200
Max Data Values Sent	4000/second
Max Data Values Received	500/second
Max Unsolicited Client Nodes	120
Module Execution Rates	100ms, 200ms, 500ms, 1s, 2s, 5s, 10s, 30s, 60s
User Memory	96 MB
Fuse Protection (Internal)	3.0 A, non-replaceable fuses
Power Dissipation	5 W typical, 7.0 W maximum

28. Inter-Controller communication guidelines

Reads are the preferred method for communications. For example, Controller B has an input parameter defined as an external reference to a parameter in Controller A. Controller B initiates unsolicited updates with a one-time request for data from Controller A. From then on, Controller A updates Controller B when the exception criteria are met (a status or data value changes). Any references in Controller B are then reading buffered data that is automatically updated by the system's communications processes.

The unsolicited send occurs at the same priority as low priority modules on the source node. When referencing a parameter in another node in an expression, you can verify communications with that parameter or node with the. CST field. The. CST field can be referenced in blocks such as the CALC block or SFC expression.

When verifying the status of a reference in another controller, CST is recommended instead of.ST. The .ST value is set by the sending node. As such, if communications is lost, the .ST value does not change and holds its last value. The CST value is set by the receiving node. Every time the module executes, the. CST value is updated with the current remote status.

Your configuration should be designed to accommodate occasional, intermittent bad status. It may not be good practice to immediately trip an emergency shutdown on a single instance of bad status. You can configure timeouts on bad status as appropriate to avoid unnecessary shutdowns.

Similarly, it may not be appropriate to design control logic with immediate response to unexpected events. On any interlock or shutdown conditional logic, provide appropriate time delays before the system takes significant action in response to the event. For example, it may be appropriate in some cases to allow a bad status for several seconds before shutting down or taking other serious action. Time delays can prevent unnecessary shutdowns in the event of intermittent events caused by electrical noise, redundant controller switchovers, partial downloads, and so on.

Buffered Asynchronous Writes - When calculation block expressions, action block expressions, and SFC/PLM/Phase Class actions write to a parameter in another node, the output is buffered. This is followed by an asynchronous write request to the destination node with the new data. These occur at a priority lower than low priority modules on the source node. Buffered asynchronous writes are also done for external reference parameters in other nodes.

29. function block

Parameters are data used in a function block to perform calculations and logic. Some parameters are defined and unchangeable for certain function block. Some parameters default to a most common value but can be modified by the user or another function block. Others must be set by the user before executing the function block. Parameters that can be changed are called writeable parameters.

Parameters can be described by the type of information they provide to the function block:

- Input parameters contain values from the operator, the field device, or other function block.
- Output parameters are the output values generated by the function block or set by the operator or another function block.
- Contained parameters are used only within the function block for calculation, logic, and status determination function.
- Mode-controlled parameters vary with the mode of the block.

Parameters can be dynamic, static, or non-volatile, depending on how they are restored after power failure:

- Dynamic parameters are calculated by the block algorithm and do not need to be restored after power failure because the block recalculates the value.
- Static parameters have a specific value that must be restored by a device for use after power failure.

- Non-volatile parameters are written on a frequent basis and the last saved value (in most cases) is restored by a device after power failure.

Some parameters are extensible. That is, you can extend or increase the number of these parameters in a function block.

Some parameters are option bit strings. They contain bit-encoded information that specify control strategy options, I/O value processing, status handling and processing, or the type of control logic used in the block.

Each parameter is transmitted in a certain data type format. This data can be transmitted between block for control, trending, alarming, and diagnostics.

The parameters visible in Control Studio may vary depending on level of configuration completed. For example, certain function blocks must be assigned to I/O before all the parameters are visible.

Function block modes are described below. Supported modes vary with each function block. Some blocks support modes that are not supported in other blocks.

Out of Service (OOS): The block algorithm is not active. The output is maintained at the last value or at a specified failure action value.

Initializing Manual (IMan): The upstream block of a cascade pair is put into this mode when its downstream partner is in a non-cascade mode. This prevents the upstream block from closing the cascade. When the downstream block returns to a cascade mode (Cas or RCas), the upstream block leaves IMan and returns to its target mode.

Local Override (LO): The block is put into this mode when tracking is activated; the output is driven to a value other than that generated by normal block execution. When tracking is deactivated, the block returns to its target mode.

Manual (Man): The block output is set directly by the operator.

Automatic (Auto): In this mode, the control algorithm of the block is active. An operator-entered setpoint is used in the control algorithm to determine the block output.

Cascade (Cas): This mode is similar to Auto except that the setpoint is supplied by another function block through the CAS_IN parameter. The block maintains a back calculation output value (BKCAL_OUT) to provide bump less mode transfer when the mode is changed.

Remote Cascade (RCas): This mode is similar to Cas except that the setpoint is supplied by an external control program through the RCAS_IN parameter. The block maintains a back-calculation output value (RCAS_OUT) to provide bump less transfer when the mode is changed.

Remote Out (ROut): This mode is similar to Man except that the OUT value is supplied by an external control program rather than directly by the operator. OUT is supplied through the ROUT_IN parameter. The block maintains a back-calculation output value (ROUT_OUT) to provide bump less transfer when the mode is changed.

30. Internal Reference Parameter

An Internal reference allows you to refer to any input, output, or parameter available in the current module. Internal references are best configured using the parameter browser available in all DeltaV expression dialogs. The browser opens a graphical list of valid blocks and parameters. By constructing a parameter path from the browser, you can avoid the potential of typographical errors and case sensitivity when referencing block parameters.

A write to an Internal reference parameter changes the value of the parameter being referenced. The value of the Internal reference parameter itself changes at the beginning of the next scan of the module.

The fields for an Internal reference parameter are described in the following table.

Name and Purpose	Type	Configurable	Readable	Writable
CST (Connection Status) - Use to see if the reference has been resolved (that is, the value has been found and can be read)	Integer	No	Yes -3 means 'external reference not resolved' -2 means 'parameter not configured' -1 means 'module not configured' 0 means 'good' 1 means 'not communicating'	No
AWST (Asynchronous Write Status) - Indicates if the last attempt to write the referenced parameter was successful	Integer	No	Yes -4 means 'write rejected' -3 means 'external reference not resolved' -2 means 'parameter not configured' -1 means 'module not configured' 0 means 'success' 1 means 'not communicating' 2 means 'write pending'	No
.ST - The status of the referenced value.	Integer	No	No	No

31. IP address

The DeltaV system automatically assigns IP addresses to the Professional PLUS, Operator and Application stations, and controllers. If Network Time Protocol servers are used, they must be assigned specific IP addresses. Reserved IP addresses can be used for other equipment such as switches, domain controllers, management stations, and so on.

If the NTP server is a DeltaV workstation, it is automatically assigned an IP address. If you use an external Network Time Protocol (NTP) server, you must assign it the following IP addresses:

- Primary NTP server on the primary network: **10.4.0.6**
- Backup NTP server on the primary network:**10.4.0.6**
- Primary NTP server on the secondary network:**10.8.0.6**
- Backup NTP server on the secondary network:**10.8.0.6**

32. dynamo set

When the displays are configured, many items such as a valve or an Analog value etc. are required to show on more than one display. typically, these are represented with the help of animated objects which will change color with opening or closing of a valve, or will show the numerical value obtained from a temperature transmitter. Such animated objects can be saved as „reusable objects" called dynamos.

32.1. PCSD Dynamo Set

Wherever possible, the displays for this project shall be created using the dynamos from the PCSD library dynamo set.

32.2. Display Conventions

32.2.1. General information on static elements

Process displays will provide a schematic representation of the process equipment. They will not be an exact replica of the P&IDs.
Wherever practical, the main process flow direction will be kept from left to right horizontally and top to bottom vertically across the display with inputs at the left and outputs at the right. Process lines will be drawn such that the number of angles is minimized.

32.2.2. Identification of plant equipment and vessels

Equipment and control loops not monitored or controlled via the Distributed Control System, will not be shown on the display; unless specifically required.
Vessel shapes will be taken from the P&IDs. Where a level input exists, vessels will be provided with a Level Bar Graph. Process and Utilities Pipe work relating to the vessel will be shown as per the P&ID. All plant equipment and vessels shown on graphics shall have the same tags as identified on the P&ID drawings.

32.2.3. Font

Description	Color	Font
Equipment Font (Display Tag)	Black	ARIAL, 10, Normal
Vessel Font	Black	ARIAL, 12, Bold
Picture Title	Black	ARIAL, 14, Bold

32.2.4. Numeric Value Form

Numerical values will be displayed using the following display format:

Engineering Range	Display Format
0 to 9.999	x.xxx
10 to 99.9	xx.xx
100 to 999.9	xxx.x
1000	xxxx

Note:

1. For the flow rates numeric display length will be 7 digits. For flow totalizers the length will be 8 digits.
2. Negative values will be indicated with "– "sign.

32.2.5. Process Line

There are four-line styles identified.

Description	Sizing	Figure	
Primary Process Lines	Line width = 2		Process colour
Secondary Process Lines	Line width = 1		Process colour
Signal Line	Line width = 1 Edge Style: EdgeStyleDot BackgroundStyle:Transparent	----------	White
Instrument Line	Line width = 1		Black

The rules for intersection of two process lines are as follows:

- ✓ Where two primary/secondary process lines cross each other, the vertical line will be broken.
- ✓ Where the secondary process line crosses a primary process line; the secondary process line will be broken.

32.2.6. Colors for static elements and background

Process displays will utilize colors as listed in following Table:

Description	Fluid Code	Color Name

Display Background		N.A	Gray 75
Equipment	**Border**	N.A	Dark Gray
	Fill	N.A	Gray
	Tag	N.A	White

32.2.7. Module Dynamos

Animated Objects or dynamic symbols i.e. Dynamos are pre-configured and pre-validated objects. The set of available dynamos can be seen in the configure mode of DeltaV Operate, and is known as a Dynamo Set. There are 2 (two) PAS dynamo sets in PCSD, Human Centered Design (HCD) dynamo set 'PCSD_HCD_130.fds'and Conventional dynamo set 'PCSD_Conventional_130.fds'.

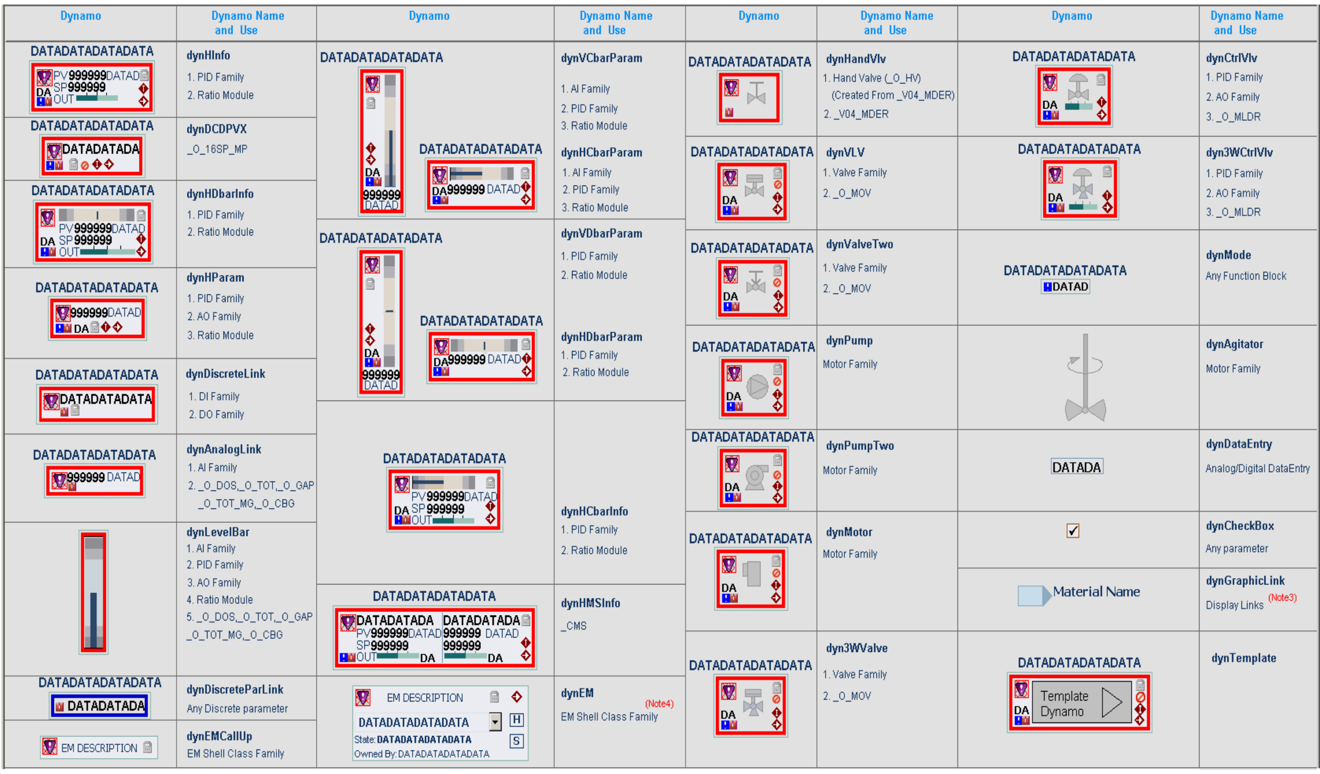
32.2.8. Colors Specification for Dynamic Elements

Process displays will utilize colors as listed in following Table for Process Units, according DeltaV Standard Library:

STATUS	COLOR
Fault Status	Yellow (Fixed/Blink)
MODBAD Status	RED(Fixed/Blink)
Running/On/Open Status	Green (Fixed)
Shutdown/Off/Close Status	RED(Fixed)

Important Note: According DeltaV Default Colour Coding in PCSD HCD Library, the Running/ON/OPEN Status has been shown with White Color and Shutdown/Off/Close Status has been shown with Gray Color. Please note, this is just default Color coding and for ESBR project, specified color in above table will be used. Therefore, in next pages, all Dynamo objects will be shown with DEFAULT COLOR according EMERSON documents that will be changed in Customized page in running system according project requirement.

32.2.9. Dynamo Objects



32.2.10. Features in HCD Dynamo

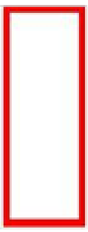
PCSD HCD dynamos are compatible with PCSD display templates.

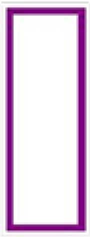
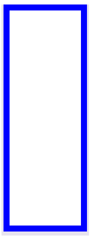
❖ **Alarm Icon**

Alarm Icon	Description
	Critical Alarm (Active or Unacknowledged). Cross-hatching is visible behind the icon when the alarm is unacknowledged and inactive.
	Warning Alarm (Active or Unacknowledged). Cross-hatching is visible behind the icon when the alarm is unacknowledged and inactive.
	Advisory Alarm (Active or Unacknowledged). Cross-hatching is visible behind the icon when the Alarm is Unacknowledged and inactive.
	Suppressed Alarm (otherwise Active). Visible only if there is no other Active or Unacknowledged Alarms. When the Suppressed Alarm Icon is visible, the Status outline surrounding the dynamo is shown instead of the Alarm Outline. Refer to tables Alarm Box and Status Box.

❖ **Alarm Box**

Color of the Alarm Box changes as shown in below table depending on the alarm priority. Alarm Box is not visible for Alert priority alarm.

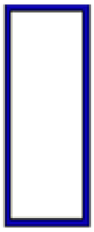
Alarm Box	Description
	Color of the Alarm box is Bright Red when Critical priority alarm is Active.

	Color of the Alarm box is Bright Yellow when Warning priority alarm is Active.
	Color of the Alarm box is Magenta when Advisory priority alarm is Active.
	Color of the Alarm box is Bright Blue when active alarm is suppressed.

The dynamo is surrounded by a colored rectangle at run time to indicate that a condition needing attention exists when there are no active alarms. Condition that set the status icons also set the visibility of the status box.

The status box can be visible at run time when the alarm box is not visible. The status box is visible and a user preference applies. The color of the status box is based on the named Status border color in the Theme Colors table.

- **Status Box**

Status Box	Description
	**Status box will be visible if any of the following conditions meets:** • Abnormal mode symbol is visible • Bad I/O • Simulation is enable • Module not running Color of status box is theme compatible.

The visibility of the Status outline can be disabled for the specific types of status as shown in table 'Status Icon'. There are Global variables for each of these indications. To turn off the visibility, uncomment the variable in User Settings and change the value to not show the outline.

The dynamo shows a Status icon and is surrounded by a Colored rectangular outline at run time to indicate that a condition needing attention exists. When there are active alarms, the alarm outline is shown instead.

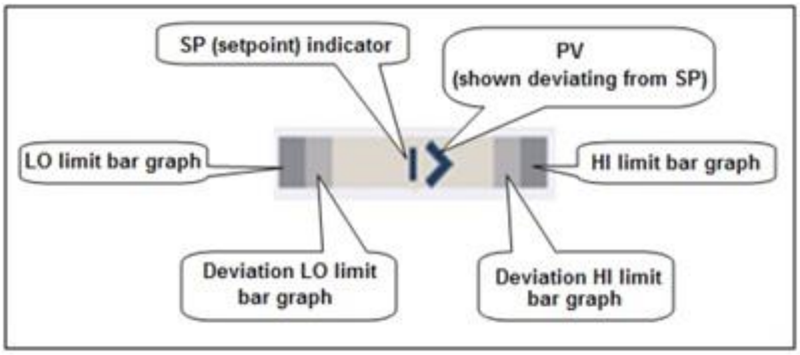
The following describes the Status icons used in the theme dynamos. These icons include: Mode, Module Running State, I/O State, Simulate Condition, Permissive Option, Interlock Option, Interlock and Permissive Option, Acquire Info Option.

PV-SP Deviation (Comparison) Bar Graph

The Deviation Bar Graph presents PID or RATIO information graphically relative to the current SP. This allows Operators to scan a Display and determine which Modules have significant deviations between PV and SP, without needing to read the corresponding numerical values. The deviation bar graph compares the value of PV and the value of SP and Displays the deviation. The perpendicular line representing SP does not move and is always in the center of the bar graph. The greater the deviation, the more visible the PV diamond becomes. The distance between PV and SP on a bar graph always represents the same amount of deviation from that SP, whether PV is above or below SP. Thus a 2% deviation between PV and SP places, SP at the same location on the bar graph, whether PV is above, or below SP. Since SP is fixed in the middle of the bar graph, it is the current SP value that defines the current range of PV shown in the bar graph.

By default, the maximum configured bar graph range uses PV_SCALE. In runtime, the Bar Graph range dynamically adjusts from this maximum, based on the current SP value. This ensures that PV is placed at the same location on the Bar Graph, whether PV is above, or below SP. Thus when SP is set at 50% of range, the PV diamond at the end of the bar would mean that PV was at 0 or 100% of PV_SCALE. If SP is then set to 25% of PV_SCALE, the PV diamond at the end of the bar graph would mean that PV was at 0 or 50%.

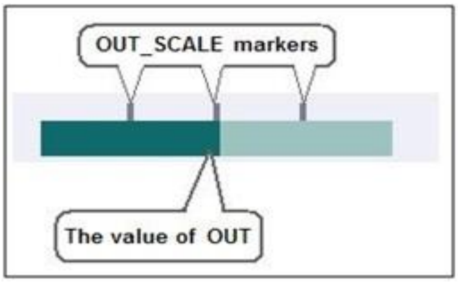
Optionally, you can have a user-defined maximum configured bar graph range. The user-defined percent of scale can be defined as either a number or a path that resolves to percent. If the value is set to 10, it equates to +/- 5% of EU range. so the maximum bar graph range would have zero scale at SP minus 5% of EU range and full scale on the bar graph is SP plus 5% of EU range for a total of 10%. When using a path, limit the value to between 0 and 100. When a path is configured for the range, the range will dynamically adjust in runtime based on the current value of that path.



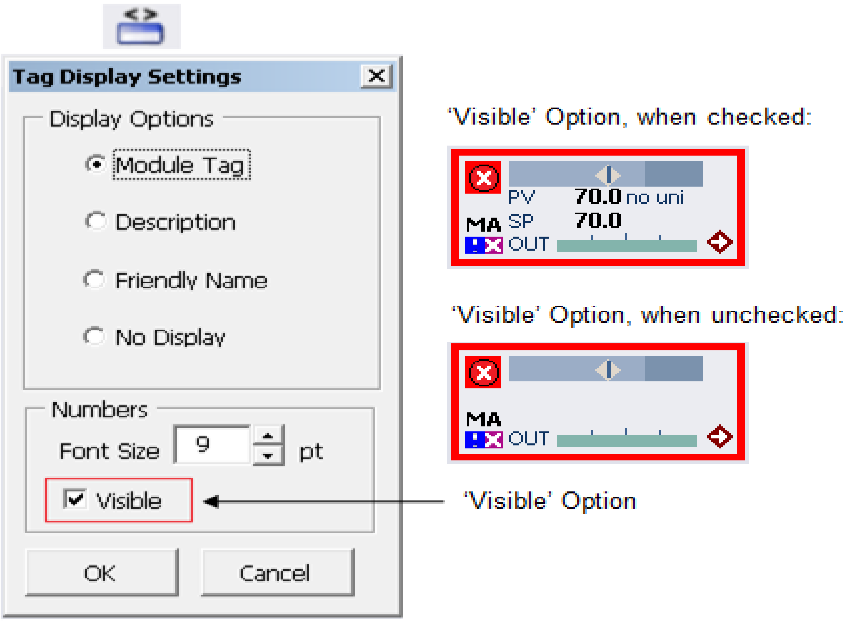
PV-SP Deviation (Comparison) Bar Graph
The deviation bar graph graphically shows the following key information:
Comparison of PV and SP - Shape recognition is used to aid detection of PV deviations. When PV equals SP, only the perpendicular SP line is visible. The greater the deviation, the more visible the PV diamond becomes, starting as an arrow and growing into the diamond shape. Knowing these shapes allows the Operator to scan a Display and quickly know how PV and SP compare.
Pattern recognition to detect the significance of any PV deviations from SP is possible; when multiple deviation bar graphs are viewed together, since the SP indication is fixed.
Comparison of PV and SP, when small deviations are important - the user-defined scaling can be defined such that even a small deviation between PV and SP is very visible and provide this information for any SP value.
Operators are shown PV and SP at the appropriate place in range. By default, the full 0 – 100% PV_SCALE range is shown in the graph. For process values such as temperature or pH where operation is required within a small percentage of the overall range, the bar graph can be defined with a partial range.
PV and SP value relative to Alarm limits - HI, DV_HI, LO and DV_LO Alarm limits are shown on the bar graph. If the PV or SP is near an Alarm limit, the Operator can determine this from the Combination Bar Graph. The indication of these Alarm locations is shown subtly (such as in Grays), providing Alarm limit information without being distracting or creating excessive visual clutter.

OUT Bar Graph

This bar graph indicates the value of the Output (OUT) in the range of OUT_SCALE by filling left to right.



OUT Bar Graph
Visibility of SP/PV parameter on Dynamos
SP/PV visibility on dynamos can be set from "Toggle Display Tag" tool from toolbar. Visible checkbox is provided to set the visibility. Refer to Working with Toolbars for other functionality of toolbar buttons.
This functionality is applicable for dynHDbarInfo, dynHCbarInfo, dynHDbarParam, dynVDbarParam, dynHCbarParam, dynVCbarParam dynamos.



SP/PV Visibility

Types of HCD Dynamos are listed below:

HCD Horizontal Information (dynHInfo)

HCD DCD Provox (dynDCDPVX)

HCD Horizontal Deviationbar Information (dynHDbarInfo)

HCD Horizontal Parameter (dynHParam)

HCD Analog Link (dynAnalogLink)

HCD Discrete Link (dynDiscreteLink)

HCD Level Bar (dynLevelBar)

HCD Discrete Parameter Link (dynDiscreteParLink)

HCD Equipment Module Call Up (dynEMCallUp)

HCD Vertical Combobar Parameter (dynVCbarParam)

HCD Horizontal Combobar Parameter (dynHCbarParam)

HCD Vertical Deviationbar Parameter (dynVDbarParam)

HCD Horizontal Deviationbar Parameter (dynHDbarParam)

HCD Horizontal Combobar Information (dynHCbarInfo)

HCD Horizontal Master-Slave Information (dynHMSInfo)

HCD Equipment Module (dynEM)

HCD Control Valve (dynCtrlVlv)

HCD 3-Way Control Valve (dyn3WCtrlVlv)

HCD 3-Way Valve (dyn3WValve)

HCD Valve (dynVLV)

HCD Valve Two(dynValveTwo)

HCD Hand Valve(dynHandVlv)

HCD Mode (dynMode)

HCD Motor (dynMotor)

HCD Pump (dynPump)

HCD Pump Two (dynPumpTwo)

HCD Agitator (dynAgitator)

HCD Data Entry (dynDataEntry)

HCD Checkbox (dynCheckBox)

HCD Graphic Link (dynGraphicLink)

HCD EM Information (dynEMInfo_1280X1024_)

HCD EM Information (dynEMInfo_1680X1050_)

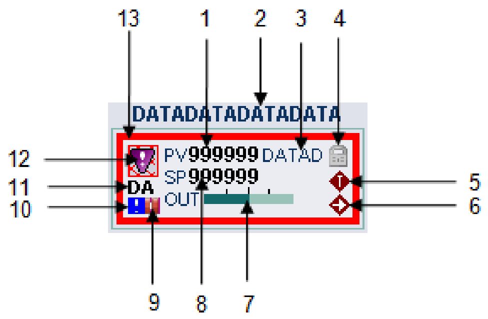
HCD EM Information (dynEMInfo_1920x1080_)

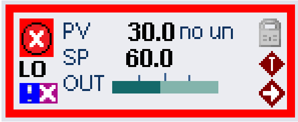
HCD Message Banner (dynMsgBanner_1280X1024_)

HCD Message Banner (dynMsgBanner_1680X1050_)

HCD Message Banner (dynMsgBanner_1920x1080_)

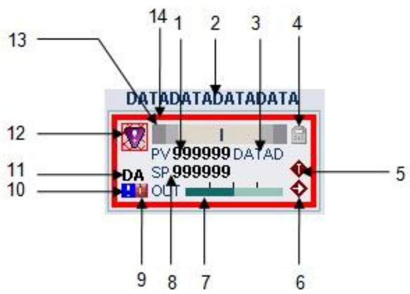
HCD Horizontal Information (dynHInfo)



Run Mode View	Description
	This dynamo provides numeric indication of process value along with its engineering unit and set point value. Out bar is also available to indicate output value.

Sr. No	Description
1	Process Value
2	Display Tag
3	Engineering Unit
4	Lock Info (Modelock)
5	Interlock Info (Analog Tracking)
6	Interlock (Analog Tracking) Bypassed
7	Out Bargrapgh
8	Setpoint
9	Status Icon
10	Abnormal Mode
11	Mode
12	Alarm Icon
13	Alarm/Status Border

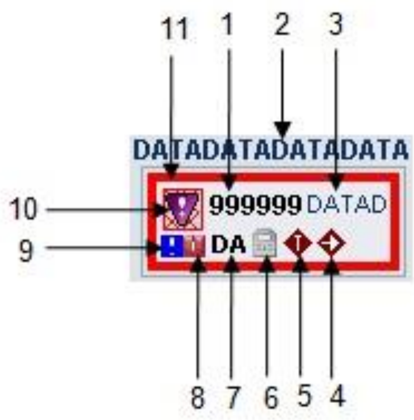
32.2.11. HCD Horizontal Deviation bar Information (dynHDbarInfo)



Sr. No	Description
1	Process Value
2	Display Tag
3	Engineering Unit
4	Lock Info (Mode lock)
5	Interlock Info (Analog Tracking)
6	Interlock (Analog Tracking) Bypassed
7	Out Barograph
8	Setpoint
9	Status Icon
10	Abnormal Mode
11	Mode
12	Alarm Icon
13	Deviation Barograph
14	Alarm/Status Border

Run Mode View	Description
	This dynamo provides indications of PV, SP and alarm limits on the deviation bar. In addition, it also provides numeric indication of process value along with its engineering unit and set point value. Out bar is also available to indicate output value.
	Visibility of process value and set point value can be set from "Toggle Display Tag" tool from Toolbar.

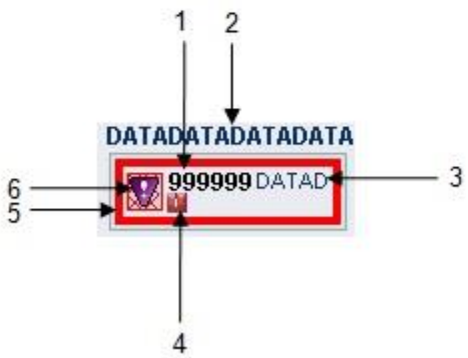
32.2.12. HCD Horizontal Parameter (dynHParam)



Sr. No	Description
1	Process Value
2	Display Tag
3	Engineering Unit
4	Interlock (Analog Tracking) Bypassed
5	Interlock Info (Analog Tracking)
6	Lock Info (Mode lock)
7	Mode
8	Status Icon
9	Abnormal Mode
10	Alarm Icon
11	Alarm/Status Border

Run Mode View	Description
	This dynamo provides numeric indication of process value along with its engineering unit.

32.2.13. HCD Analog Link (dynAnalogLink)

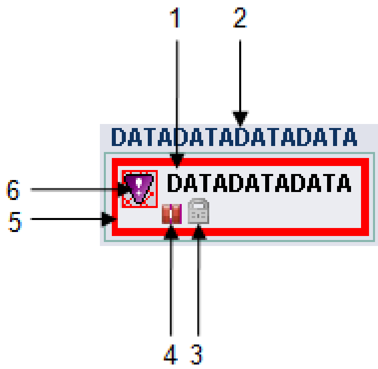


Sr. No	Description
1	Process Value or Any Custom Parameter Value when 'Use Custom Parameter and Scale' is checked on the dynamo form.
2	Display Tag
3	Engineering Unit
4	Status Icon
5	Alarm/Status Border
6	Alarm Icon

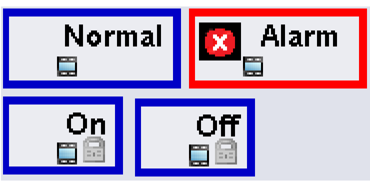
Run Mode View	Description

	This dynamo provides numeric indication of process value or any custom parameter value along with its engineering unit.

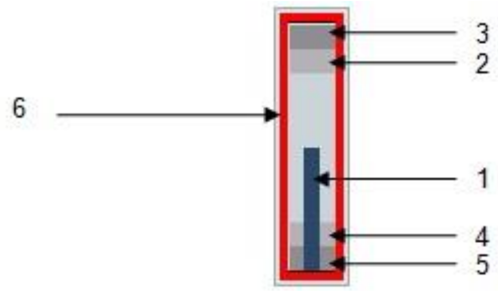
32.2.14. HCD Discrete Link (dynDiscreteLink)



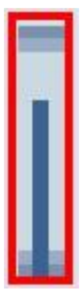
Sr. No	Description
1	Discrete Input or Discrete Output status
2	Display Tag
3	Lock Info (Mode locked), Applicable for DO/DO SOFT module.
4	Status Icon
5	Alarm/Status Border
6	Alarm Icon

Run Mode View	Description
	This dynamo is used for displaying the Discrete Input and Discrete Output status.

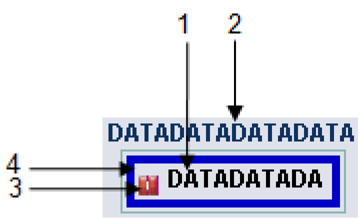
32.2.15. HCD Level Bar (dynLevelBar)



Sr. No	Description
1	Process Value
2	HI Bar
3	HI-HI Bar
4	LO Bar
5	LO-LO Bar
6	Alarm Border

Run Mode View	Description
	Dynamo is used to display an Analog value in a bar graph format. Size can be increased/ decreased using slew keys.

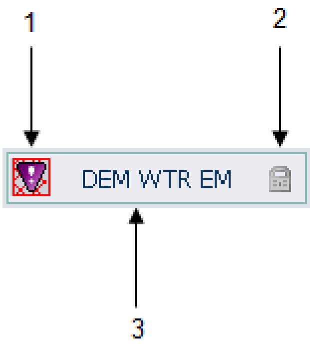
32.2.16. HCD Discrete Parameter Link (dynDiscreteParLink)



Sr. No	Description
1	Status
2	Display Tag
3	Status Icon
4	Status Border

Run Mode View	Description
	Displays ON text and OFF text depending on the discrete value. ON text and OFF text are user configurable.

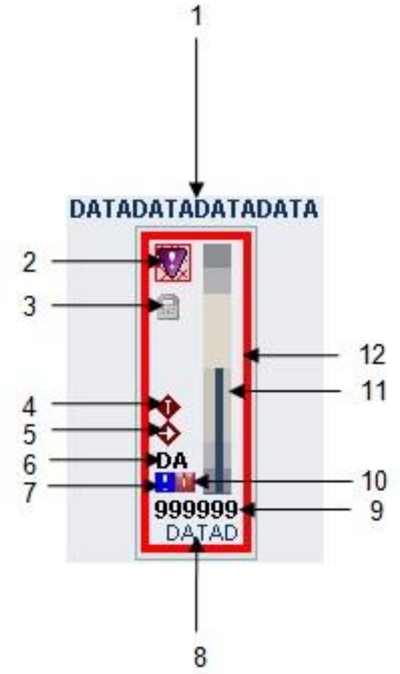
32.2.17. HCD Equipment Module Call Up (dynEMCallUp)



Sr. No	Description
1	Alarm
2	Lock Info (Acquired by a higher level entity)
3	Equipment Module Description

Run Mode View	Description
	Normal

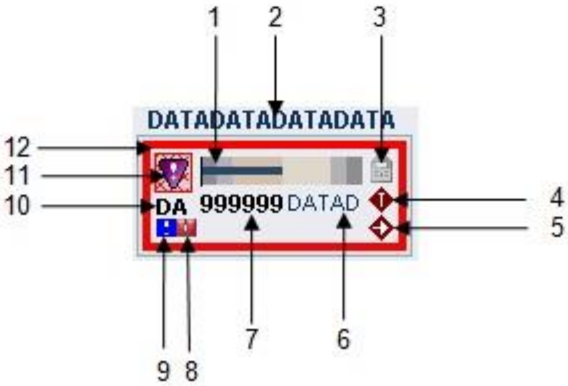
32.2.18. HCD Vertical Combo bar Parameter (dynVCbarParam)



Sr. No	Description
1	Display Tag
2	Alarm Icon
3	Lock Info (Mode lock)
4	Interlock Info (Analog Tracking)
5	Interlock (Analog Tracking) Bypassed
6	Mode
7	Abnormal Mode
8	Engineering Unit
9	Process Value
10	Status Icon
11	Combination (Combo) barograph
12	Alarm/Status Border

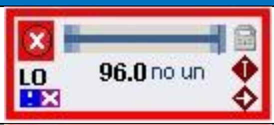
Run Mode View	Description
	This dynamo provides indications of PV, SP and alarm limits on the combo bar. In addition, it also provides numeric indication of process value along with its engineering unit.
	Visibility of process value can be set from "Toggle Display Tag" tool from Toolbar.

32.2.19. HCD Horizontal Combobar Parameter (dynHCbarParam)

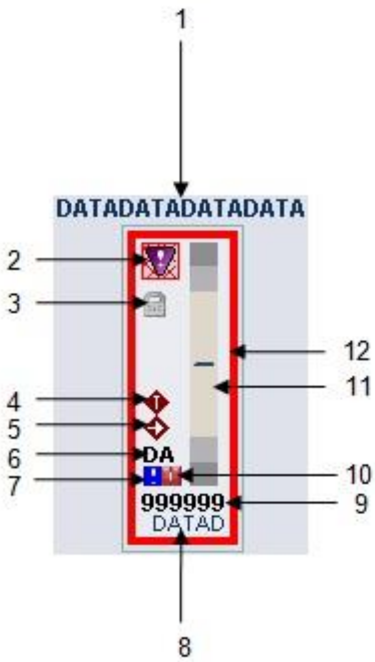


Sr. No	Description
1	Combination (Combo) barograph
2	Display Tag
3	Lock Info (Mode lock)
4	Interlock Info (Analog Tracking)
5	Interlock (Analog Tracking) Bypassed
6	Engineering Unit
7	Process Value
8	Status Icon

9	Abnormal Mode
10	Mode
11	Alarm Icon
12	Alarm/Status Border

Run Mode View	Description
	This dynamo provides indications of PV, SP and alarm limits on the combo bar. In addition, it also provides numeric indication of process value along with its engineering unit.
	Visibility of process value can be set from "Toggle Display Tag" tool from Toolbar.

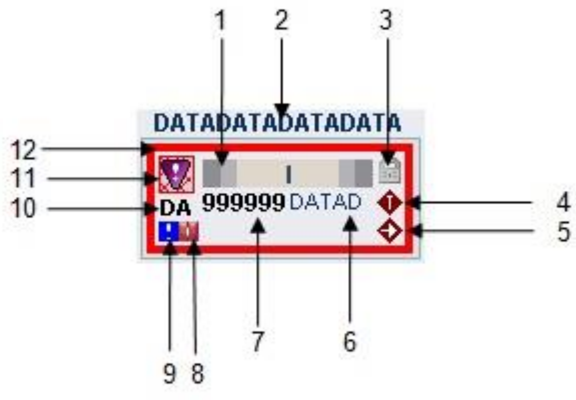
32.2.20. HCD Vertical Deviation bar Parameter (dynVDbarParam)



Sr. No	Description
1	Display Tag
2	Alarm Icon
3	Lock Info (Mode lock)
4	Interlock Info (Analog Tracking)
5	Interlock (Analog Tracking) Bypassed
6	Mode
7	Abnormal Mode
8	Engineering Unit
9	Process Value
10	Status Icon
11	Deviation barograph
12	Alarm/Status Border

Run Mode View	Description
	This dynamo provides indications of PV, SP and alarm limits on the deviation bar. In addition, it also provides numeric indication of process value along with its engineering unit.
	Visibility of process value can be set from "Toggle Display Tag" tool from Toolbar.

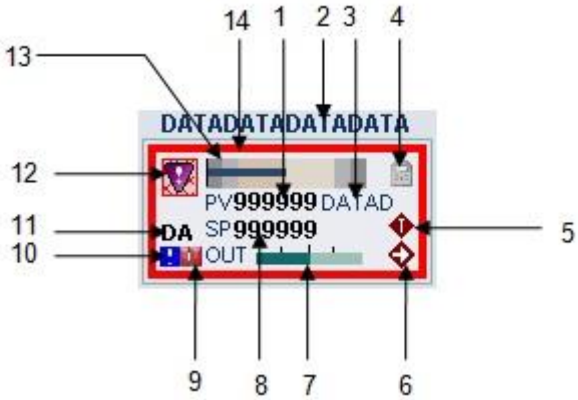
32.2.21. HCD Horizontal Deviation bar Parameter (dynHDbarParam)



Sr. No	Description
1	Deviation barograph
2	Display Tag
3	Lock Info (Mode lock)
4	Interlock Info (Analog Tracking)
5	Interlock (Analog Tracking) Bypassed
6	Engineering Unit
7	Process Value
8	Status Icon
9	Abnormal Mode
10	Mode
11	Alarm Icon
12	Alarm/Status Border

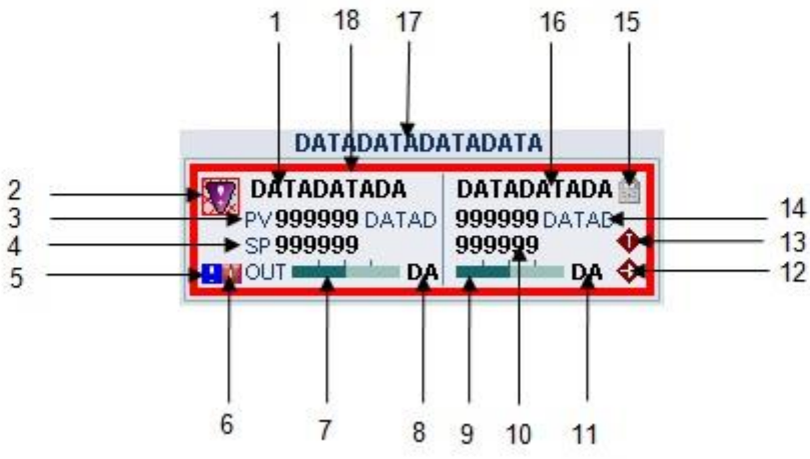
Run Mode View	Description
	This dynamo provides indications of PV, SP and alarm limits on the deviation bar. In addition, it also provides numeric indication of process value along with its engineering unit.
	Visibility of process value can be set from "Toggle Display Tag" tool from Toolbar.

32.2.22. HCD Horizontal Deviationbar Parameter (dynHDbarParam)



Sr. No	Description
1	Process Value
2	Display Tag
3	Engineering Unit
4	Lock Info (Mode lock)
5	Interlock Info (Analog Tracking)
6	Interlock (Analog Tracking) Bypassed
7	Out barograph
8	Setpoint
9	Status Icon
10	Abnormal Mode
11	Mode

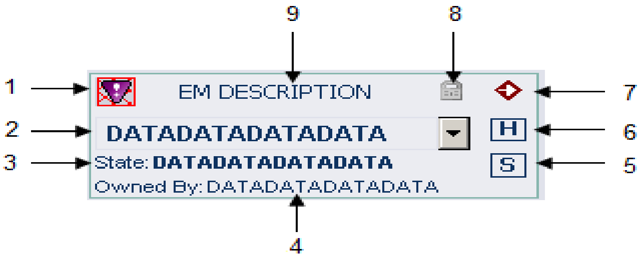
32.2.23. HCD Horizontal Master-Slave Information (dynHMSInfo)



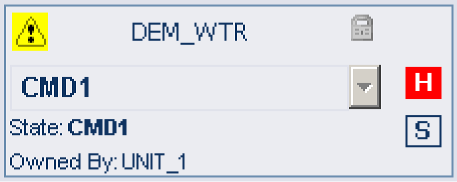
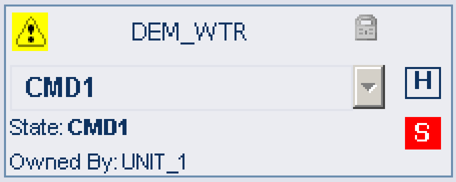
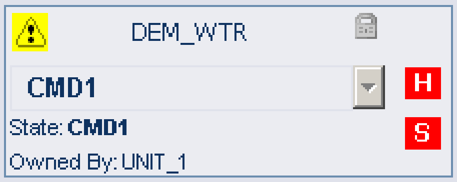
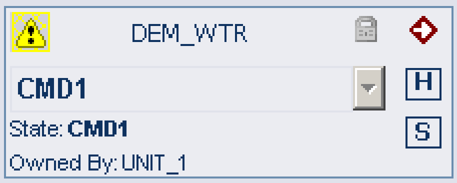
Sr. No	Description
1	Indication for Master
2	Alarm Icon
3	Process Value for Master
4	Setpoint for Master
5	Abnormal Mode
6	Status Icon
7	Out barograph for Master
8	Mode of Master
9	Out barograph for Slave
10	SP Indication for Slave
11	Mode of Slave
12	Interlock (Analog Tracking) Bypassed
13	Interlock Info (Analog Tracking)
14	Process Value for Slave
15	Lock Info (Mode lock)
16	Indication for Slave
17	Display Tag
18	Alarm/Status Border

Run Mode View	Description
	This dynamo provides indications of PV, SP and OUT for Master and Slave PID. In addition, it also provides numeric indication of process value along with its engineering unit and set point value. Out bar is also available to indicate output value.

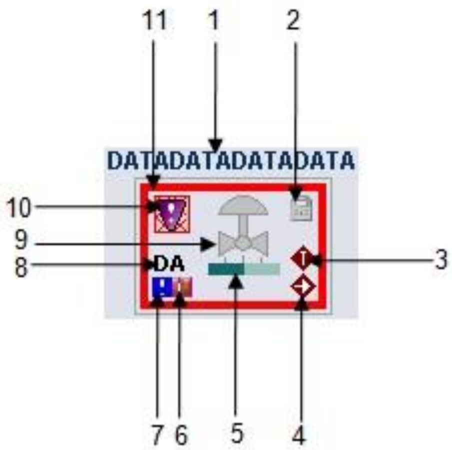
32.2.24. HCD Equipment Module (dynEM)



Sr. No	Description
1	Alarm Icon
2	Equipment Module Command Drop Box
3	Equipment Module state
4	Owner Info
5	Indication of SM condition. On click of this icon opens Sentinel Monitor detail display PCSD_EM_SM_130_DT
6	Indication of SM condition. On click of this icon opens Sentinel Monitor detail display PCSD_EM_HM_130_DT
7	HM / SM bypass indication
8	Lock Info (Acquired by a higher level entity)
9	Equipment Module Description

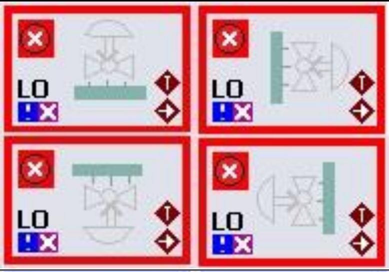
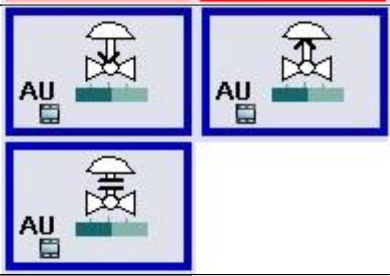
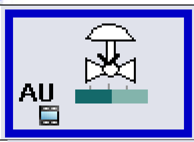
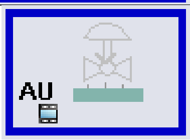
Run Mode View	Description
	HM condition active
	SM condition active
	HM and SM conditions active
	HM and SM conditions bypassed

32.2.25. HCD Control Valve (dynCtrlVlv)

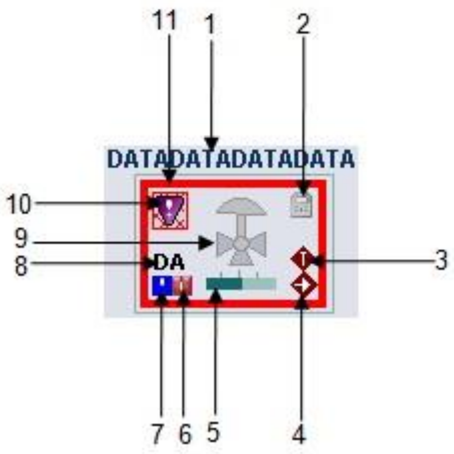


Sr. No	Description
1	Display Tag
2	Lock Info (Mode lock)
3	Interlock Info (Analog Tracking)
4	Interlock (Analog Tracking) Bypassed
5	Output barograph

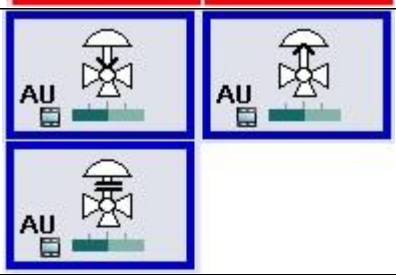
6	Status Icon
7	Abnormal Mode
8	Mode
9	Control Valve Body
10	Alarm Icon
11	Alarm/Status Border

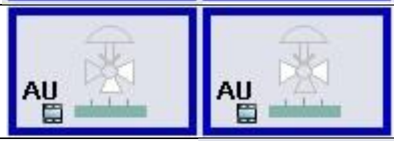
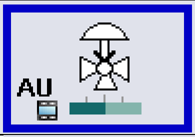
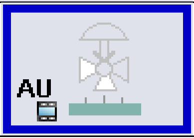
Run Mode View	Description
	Possible Rotation (CCW): 0, 90, 180, 270
	Possible Fail-Safe Options: Fail to Close, Fail to Open, Hold on Failure.
	Valve Open State
	Valve Close State

32.2.26. HCD 3-Way Control Valve (dyn3WCtrlVlv)

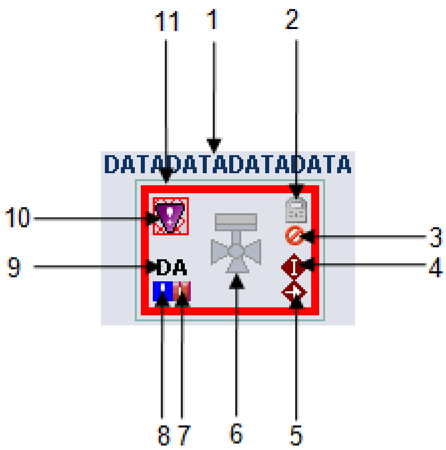


Sr. No	Description
1	Display Tag
2	Lock Info (Modelock)
3	Interlock Info (Analog Tracking)
4	Interlock (Analog Tracking) Bypassed
5	Output Bar
6	Status Icon
7	Abnormal Mode
8	Mode
9	3-Way Control Valve Body
10	Alarm Icon
11	Alarm/Status Border

Run Mode View	Description
	Possible Rotation (CW): 0, 90, 180, 270.
	Possible Fail-Safe Indication Options: Fail to Close, Fail to Open, Hold on Failure.

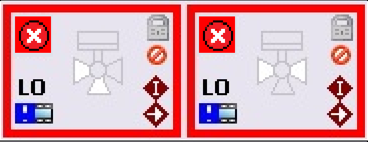
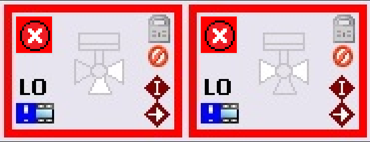
	Possible Normal Positions (Common Port- Left): 1. Closed Port: Right; Open Port: Bottom 2. Closed Port: Bottom; Open Port: Right
	Possible Normal Positions (Common Port- Right): 1. Closed Port: Left; Open Port: Bottom 2. Closed Port: Bottom; Open Port: Left
	Valve Open State
	Valve Close State

32.2.27. HCD 3-Way Valve (dyn3WValve)

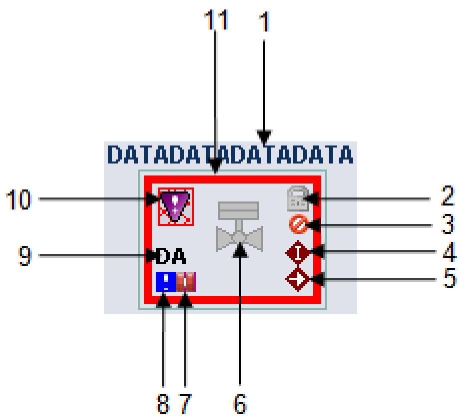


Sr. No	Description
1	Display Tag
2	Lock Info (Mode lock)
3	No Permit Indication
4	Interlock Info
5	Interlock / Permissive / Force Setpoint Bypassed
6	HCD 3-Way Valve Body
7	Status Icon
8	Abnormal Mode

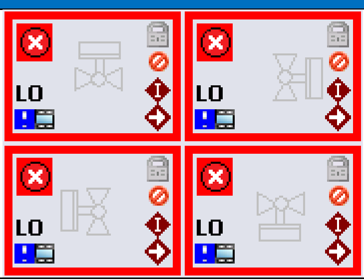
9	Mode
10	Alarm Icon
11	Alarm/Status Border

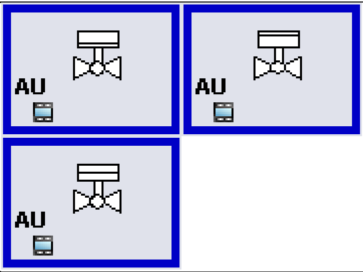
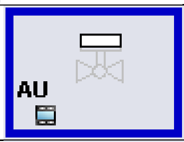
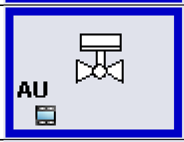
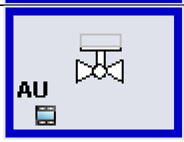
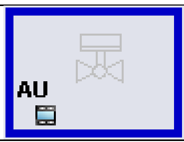
Run Mode View	Description
 State 0 State 1	Valve status indication: For example, 1. Valve Closed: ‘State 0’ 2. Valve Open: ‘State 1’
	Possible Rotation (CW): 0, 90, 180, 270
	Possible Passive State Positions (Common Port- Left): 1. Closed Port: Right; Open Port: Bottom 2. Closed Port: Bottom; Open Port: Right
	Possible Passive State Positions (Common Port- Right): 1. Closed Port: Left; Open Port: Bottom 2. Closed Port: Bottom; Open Port: Left
	Valve with ‘Automatic Valve (Show Actuator)’ option unchecked on dynamo form.

32.2.28. HCD Valve (dynVLV)

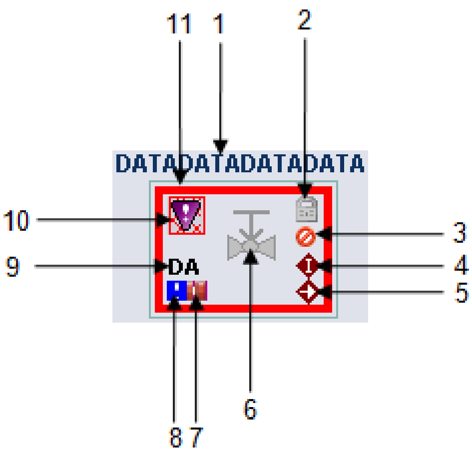


Sr. No	Description
1	Display Tag
2	Lock Info (Mode lock)
3	No Permit Indication
4	Interlock Info
5	Interlock / Permissive / Force Setpoint Bypassed
6	HCD Valve Body
7	Status Icon
8	Abnormal Mode
9	Mode
10	Alarm Icon
11	Alarm/Status Border

Run Mode View	Description
	Possible Rotation (CW): 0, 90, 180, 270

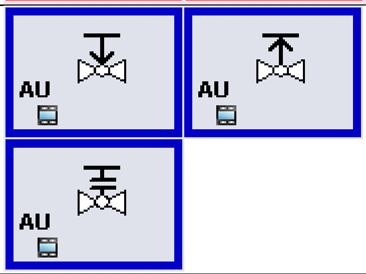
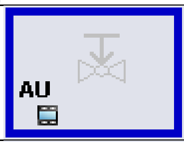
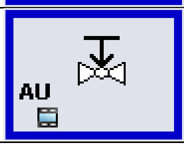
	Failsafe Indications : Fail to Close, Fail to Open, Hold on Failure
	Valve Opening state
	Valve Open State
	Valve Closing State
	Valve Closed State

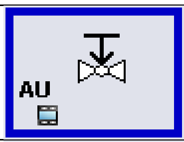
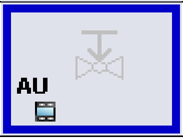
32.2.29. HCD Valve Two (dynValveTwo)



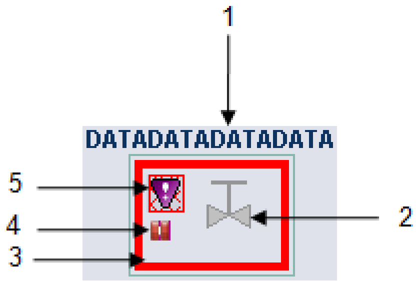
Sr. No	Description
1	Display Tag
2	Lock Info (Modelock)

3	No Permit Indication
4	Interlock Info
5	Interlock / Permissive / Force Setpoint Bypassed
6	HCD Valve Two Body
7	Status Icon
8	Abnormal Mode
9	Mode
10	Alarm Icon
11	Alarm/Status Border

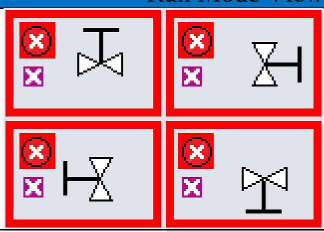
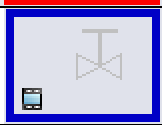
Run Mode View	Description
	Possible Rotation (CW): 0, 90, 180, 270
	Failsafe Indications: Fail to Close, Fail to Open, Hold on Failure
	Valve Opening state
	Valve Open State

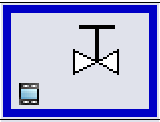
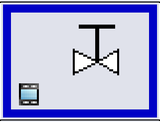
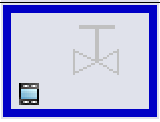
	Valve Closing State
	Valve Close State

32.2.30. HCD Hand Valve (dynHandVLV)



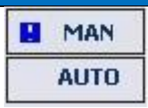
Sr. No	Description
1	Display Tag
2	HCD Hand Valve Body
3	Alarm/Status Border
4	Status Icon
5	Alarm Icon

Run Mode View	Description
	Possible Orientation (CW): 0, 90, 180, 270.
	Valve Opening state

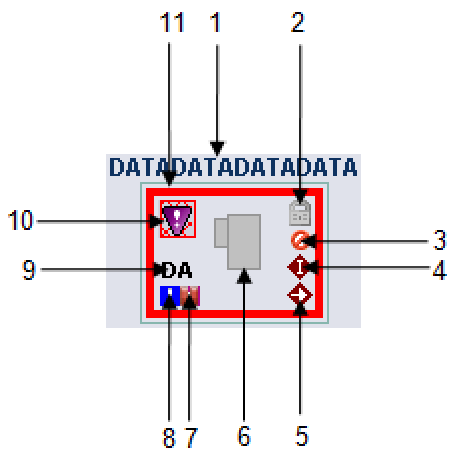
	Valve Open State
	Valve Closing State
	Valve Close State

32.2.31. HCD Mode (dynMode)

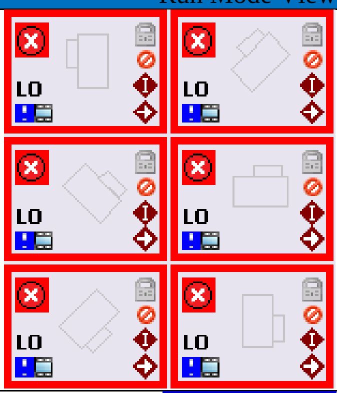
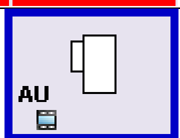
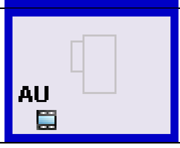
Sr. No	Description
1	Display Tag
2	Control Module Mode
3	Abnormal Mode

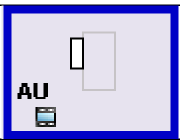
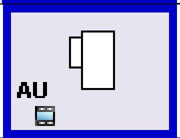
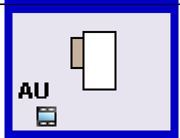
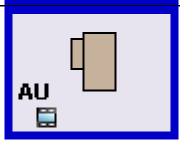
Run Mode View	Description
	It describes the mode of given function block. Abnormal Mode icon is visible either of following is true: Actual mode of the function block is not equal to Normal mode. Actual mode of the function block is not equal to the Target mode.

32.2.32. HCD Motor (dynMotor)

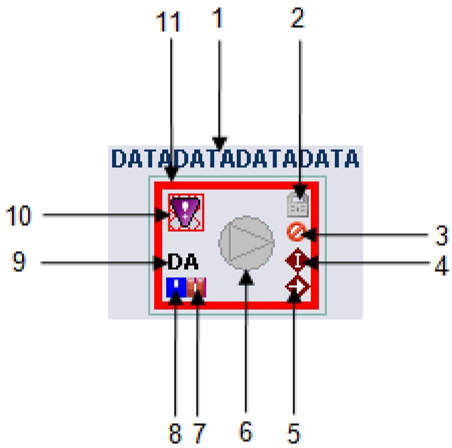


Sr. No	Description
1	Display Tag
2	Lock Info (Mode lock)
3	No Permit Indication
4	Interlock Info
5	Interlock / Permissive / Force Setpoint Bypassed
6	HCD Motor Body
7	Status Icon
8	Abnormal Mode
9	Mode
10	Alarm Icon
11	Alarm/Status Border

Run Mode View	Description
	Possible Rotation (CW): 0, 45, 90, 180, 270, 315.
	Motor Started
	Motor Stopped

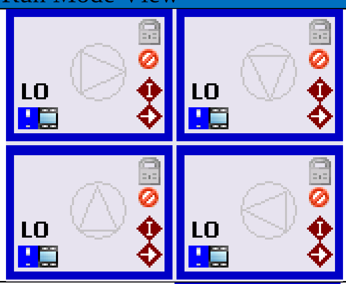
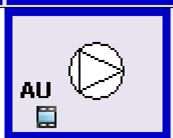
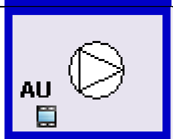
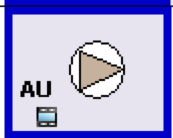
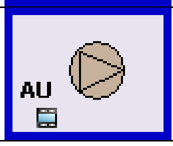
	Motor going forward
	Motor forward
	Motor going reverse
	Motor reverse

32.2.33. HCD Pump (dynPump)

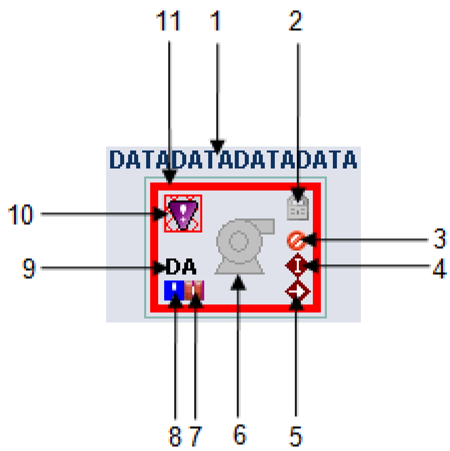


Sr. No	Description
1	Display Tag
2	Lock Info (Modelock)
3	No Permit Indication
4	Interlock Info
5	Interlock / Permissive / Force Setpoint Bypassed
6	HCD Pump Body

7	Status Icon
8	Abnormal Mode
9	Mode
10	Alarm Icon
11	Alarm/Status Border

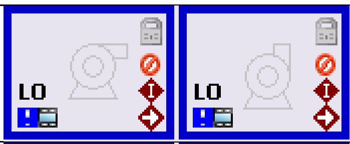
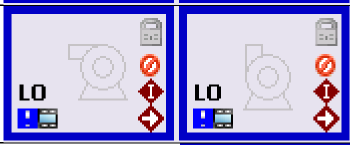
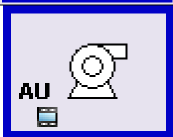
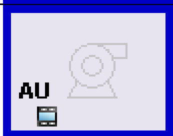
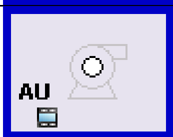
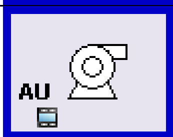
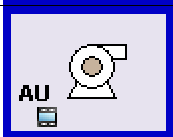
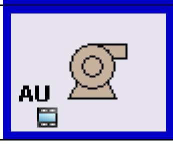
Run Mode View	Description
	Possible Orientations (with Rotate and/or Flip Option): 0, 90, 180, 270
	Pump Started
	Pump Stopped
	Pump going forward
	Pump forward
	Pump going reverse
	Pump reverse

32.2.34. HCD Pump Two (dynPumpTwo)

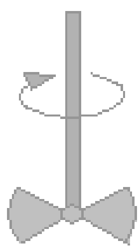


Sr. No	Description
1	Display Tag
2	Lock Info (Mode lock)
3	No Permit Indication
4	Interlock Info
5	Interlock / Permissive / Force Setpoint Bypassed
6	HCD Pump Two Body
7	Status Icon
8	Abnormal Mode
9	Mode
10	Alarm Icon
11	Alarm/Status Border

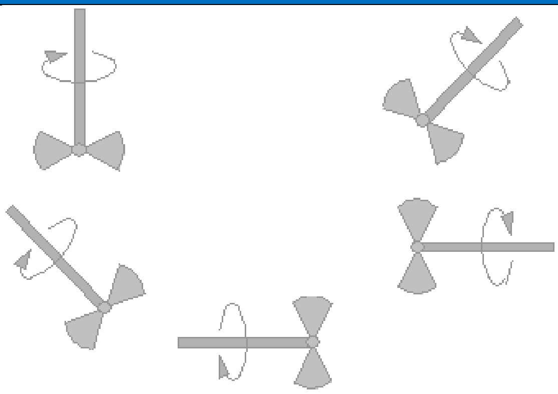
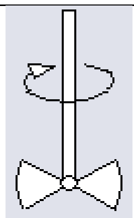
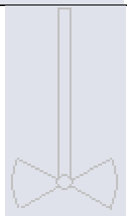
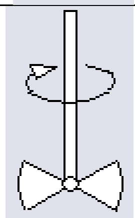
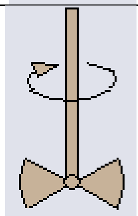
Run Mode View	Description

	Possible Orientations with Rotate Option: 0, 90
	Possible Orientations with Flip Option: 180, 270
	Pump Started
	Pump Stopped
	Pump going forward
	Pump forward
	Pump going reverse
	Pump reverse

32.2.35. HCD Agitator (dynAgitator)

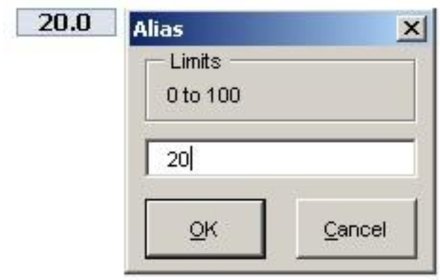


Sr. No	Description
1	A resizable, animated Agitator symbol provides feedback information on device State (State 0, State 1, State 2, Undefined).

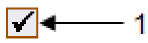
Run Mode View	Description
	Possible Rotation (CCW): 0, 45, 90, 270, 315.
	Agitator Started
	Agitator Stopped
	Agitator Forward / Fast
	Agitator Reverse / Slow

32.2.36. HCD Data Entry (dynDataEntry)

Sr. No	Description
1	Analog/Digital Data Entry

Run Mode View	Description
	Provides the facility of Data Entry for Analog/Discrete parameters.

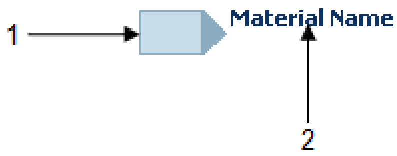
32.2.37. HCD Checkbox (dynCheckBox)

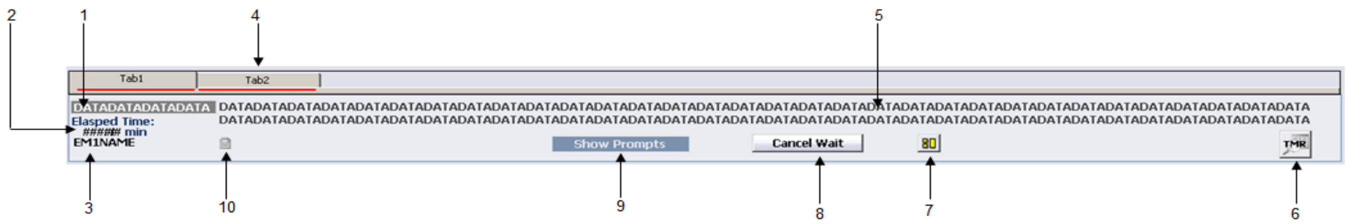


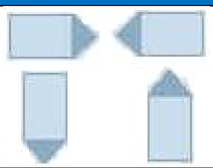
Sr. No	Description
1	On / Off Value

Run Mode View	Description
	Checked Status
	Unchecked Status

32.2.38. HCD Graphic Link (dynGraphicLink)

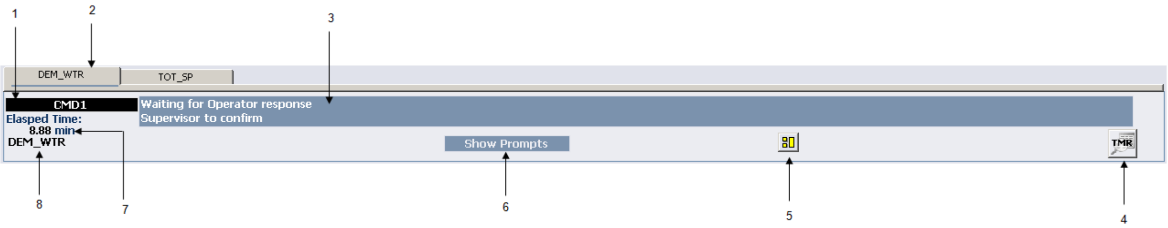




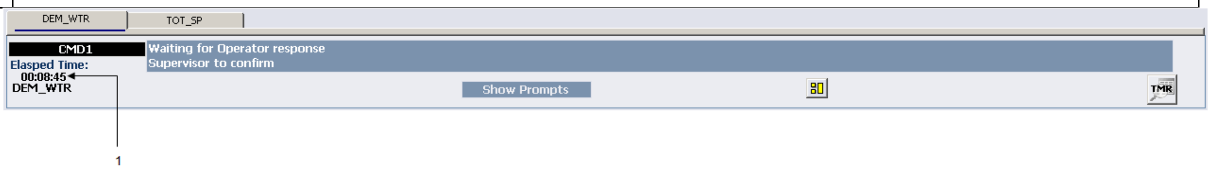
Run Mode View	Description
	Possible orientation of the dynamo (from left ; from right ; from top ; from bottom) Possible text location (left of arrow ; right of arrow ; top of arrow ; bottom of arrow)

32.2.39. HCD EM Information (dynEMInfo_1680X1050_)

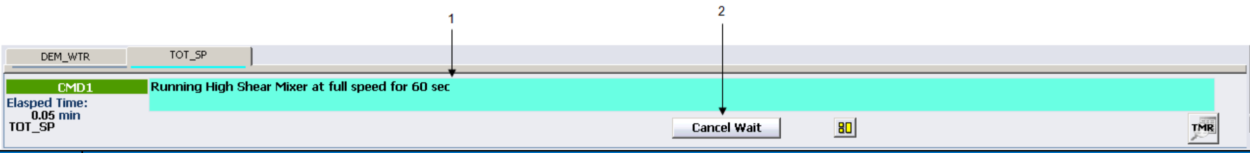
Sr. No	Description
1	Current state of the selected equipment module
2	Time elapsed in the current step of selected equipment module
3	Name of the selected equipment module. Opens the faceplate of the equipment module when clicked.
4	Tab strip for all the equipment modules assigned to this dynamo in order to choose the EM whose messages are to be displayed
5	Displays the message for the operator generated by Equipment module (MSG1 and MSG2 parameters) and supports Multilanguage (optional). The color of this box changes to blue, bright yellow, bright red, bright cyan as per the type of message set in message type parameter.This window also shows Prompt when clicked on 'Show Prompt' button.
6	Click on ‘TMR‘ button opens the detail display of overview timer PCSD_MF_OVRTMR_130_DT, displaying information about individual timers.
7	'Hold' button when clicked allows operator to hold the equipment module
8	“Cancel Wait” button when clicked enables the phase logic to proceed further without waiting for the process condition to be achieved. This is visible only when Cancel Wait is activated by EM.
9	'Show Prompts' button when clicked opens the OAR Prompt Window to allow operator to answer OAR prompt. This is visible only when OAR is activated by EM.
10	Lock Info (Acquired by a higher-level entity and HOLD_REQ is false).



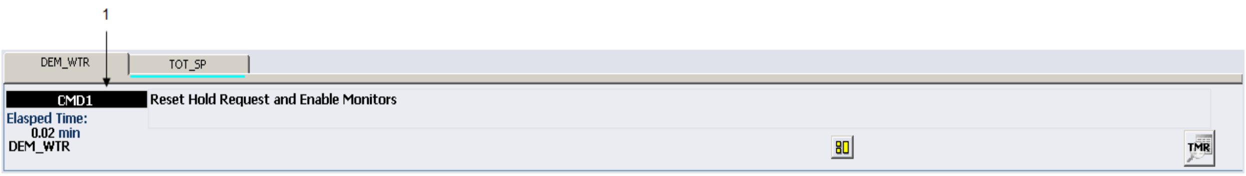
Sr. No	Description
1	Name of the command currently running.
2	More than one equipment module may be configured for the same message bar. This tab strip allows the operator to choose the EM, the messages for which need to be displayed.
3	Equipment module message (MSG1 and MSG2 parameters). The color of this box reflects the type of message (set in parameter MSG_TYPE).
4	Clicking on 'TMR' button opens the details display of overview timers displaying information about individual timers.
5	Operator hold button that allows operator to put THE EQUIPMENT MODULE in hold.
6	Allows operator to answer OAR prompts. This is visible only when an OAR is active.
7	Step time of the currently active step in the command.
8	Opens the faceplate of the equipment module.



Sr. No	Description
1	Step time of the currently active step in the Command in the format DD:HH:MM:SS (Days, Hours, Minutes, Seconds). The step time can be viewed in this format by setting the parameter HH_MM_SS_ENAB in the MONITOR composite to True.



Sr. No	Description
1	Message with cancel wait option. The cyan background color indicates the possibility of cancelling this process specific wait by the user with suitable access i.e., either the supervisor or the operator.
2	'Cancel Wait' button is active when cancel wait is activated by THE EQUIPMENT MODULE code. The "Cancel Wait" button when clicked opens data entry with electronic signature window.



Sr. No	Description
1	Name of the command currently running For command field Background is black and text color is white, irrespective of any state.