SEL-3505
SEL-3505-3

Real-Time Automation Controller

Instruction Manual

20210720

SEL SCHWEITZER ENGINEERING LABORATORIES

PM3505-01

PM3505-01

Table of Contents

This page intentionally left blank

List of Tables

This page intentionally left blank

List of Figures

Preface

Safety Information

⚠CAUTION
To ensure proper safety and operation, the equipment ratings, installation instructions, and operating instructions must be checked before commissioning or maintenance of the equipment. The integrity of any protective conductor connection must be checked before carrying out any other actions. It is the responsibility of the user to ensure that the equipment is installed, operated, and used for its intended function in the manner specified in this manual. If misused, any safety protection provided by the equipment may be impaired.

Dangers, Warnings, and Cautions

This manual uses three kinds of hazard statements, defined as follows:

⚠DANGER
Indicates an imminently hazardous situation that, if not avoided, **will** result in death or serious injury.

⚠WARNING
Indicates a potentially hazardous situation that, if not avoided, **could** result in death or serious injury.

⚠CAUTION
Indicates a potentially hazardous situation that, if not avoided, **may** result in minor or moderate injury or equipment damage.

Safety Symbols

The following symbols are often marked on SEL products.

	**⚠CAUTION** Refer to accompanying documents.	**⚠ATTENTION** Se reporter à la documentation.
	Earth (ground)	Terre
	Protective earth (ground)	Terre de protection
	Direct current	Courant continu
	Alternating current	Courant alternatif
	Both direct and alternating current	Courant continu et alternatif
	Instruction manual	Manuel d’instructions

Safety Marks

The following statements apply to this device.

General Safety Marks

⚠CAUTION There is danger of explosion if the battery is incorrectly replaced. Replace only with Rayovac no. BR1632 or equivalent recommended by manufacturer. See Owner's Manual for safety instructions. The battery used in this device may present a fire or chemical burn hazard if mistreated. Do not recharge, disassemble, heat above 100°C or incinerate. Dispose of used batteries according to the manufacturer's instructions. Keep battery out of reach of children.	⚠ATTENTION Une pile remplacée incorrectement pose des risques d'explosion. Remplacez seulement avec un Rayovac no BR1632 ou un produit équivalent recommandé par le fabricant. Voir le guide d'utilisateur pour les instructions de sécurité. La pile utilisée dans cet appareil peut présenter un risque d'incendie ou de brûlure chimique si vous en faites mauvais usage. Ne pas recharger, démonter, chauffer à plus de 100°C ou incinérer. Éliminez les vieilles piles suivant les instructions du fabricant. Gardez la pile hors de la portée des enfants.
For use in Pollution Degree 2 environment.	Pour l'utilisation dans un environnement de Degré de Pollution 2.
Ambient air temperature shall not exceed 40°C (104°F).	La température de l'air ambiant ne doit pas dépasser 40°C (104°F).
Tightening Torque Terminal Blocks: 0.8 Nm (7 in-lb)	Couple de serrage Borniers : 0,8 Nm (7 livres-pouce)

Other Safety Marks

⚠DANGER Disconnect or de-energize all external connections before opening this device. Contact with hazardous voltages and currents inside this device can cause electrical shock resulting in injury or death.	⚠DANGER Débrancher tous les raccordements externes avant d'ouvrir cet appareil. Tout contact avec des tensions ou courants internes à l'appareil peut causer un choc électrique pouvant entraîner des blessures ou la mort.
⚠DANGER Contact with instrument terminals can cause electrical shock that can result in injury or death.	⚠DANGER Tout contact avec les bornes de l'appareil peut causer un choc électrique pouvant entraîner des blessures ou la mort.
⚠WARNING Use of this equipment in a manner other than specified in this manual can impair operator safety safeguards provided by this equipment.	⚠AVERTISSEMENT L'utilisation de cet appareil suivant des procédures différentes de celles indiquées dans ce manuel peut désarmer les dispositifs de protection d'opérateur normalement actifs sur cet équipement.
⚠WARNING Have only qualified personnel service this equipment. If you are not qualified to service this equipment, you can injure yourself or others, or cause equipment damage.	⚠AVERTISSEMENT Seules des personnes qualifiées peuvent travailler sur cet appareil. Si vous n'êtes pas qualifiés pour ce travail, vous pourriez vous blesser avec d'autres personnes ou endommager l'équipement.
⚠CAUTION Equipment components are sensitive to electrostatic discharge (ESD). Undetectable permanent damage can result if you do not use proper ESD procedures. Ground yourself, your work surface, and this equipment before removing any cover from this equipment. If your facility is not equipped to work with these components, contact SEL about returning this device and related SEL equipment for service.	⚠ATTENTION Les composants de cet équipement sont sensibles aux décharges électrostatiques (DES). Des dommages permanents non-décelables peuvent résulter de l'absence de précautions contre les DES. Raccordez-vous correctement à la terre, ainsi que la surface de travail et l'appareil avant d'en retirer un panneau. Si vous n'êtes pas équipés pour travailler avec ce type de composants, contacter SEL afin de retourner l'appareil pour un service en usine.
⚠CAUTION The device contains components sensitive to Electrostatic Discharge (ESD). When working on the device with the front panel removed, work surfaces and personnel must be properly grounded or equipment damage may result.	⚠ATTENTION Cet appareil contient des pièces sensibles aux décharges électrostatiques. Quand on travaille sur le appareil avec les panneaux avant ou du dessus enlevés, toutes les surfaces et le personnel doivent être mis à la terre convenablement pour éviter les dommages à l'équipement.
⚠CAUTION Looking into optical connections, fiber ends, or bulkhead connections can result in hazardous radiation exposure.	⚠ATTENTION Regarder vers les connecteurs optiques, les extrémités des fibres oules connecteurs de cloison peut entraîner une exposition à des rayonnements dangereux.
⚠CAUTION Use of controls or adjustments, or performance of procedures other than those specified herein, may result in hazardous radiation exposure.	⚠ATTENTION L'utilisation de commandes ou de réglages, ou l'application de tests de fonctionnement différents de ceux décrits ci-après peuvent entraîner l'exposition à des radiations dangereuses.

General Information

LED Emitter

⚠CAUTION
Use of controls, adjustments, or performance of procedures other than those specified herein may result in hazardous radiation exposure.

⚠CAUTION
Looking into optical connections, fiber ends, or bulkhead connections can result in hazardous radiation exposure.

The SEL-3505 Real-Time Automation Controller (RTAC) is a Class 1 LED Product and complies with IEC 60825-1:1993 + A1:1997 + A2:2001.

The following tables shows LED information specific to the SEL-3505 Real-Time Automation Controller for ETH 1 and ETH 2, the ports that optionally use LED transmitters.

LED Information (Multimode Option)

Item	Detail
Mode	Multimode (62.5 µm fiber)
Wavelength	1300 nm
Source	LED
Connector type	LC
Output power	–14 to –19 dBm

LED Information (Single-Mode Option)

Item	Detail
Mode	Single-mode (9 µm fiber)
Wavelength	1300 nm
Source	Class 1 LASER
Connector type	LC
Output power	–10 to –15 dBm

LED Safety Warnings and Precautions

- Do not look into the end of an optical cable connected to an optical output.
- Do not look into the fiber ports/connectors.
- Do not perform any procedures or adjustments that are not described in this manual.
- During installation, maintenance, or testing of the optical ports only use test equipment classified as Class 1 laser products.
- Incorporated components such as transceivers and laser/LED emitters are not user serviceable. Units must be returned to SEL for repair or replacement.

Environmental Conditions and Voltage Information

The following table lists important environmental and voltage information.

Condition	Range/Description
Indoor/outdoor use	Indoor
Altitude	Up to 2000 m
Relative humidity	5 to 95%
Overvoltage	Category II
Pollution	Degree 2
Insulation class	Class I equipment
Atmospheric pressure	80 to 110 kPa

Instructions for Cleaning and Decontamination

Use care when cleaning the SEL-3505 RTAC. Use a mild soap or detergent solution and a damp cloth to clean the chassis. Do not use abrasive materials, polishing compounds, or harsh chemical solvents (such as xylene or acetone) on any surface.

Copyrighted Software

The software included in this product may contain copyrighted software licensed under terms that give you the opportunity to receive source code. You may obtain the applicable source code from SEL by sending a request to:

Legal Department
GPL Compliance
Schweitzer Engineering Laboratories, Inc.
One Schweitzer Drive
Pullman, WA 99163

Please include your return address, product number, and firmware revision.

Technical Support

We appreciate your interest in SEL products and services. If you have questions or comments, please contact us at:

Schweitzer Engineering Laboratories, Inc.
2350 NE Hopkins Court
Pullman, WA 99163-5603 U.S.A.
Tel: +1.509.338.3838
Fax: +1.509.332.7990
Internet: selinc.com/support
Email: info@selinc.com

⚠WARNING
Use of this equipment in a manner other than specified in this manual can impair operator safety safeguards provided by this equipment.

Section 1

Introduction and Specifications

Overview

As a member of the SEL-3530 Real-Time Automation Controller (RTAC) family, the compact and low power SEL-3505 RTAC is a powerful automation platform that combines the best features of an embedded real-time operating system, and secure communications with flexible, feature-rich IEC 61131 compliant programmability. The SEL-3505 can provide any degree of functionality from that of a simple intelligent port switch to the sophisticated communication and data handling required for advanced integration projects. The SEL-3505 features secure communications, advanced data concentration, high-speed logic processing, flexible engineering access, and protocol conversion capabilities between multiple built-in client/server protocols. The SEL-3505 gives the integrator the necessary tools to easily integrate and concentrate information from the wide variety of microprocessor-based devices found in today's substations. The SEL-3505 is also bundled with intuitive software to quickly configure large automated systems. The SEL-3505-3 provides three serial ports instead of four, eight digital inputs and three digital outputs. In this manual, unless otherwise specified, the SEL-3505 and SEL-3505-3 are both referred to as SEL-3505, SEL-3505 RTAC, or just RTAC.

Features

Physical

SEL-3505 Standard	
Ethernet	Two rear-panel ports (RJ45, LC fiber option)
Device USB	One front panel
IRIG-B Output	Demodulated IRIG-B through rear-panel (DB-9) ports
Light Sensor	One front-panel light sensor
Accelerometer	Internal 3-Axis
SEL-3505 Specific	
EIA-232 Serial	Two rear-panel (DB-9) ports
EIA-232/EIA-422/ EIA-485 Serial	Two rear-panel (DB-9) ports
IRIG-B Input	One front-panel compression connector for demodulated IRIG-B
Digital Input	One rear panel
Digital Output	One solid-state rear panel
SEL-3505-3 Specific	
EIA-232/EIA-422/ EIA-485 Serial	Two rear-panel, one front-panel (DB-9) ports
IRIG-B Input	One rear-panel compression connector for demodulated IRIG-B
Digital Input	Eight front panel
Digital Output	Three electromechanical rear panel

Processing System

Embedded Real-Time Operating System
The embedded SEL Linux® operating system provides exceptional speed, flexibility, and functionality along with increased security.
Watchdog Timer
A separate watchdog microcontroller system provides an extra level of computer system reliability. The microcontroller will activate an alarm and halt all input/output activity if there is a problem with the IEC 61131 logic engine.
Diagnostics and Logging
Manage users, view logs, and check report diagnostics via the built-in web server. Access the historical database for diagnostics and logging. Use open database connectivity (ODBC) to view security and diagnostics logs in off-the-shelf ODBC compliant programs.

Security

Malware Protection

exe-GUARD® anti-malware technology protects the RTAC in two ways:

1. Only authorized tasks are allowed to run on the system.
2. Mandatory Access Control restricts privileges of programs and services to the absolute minimum required to function on the system.

There is no need for virus definition files because only whitelisted or preapproved tasks are allowed to run on the RTAC. SEL Whitelist operates at the core of the RTAC operating system, where it intercepts every program before it is executed. Using advanced cryptographic algorithms, Whitelist inspects a program's binary image before it is allowed to execute, verifying its legitimacy and integrity against a known digital signature. As defined by user configuration, unauthorized changes to the system result in either an alarm indication or device reset to factory defaults. Similarly, any modification or replacement of an existing firmware binary is also revealed by the digital signature verification process and results in the same denial of execution and security response.

Mandatory Access Control fine-tunes the system security policy so that programs and services are constrained to the absolute minimum access privileges required to function. Defining this minimal privilege set at design time ensures firmware processes can be locked to their minimal scope of influence in the system.

User Authentication

The RTAC authenticates user role-based local accounts through Active Directory using Lightweight Directory Access Protocol (LDAP).

Automation and Control

Industry Standard (IEC 61131) Logic Processing

The RTAC includes the IEC 61131 programming environment with standard, custom, and Ethernet libraries for logic processing and scaling of collected data. Two configurable processing cycles give flexibility to run high-speed automation tasks at speeds as fast as four milliseconds while SCADA and other lower-speed tasks can run at a slower rate.

Data Concentration and Protocol Conversion

Collect data from attached devices and serve to other devices via popular industry-standard and SEL protocols.

System Configuration and Maintenance

Configure the RTAC network interface, user account, and security settings over the network via the web interface. Next, configure protocol communications, program IEC 61131 logic, and send project settings with ACSELERATOR RTAC® SEL-5033 Software. After sending project settings, view and force data values to test user logic and data-mapping schemes.

Time Synchronization

Time Synchronization for Connected Devices

The SEL-3505 provides a demodulated IRIG-B output signal to connected intelligent electronic devices (IEDs) via all EIA-232/EIA-422/EIA-485 ports.

IRIG-B Generation

In the absence of an external IRIG-B signal, the SEL-3505 generates a demodulated IRIG-B signal. If desired, you can synchronize the system clock with a network time server (via SNTP, NTP, or PTP) and output SNTP or NTP from the RTAC to other devices on the network.

Models, Options, and Accessories

Models

This manual does not provide complete ordering information. For complete information, see the latest SEL-3505 and SEL-3505-3 Model Option Tables (MOT) at selinc.com, under Literature, Model Option Table.

Options

Communications Options (Ethernet)		
Two Ethernet ports (rear), 10/100BASE-T copper Ethernet port		
Two Ethernet ports (rear), 100BASE-FX fiber-optic Ethernet port		
Two Ethernet ports (rear), one 10/100BASE-T copper, one 100BASE-FX fiber-optic		
Two Ethernet ports (rear), 100BASE-LX10 single-mode fiber-optic Ethernet port		
Two Ethernet ports (rear), one 10/100BASE-T copper, one 100BASE-LX10 single-mode fiber-optic		
SEL-3505 Communications Options		
Modem (56 kbps internal dial-up)		
Mounting		
Surface		
DIN		
Power Supply		
12–24 Vdc		
24–48 Vdc		
SEL-3505-3 Digital Input Rating		
Four input ranges:		
12 V:	ON: 9.6–18 Vdc	OFF: <7.2 Vdc
24 V:	ON: 19.2–28.8 Vdc	OFF: <11 Vdc
48 V:	ON: 38.4–52.8 Vdc	OFF: <28.8 Vdc
125 V:	ON: 100–135.5 Vdc	OFF: <75 Vdc
Firmware Features		
IEC 61850 GOOSE		
IEC 61850 MMS server		
FileIO library		
SVP library with modal analysis		
Environment		
Conformal coating for chemically harsh and high-moisture environments		

Accessories

For all SEL-3505 mounting accessories, including adapter plates, visit selinc.com/products/accessories. Contact your Technical Service Center or the SEL factory for additional details and ordering information for all other accessories.

Table 1.1 Optional Accessories

Product	Description
SEL-2401	Precise Timing Source—Satellite-Synchronized Clock
SEL-3010	Automatic Voice Notification—Event Messenger
SEL-3025	Secure Communications—Serial Shield
SEL-2925	Secure Communications—Wireless Encrypting Transceiver
SEL-2810	200 µm Fiber-Optic Transceiver with IRIG-B
SEL-2812	Multimode Fiber-Optic Transceiver with IRIG-B
SEL-2814	Multimode Fiber-Optic Transceiver with hardware handshaking
SEL-2815	Multimode Fiber-Optic Transceiver—15 km
SEL-2829	Single-Mode Fiber-Optic Transceiver—23 km
SEL-2830	Single-Mode Fiber-Optic Transceiver—80 km
SEL-2831	Single-Mode Fiber-Optic Transceiver—110 km

Applications

The functions of the RTAC make it extremely versatile and powerful. You can combine basic functions of the RTAC to meet the requirements of your application. Several of the most popular applications are listed below and shown later in this section.

Table 1.2 Popular RTAC Functions

Applications	Description
Port Switch	Connects a single serial port to multiple serial ports.
Device Server Port Server Network Gateway	Connects a single network port to multiple serial ports.
Protocol Translator Protocol Gateway	Connects multiple systems that use different protocols.
PAC PLC Logic Processor	Monitors digital and analog inputs to transmit to a central location plus performs IEC 61131-3 logic.
RTU SCADA Data Concentrator	Monitors digital and analog inputs to transmit to a central location with standard SCADA protocols.
Time Synchronization	Selects the best time from different sources and provides IRIG-B and/or NTP output to connected devices.
Security Gateway	Provides firewall protection to incoming Ethernet communications as well as encryption for individual Ethernet sessions.
Synchrophasor Processor	Connects phasor measurement unit (PMU) data to other devices and processes through protocol conversion.

Intelligent Port Switch

Flexible communications parameters make the RTAC a great choice for almost any port switching application. Although RTAC multitasking/multiuser and data handling capabilities make it a very powerful remote automation platform, it is still an economical choice for port switching applications. The time-synchronization capabilities of the RTAC add to its value in this application. An example of this application is shown in *Figure 1.1*.

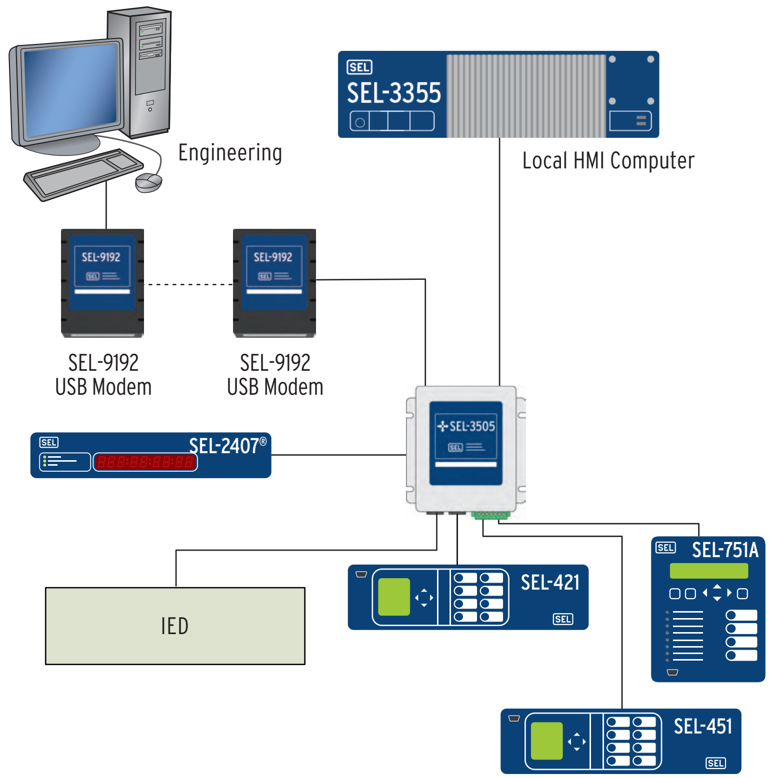


Figure 1.1 Engineering Communication Through a Serial Access Point

Network Gateway

The SEL-3505 has two rear Ethernet ports. This allows the SEL-3505 to make serially connected devices available to high-speed networks through virtual terminal connection ports. For example, Ethernet users can establish secure engineering access Telnet sessions to the SEL-3505 and communicate with intelligent electronic devices (IED) connected to the SEL-3505 serial ports. See *Figure 1.2* for an example of this application.

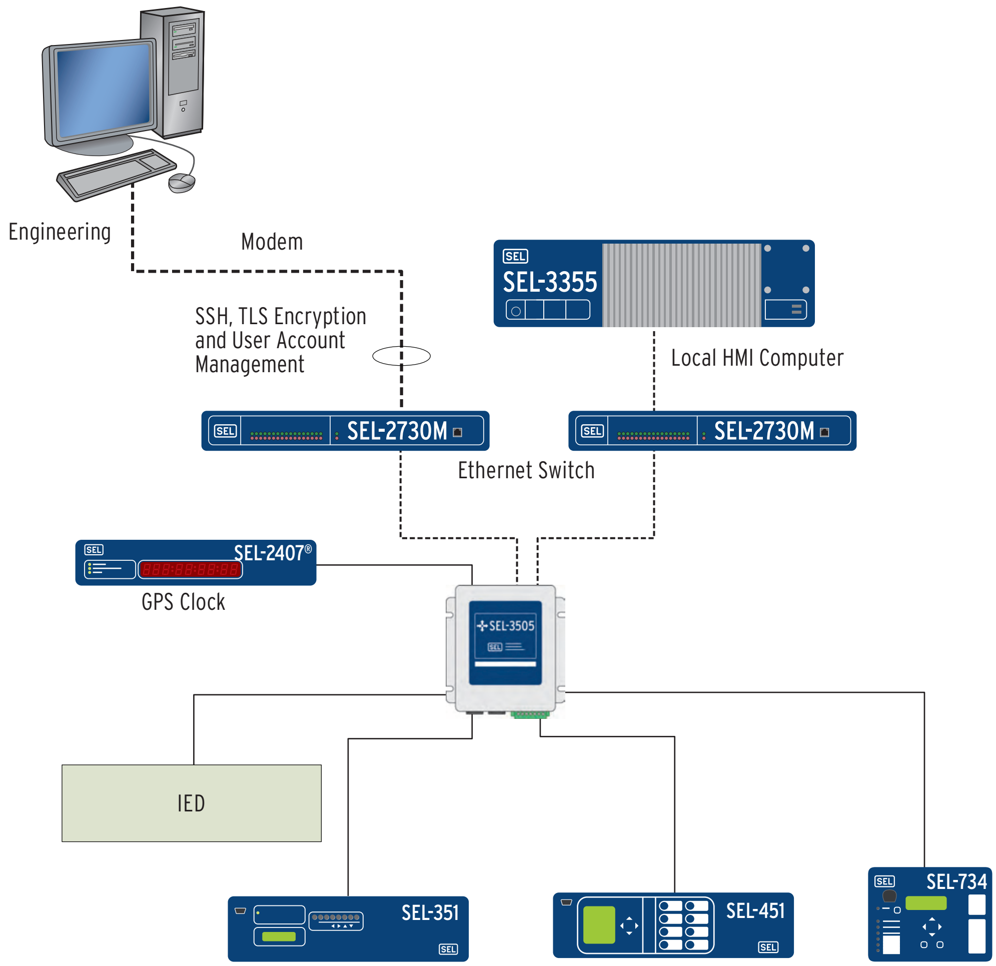


Figure 1.2 Engineering Communication Through a Network Access Point

Protocol Gateway

Collect downstream data with client protocols and send these data to your upstream HMI, RTU or SCADA master with server protocols, converting the data from one protocol to another in the process. RTAC multitasking/multiuser and data handling capabilities make it a great choice for data concentration. An example of this application is shown in *Figure 1.3*.

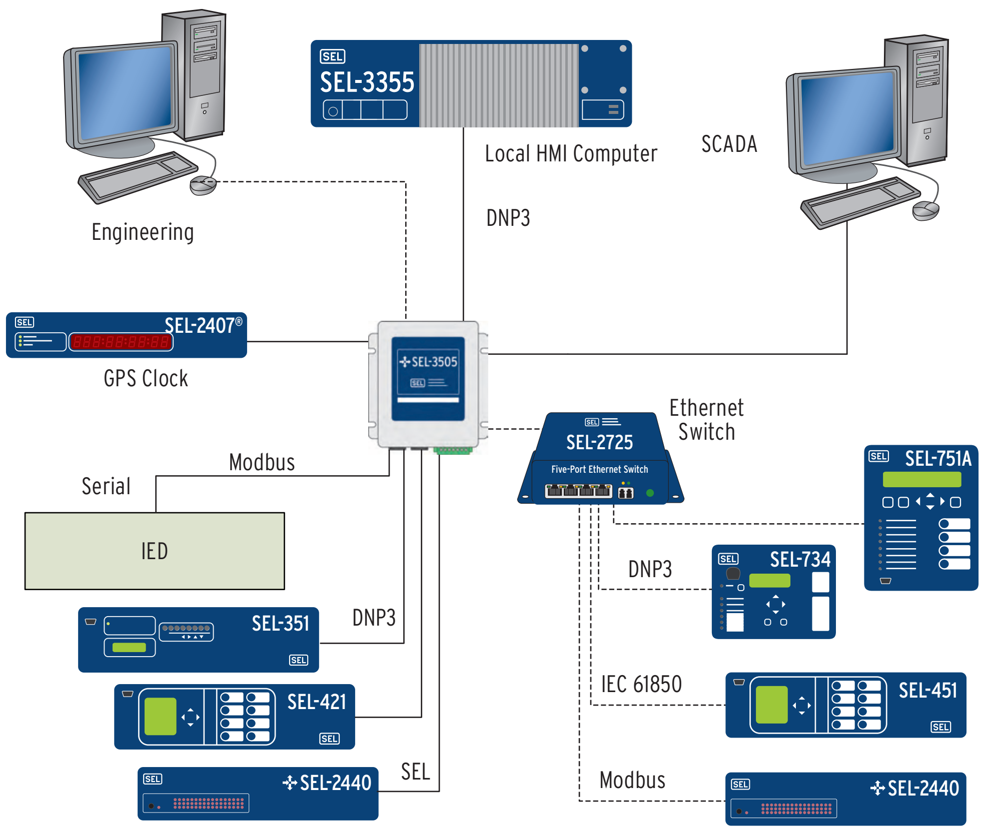


Figure 1.3 Protocol Conversion

Time-Synchronization Source

Synchronize the time clocks in attached devices that accept a demodulated IRIG-B time signal. The demodulated IRIG-B signal is regenerated in the RTAC from an external demodulated source, such as a GPS satellite clock receiver, SNTP/NTP/PTP source, or serial or Ethernet protocol such as DNP3. If an external clock source is not available, the RTAC generates an IRIG-B signal from its internal clock, allowing device synchronization to a common clock for improved sequence-of-events analysis. An example of this application is shown in *Figure 1.4*.

To determine how many devices an IRIG-B output is able to synchronize, a parallel resistance calculation is required because the input resistance of connected devices significantly affects this number. The Specifications section of each RTAC product provides the output drive capability. If the RTAC output resistance is greater than or equal to the parallel resistance calculation of connected devices, all devices can be synchronized by the RTAC.

For example, most SEL relays (excluding SEL-100 and SEL-200 Series Relays) have an IRIG-B input resistance of approximately 2500 ohms. For a BNC demodulated IRIG-B output that supports a drive capacity of 25 ohms, this allows approximately 20 SEL relays to be connected in parallel and be time-synchronized. For most serial port IRIG-B outputs, the drive resistance capability is approximately 500 ohms. This allows 5 SEL relays with an IRIG-B input resistance of approximately 2500 ohms to be connected in parallel.

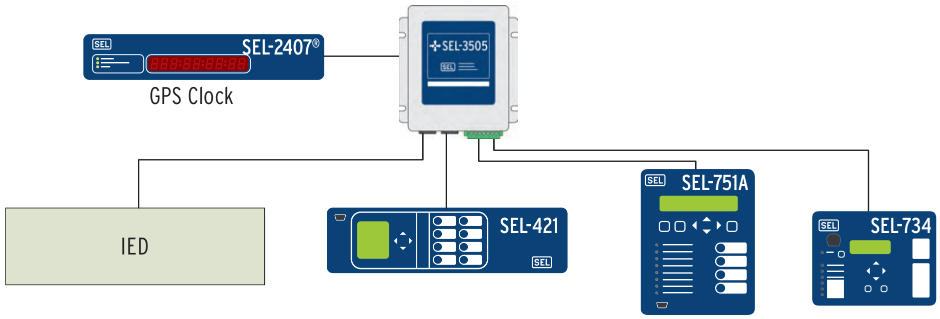


Figure 1.4 Distribute Time With Serial Communications Cables

Synchrophasor Processor

Move synchrophasor data to SCADA Operations Centers using standard protocols, such as DNP3. Include time stamps and time quality in the SCADA message to allow for system-wide usage of synchrophasor data. Within the RTAC logic engine, you can perform complex math and logic calculations on synchrophasor data collected from SEL relays and other IEEE C37.118 compliant devices. See *Figure 1.5* for an example of this application.

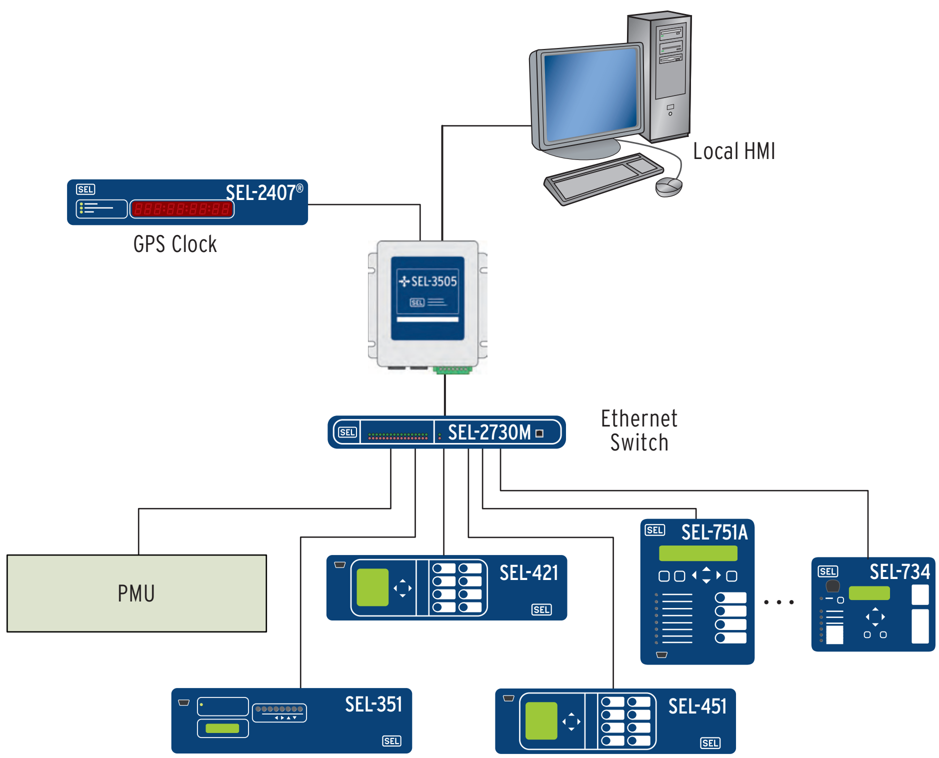


Figure 1.5 System Control and Synchrophasor Data Concentration

SCADA Data Concentrator (RTU/PLC)

Use the SEL-3505 with your protective relays and other IEDs as the substation SCADA data concentrator. You can configure the SEL-3505 to collect and view station-wide sequential events recorder (SER) and event reports. Use the SEL-3505-3 to pick up some digital inputs and provide digital outputs in a remote cabinet. Retrieve asset optimization data from SEL or other IEDs to maintain the best possible system reliability.

Access the remote SEL-3505 through the Ethernet connection, and use any web browser to manage users, view diagnostics, and access logs. Establish a remote connection with any IED connected to the SEL-3505 through engineering access communications channels. Use the SEL Fast Message protocol to maintain SCADA control and metering updates throughout the engineering access connection. Remotely manage protection and control settings in attached relays using ACSELERATOR QuickSet® SEL-5030 Software. See *Figure 1.6* for an example of this application.

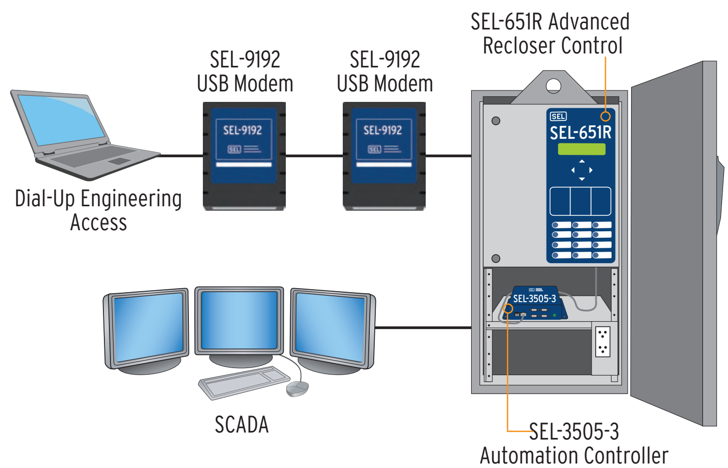


Figure 1.6 SCADA Communication Through a Single Access Point

Security Gateway

Secure engineering access and other Ethernet tunneled serial communications on the automation network with SSL/TLS or SSH encryption. Implement system security auditing, logging, and password restrictions to enforce NERC standards. Comply with role-based requirements by implementing per-user security profiles. Optionally incorporate serial and wireless encrypting devices to further secure communications to any device. Monitor the integrated light sensor and accelerometer for unauthorized cabinet access. See *Figure 1.7* for an example of this application.

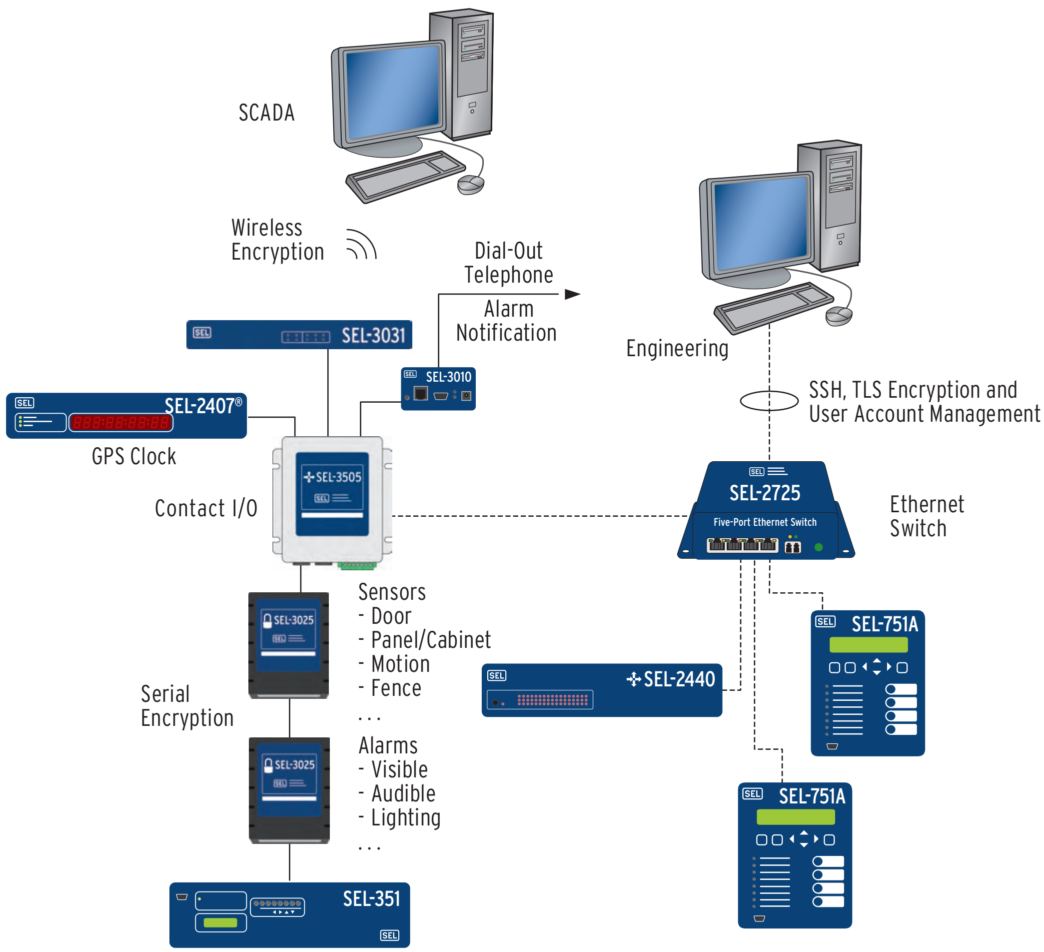


Figure 1.7 Security Through a Single Access Point

Logic Processor (Automation)

Automate existing installations with modern IEC 61131 applications. Ensure compatibility with any SEL device through the MIRRORED BITS protocol. Take advantage of multiprotocol support to collect SCADA information, process control commands, and SNTP/NTP time synchronization through a single communications link to each Ethernet device. Scale values and calculate logic in a familiar IEC 61131 configuration environment. Enjoy secure, encrypted communication to any device on the substation network or serial channel. See *Figure 1.8* for an example of this application.

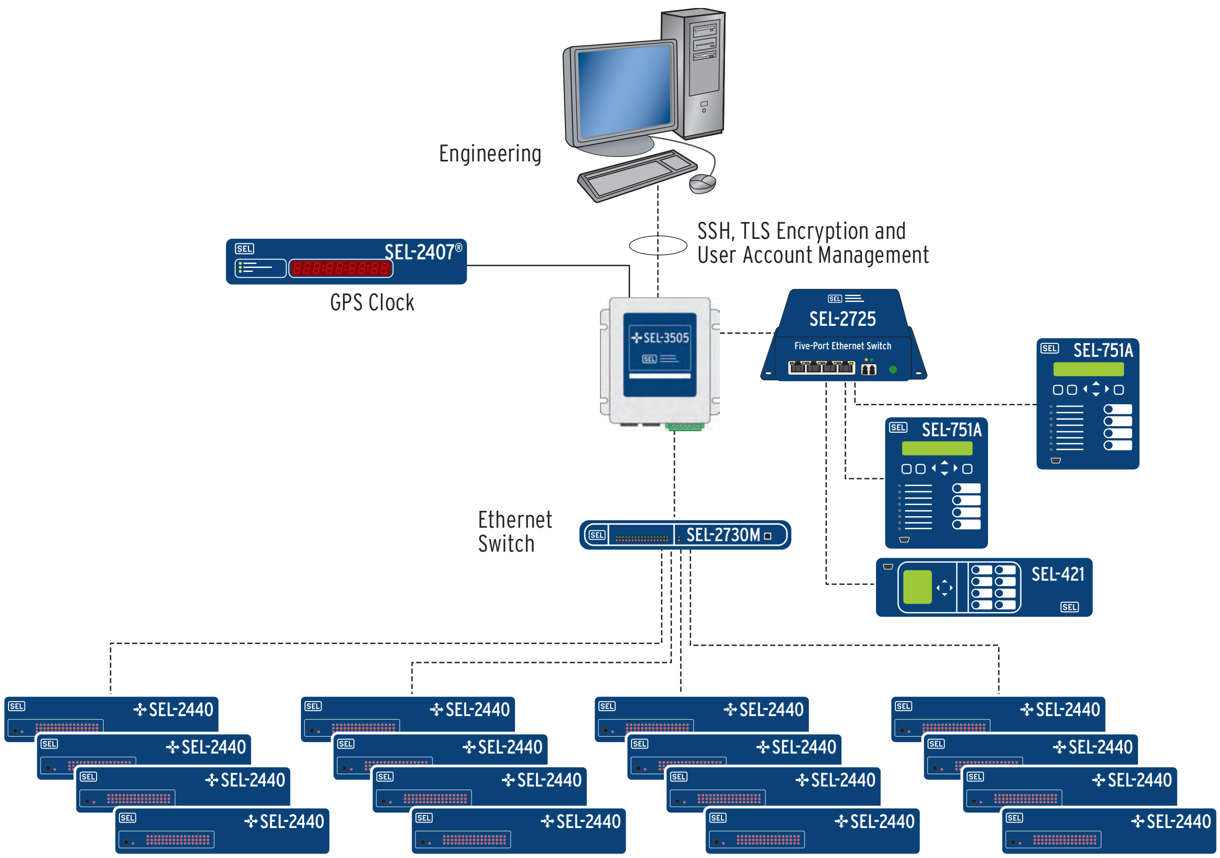


Figure 1.8 Automate and Integrate With Communication and Logic

Specifications

Compliance

Designed and manufactured under an ISO 9001 certified quality management system

47 CFR 15B, Class A

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

UL Certified to U.S. and Canadian safety standards (File E220228; NRAQ, NRAQ7)

CE Mark (does not apply to units with dial-up modem)

General

Operating System

SEL Linux® Yellowstone running Linux kernel 3.x with real-time preemption patches

Operating Temperature Range

–40° to +85°C (–40° to +185°F)
Note: Not applicable to UL applications.

Operating Environment

Pollution Degree:	2
Overvoltage Category:	II
Insulation Class:	Class I equipment
Relative Humidity:	5%–95%, noncondensing
Maximum Altitude:	2000 m

Weight (Maximum)

2.27 kg (5 lb)

Processing and Memory

Processor Speed:	333 MHz
Memory:	512 MB DDR2 ECC RAM
Storage:	2 GB

Security Features

Account Management:	User Accounts User Roles LDAP Central Authentication RADIUS Central Authentication Strong Passwords Inactive Account Logouts
Intrusion Detection:	Access/Audit Logs Alarm LED Light Sensor 3-Axis Accelerometer
Encrypted Communication:	SSL/TLS, SSH HTTPS

Automation Features

Protocols

Client

DNP3 Serial, DNP3 LAN/WAN, Modbus RTU, Modbus TCP, SEL ASCII, SEL Fast Messaging, LG 8979, IEEE C37.118, CP2179, SNMP, SES-92, CDC Type II, Courier, IEC 60870-5-103, EtherNet/IP Explicit Message Client

Server

DNP3 Serial, DNP3 LAN/WAN, Modbus RTU, Modbus TCP, SEL Fast Messaging, LG 8979, SES 92, IEC 61850 MMS, IEC 60870-5-101/104, IEEE C37.118, FTP, SFTP, CDC Type II, EtherNet/IP Implicit Message Adapter

Peer-to-Peer

IEEE-61850 GOOSE, SEL MIRRORED BITS Communications, Network Global Variables (NGVL), Parallel Redundancy Protocol

Engineering Access

Modes:	SEL Interleaved, Direct
Port Server:	Map Serial Ports to IP Ports
Secure Web Server:	Diagnostic and Communications Data

Time-Code Input (Demodulated IRIG-B)

On (1) State:	$V_{ih} \geq 2.2$ V
Off (0) State:	$V_{il} \leq 0.8$ V
Input Impedance:	1.5 kΩ
Accuracy:	250 ns

Time-Code Output (Demodulated IRIG-B)

On (1) State:	$V_{oh} \geq 2.4$ V
Off (0) State:	$V_{ol} \leq 0.8$ V
Load:	50 Ω
Output Drive Levels	
Serial Port:	TTL 5 mA, 2.4 Vdc, 500 Ω

Network Time Protocol (NTP) Modes

NTP Client:	As many as three configurable servers
NTP Server	

Precise Time Protocol (PTP)

PTP Client:	Peer delay request and end-to-end path delay supported

Communications Ports

Ethernet Ports

Ports:	2 rear
Data Rate:	10 or 100 Mbps
Rear Connectors:	RJ45 Female or LC Fiber (single-mode or multimode, 100 Mbps only)

SEL-3505 Serial Ports

Ports:	4
Type:	2 EIA-232/EIA-485 (software selectable on Ports 1 and 2) 2 EIA-232 (Ports 3 and 4)
Data Rate:	300 to 115200 bps

Connector:	DB-9 Female
Time Synchronization:	IRIG-B

SEL-3505-3 Serial Ports

Ports:	3
Type:	EIA-232/EIA-485 (software selectable)
Data Rate:	300 to 115200 bps
Connector:	DB-9 Female
Time Synchronization:	IRIG-B

USB Ports

Ports:	1
1 Device Port:	Type B

Fiber Optics

Class 1 LASER/LED

Product:	IEC 60825-1:1993 + A1:1997 + A2:2001
Data Rate:	100 Mbps
Connector Type:	LC
Wavelength:	1300 nm
Multimode Option:	62.5 µm fiber
TX Max. Power:	–14 dBm
TX Min. Power:	–20 dBm
RX Sensitivity:	–31 dBm
RX Overload:	–14 dBm
Min. TX Level:	–20 dBm
Min. RX Sensitivity:	–31 dBm
Optical Budget:	11 dBm
Max. Distance:	2 km
Single-Mode Option:	9 µm fiber
TX Max. Power:	–8 dBm
TX Min. Power:	–15 dBm
RX Sensitivity:	–25 dBm
RX Overload:	–8 dBm
Min. TX Level:	–15 dBm
Min. RX Sensitivity:	–25 dBm
Optical Budget:	10 dBm
Max Distance:	15 km

SEL-3505 Input (Units Manufactured Prior to April 2017)

Optoisolated Control Inputs

Software Settings:

ON:	15–30 Vdc
OFF:	<5 Vdc
Pickup/Dropout Delay:	1–30000 ms
Current Draw at Nominal DC Voltage:	2–4 mA

Optoisolated Control Inputs

4 Input Ranges:

12 V:	ON: 9.6–18 Vdc	OFF: <7.2 Vdc
24 V:	ON: 19.2–28.8 Vdc	OFF: <11 Vdc
48 V:	ON: 38.4–52.8 Vdc	OFF: <28.8 Vdc
125 V:	ON: 100–135.5 Vdc	OFF: <75 Vdc

Current Draw at Nominal

DC Voltage:	12 V: 2–6 mA 24 V: 4–7 mA 48 V: 2–5 mA 125 V: 2–4 mA

Solid-State Output (SEL-3505 Units Manufactured Prior to April 2017)

100 mA continuous

0–250 Vac/Vdc Operational Voltage

Max. On Resistance:	50 Ω
Min. Off Resistance:	10 MΩ
Insulation:	2500 Vac
Wiring size:	14 AWG Max. 26 AWG Min. 0.4 mm Min. Insulation 105°C, 250 V Min.

Electromechanical Outputs

Mechanical Durability:	10 M no-load operations

DC Output Ratings

Voltage:	250 Vdc
Rated Voltage Range:	19.2–275 Vdc
Rated Insulation Voltage:	300 Vdc
Make:	30 A @ 250 Vdc per IEEE C37.90
Continuous Carry:	6 A @ 70°C; 4 A @ 85°C
Contact Protection:	360 Vdc, 40 J MOV protection across open contacts
Operation Time (Coil Energization to Contact Closure, Resistive Load):	Pickup/Dropout Time ≤ 8 ms typical

Breaking Capacity (10,000 Operations):

48 V	0.50 A	L/R = 40 ms
125 V	0.30 A	L/R = 40 ms

Cyclic Capacity (2.5 Cycle/Second):

48 V	0.50 A	L/R = 40 ms
125 V	0.30 A	L/R = 40 ms

Note: Make per IEC 60255-0-20:1974.

AC Output Ratings

Rated Operational Voltage:	240 Vac
Rated Insulation Voltage:	300 Vac
Utilization Category:	AC-15 (control of electromechanical loads > 72 VA)
Contact Rating Designation:	B300 (B = 5 A, 300 = rated insulation voltage)
Contact Protection:	270 Vac, 40 J
Continuous Carry:	3 A @ 120 Vac 1.5 A @ 240 Vac
Rated Frequency:	50/60 ± 5 Hz

Operating Time (coil energization to contact closure): Pickup/Dropout Time ≤ 8 ms

Electrical Durability Make VA Rating: 3600 VA, cosϕ = 0.3

Electrical Durability Break VA Rating: 360 VA, cosϕ = 0.3

Power Supply

Complies with IEC HiPot and Impulse standards, except when connected to substation battery. The auxiliary (power supply) circuit must be connected to a battery (or other external power supply meeting application requirements) that is not used for switching inductive loads and will provide the required hold-up time.

Input Voltage

Rated Voltage: 12–24 Vdc
24–48 Vdc

Note: 12 V power supply rating does not meet the minimum operating voltage requirement of IEEE 1613.

Operational Voltage Range: 9.8–30 Vdc polarity dependent
19.2–57.6 Vdc polarity dependent

Peak Inrush Current

12 Vdc: 19 A

24 Vdc: 44 A

48 Vdc: 91 A

Power Consumption

DC: 7 W (with dual fiber Ethernet)
5 W (with dual copper Ethernet)

Fuse Rating (Internal)

F1:

Type: time lag T

Current Rating: 3.15 A

Voltage Rating: 250 Vac, 300 Vdc

IEC 60127-2/5: H = 1500 A @ 250 Vac, p.f. = 0.7 – 0.8

UL 248-14: 10 kA @ 125 Vac, p.f. = 0.7 – 0.8 / 1500 A @ 250 Vac, p.f. = 0.7 – 0.8 / 1500 A @ 300 Vdc

Product Standards

Communications Equipment in Utility Substations: IEC 61850-3:2013
IEEE 1613-2009
Severity Level: Class 1 (excluding dial-up modem)

Measuring Relays and Protection Equipment: IEC 60255-26:2013*
IEC 60255-27:2013

* Acceptance Criteria C applied to 0% dc voltage dips for 10 ms. The auxiliary (power supply) circuit is intended to be connected to a battery (or other external power supply meeting application requirements) that is not used for switching inductive loads and will provide the required hold-up time.

Type Tests

Environmental Tests

Enclosure Protection: IEC 60529:2001 + CRGD:2003
Severity Level: IP30 (excluding the terminal blocks)

Vibration Resistance: IEEE 1613-2009
IEC 60255-21-1:1988
Severity Level:
Endurance Class 2
Response Class 2

Shock Resistance: IEEE 1613-2009
IEC 60255-21-2:1988
Severity Level:
Shock Withstand, Bump Class 1
Shock Response Class 2

Seismic: IEC 60255-21-3:1993
Severity Level:
Quake Response Class 2

Cold, Operational and Storage: IEC 60068-2-1:2007
Severity Level:
–40°C, 16 hours

Dry Heat, Operational and Storage: IEC 60068-2-2:2007
Severity Level:
85°C, 16 hours

Damp Heat, Cyclic: IEC 60068-2-30:2005
Severity Level:
25°–55°C, 6 cycles,
95% relative humidity

Damp Heat, Steady State: IEC 60068-2-78:2012
Severity Level:
40°C, 240 hours,
93% relative humidity

Dielectric Strength and Impulse Tests

The following standards only apply if the device is not connected directly to the station battery.

Dielectric (HiPot): IEC 60255-27:2013
IEEE C37.90-2005 Class B,
Section 8: Dielectric Tests
Dielectric Strength Section
Severity Level:
2500 Vac for one minute on contact inputs, contact outputs
1600 Vdc for one minute on power supply

Impulse: IEC 60255-27:2013
IEEE C37.90-2005 Class B
Severity Level: 0.5 J, 2.5 kV

RFI and Interference Tests

EMC Immunity

Electrostatic Discharge Immunity: IEEE C37.90.3-2001
IEC 61000-4-2:2008
Severity Level:
2, 4, 6, 8 kV contact discharge;
2, 4, 8, 15 kV air discharge

Magnetic Field Immunity: IEC 61000-4-8:2009
Severity Level:
1000 A/m for 3 seconds,
100 A/m for 1 minute
IEC 61000-4-9:2001
Severity Level: 1000 A/m

Power Supply Immunity: IEC 61000-4-11:2004
IEC 61000-4-17:1999+A1:2001+A2:2008
IEC 61000-4-29:2000

Radiated RF Immunity: IEC 61000-4-3:2010
Severity Level: 10 V/m, excluding dial-up modem
IEEE C37.90.2-2004
Severity Level: 35 V/m, excluding dial-up modem

Fast Transient, Burst Immunity:	IEC 61000-4-4:2012 Severity Level: 4 kV @ 5.0 kHz 2 kV @ 5.0 kHz for comm. ports
Surge Withstand Capability Immunity:	IEEE C37.90.1-2002 Severity Level: 2.5 kV oscillatory 4 kV fast transient Excluding dial-up modem IEC 61000-4-18:2006 + A1:2010 Severity Level: 2.5 kV common-mode 1.0 kV differential-mode 1 kV common-mode on comm. ports
Surge Immunity:	IEC 61000-4-5:2005 Severity Level: 1 kV line-to-line 2 kV line-to-earth 2 kV comm. ports
Conducted RF Immunity:	IEC 61000-4-6:2008 Severity Level: 10 Vrms
Digital Radio Telephone RF Immunity:	ENV 50204:1995 Severity Level: 10 V/m at 900 MHz and 1.89 GHz

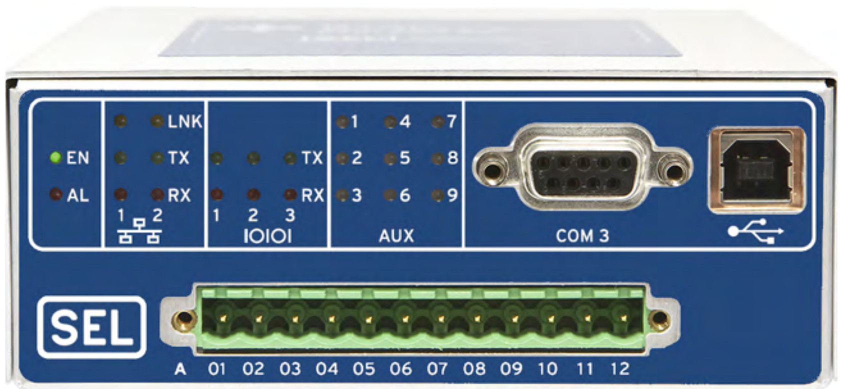
EMC Emissions

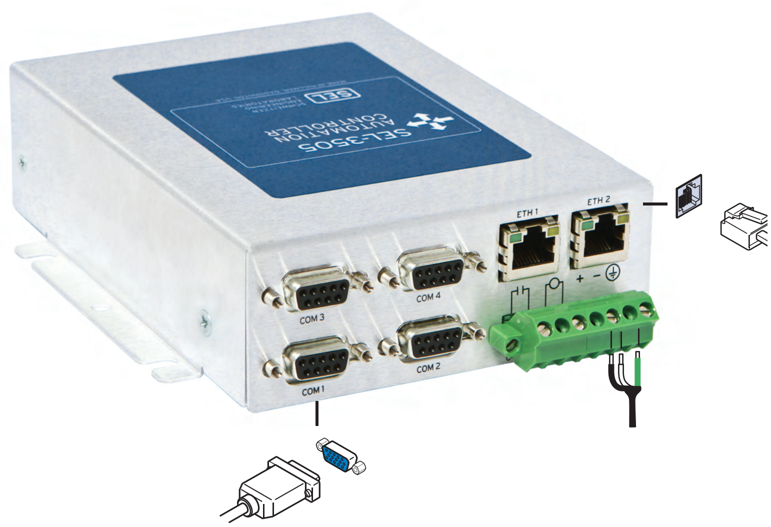
Radiated and Conducted Emissions:	CISPR 11:2009+A1:2010 CISPR 22:2008 ANSI C63.4-2014 Class A Canada ICES-001(A) / NMB-001(A)

Section 2
Installation

Overview

The first steps in applying the SEL-3505 Real-Time Automation Controller (RTAC) are installing and connecting the device. This section describes common installation features and requirements. To install and connect the controller safely and effectively, you must be familiar with RTAC configuration features and options.





Device Placement

NOTE: For IEC 60255-27 compliant applications, the following applies.

The top surfaces of barriers that are accessible in normal use meet at least the requirements of the protective type IP4X. The top has sufficient mechanical strength, stability, and durability to maintain the specified degree of protection and is firmly secured in place in such a way that it can only be removed by the use of a tool. If the unit is mounted in an orientation such that the rear surface, left rear bottom corner, or right rear bottom corner could be considered accessible in normal use and part of the top surface, the unit must be installed in an external enclosure to prevent access in normal use. If the external enclosure has a top surface that is accessible in normal use, the top surface of the external enclosure must meet at least the requirements of the protective type IP4X according to IEC 60529 and have sufficient mechanical strength, stability, and durability to maintain the specified degree of protection and be firmly secured in place in such a way that it can only be removed by the use of a tool.

Mount the SEL-3505 in a sheltered indoor environment (a building or an enclosed cabinet) that does not exceed the temperature and humidity ratings for the device (see *Specifications on page 1.13*). The unit is rated Installation/ Overvoltage Category II and Pollution Degree 2. This rating allows mounting of the unit indoors or in an outdoor (extended) enclosure where the unit is protected against exposure to direct sunlight, precipitation, and full wind pressure, but temperature and humidity are not controlled.

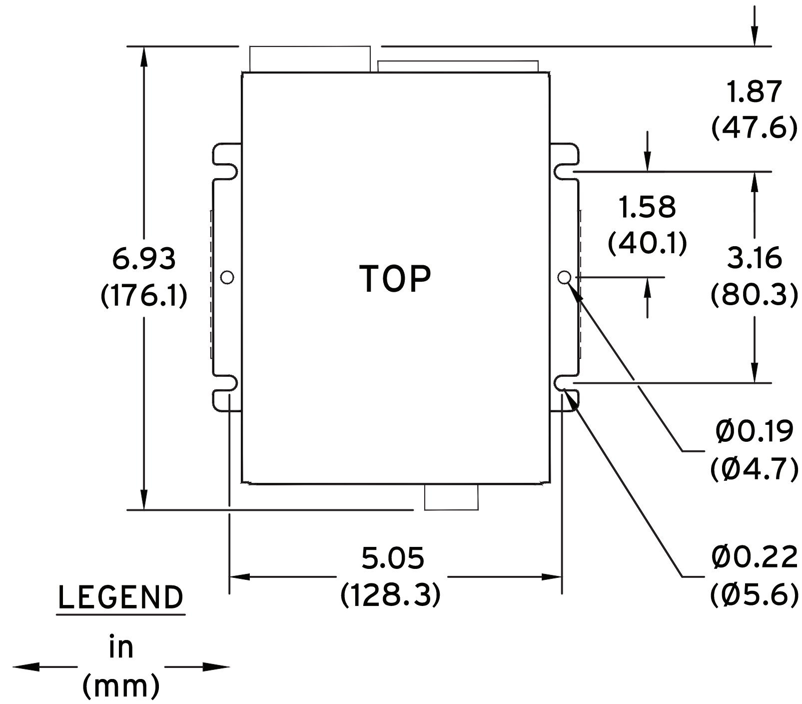


Figure 2.1 Surface-Mount Dimensions

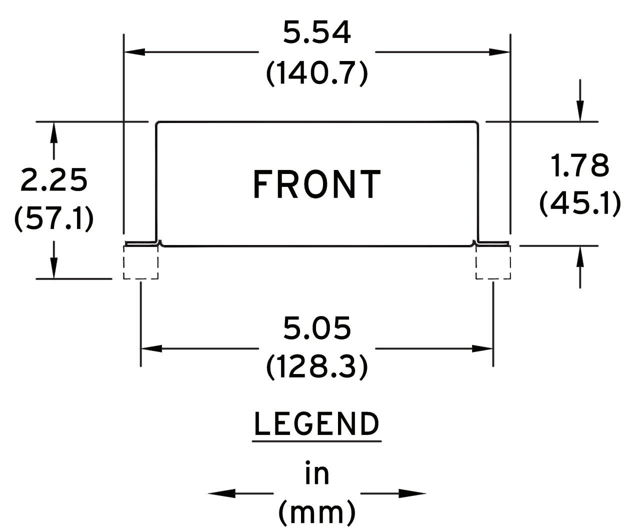


Figure 2.2 DIN-Mount Dimensions

Front- and Rear-Panel Drawings

The following figures show the front-panel status LEDs provided to simplify system troubleshooting and the rear-panel connectors used for communications and wiring.

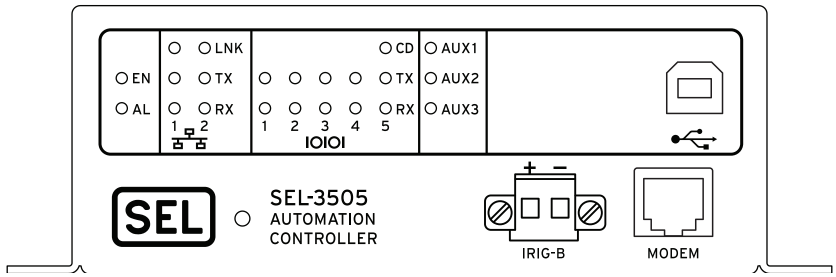


Figure 2.3 SEL-3505 Front-Panel, Surface-Mount Drawing

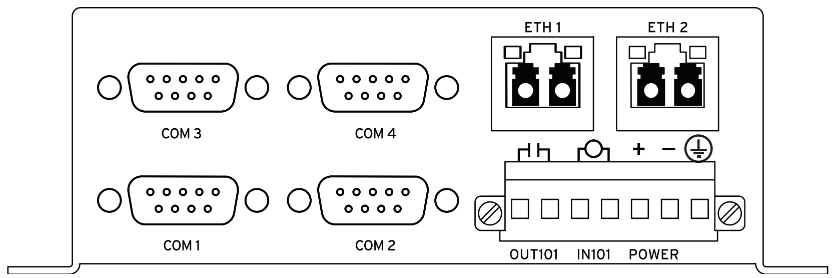


Figure 2.4 SEL-3505 Rear-Panel, Surface-Mount Drawing

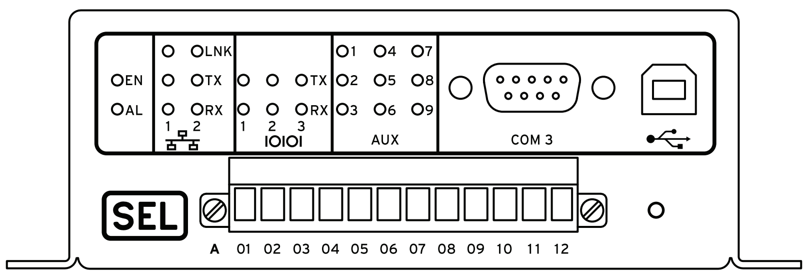


Figure 2.5 SEL-3505-3 Front-Panel, Surface-Mount Drawing

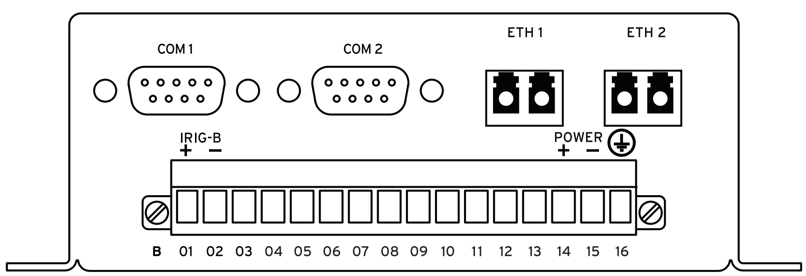


Figure 2.6 SEL-3505-3 Rear-Panel, Surface-Mount Drawing

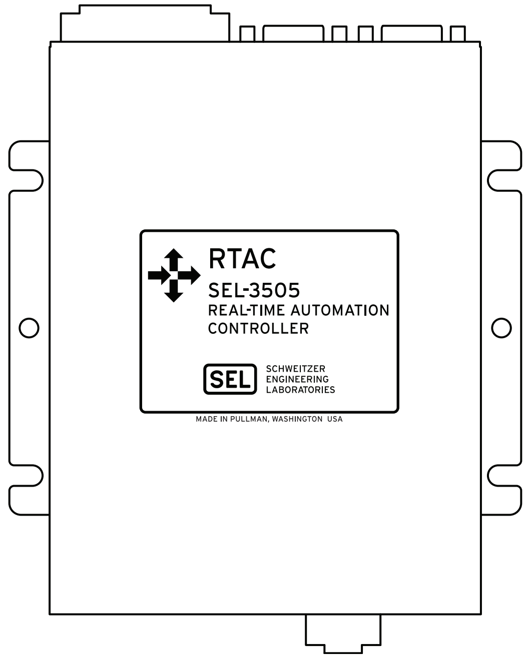


Figure 2.7 SEL-3505 Top-Panel, Surface-Mount Drawing

NOTE: OUT101 is NC when the setting Tie Alarm LED to OUT101 is checked and NO when the setting is not checked.

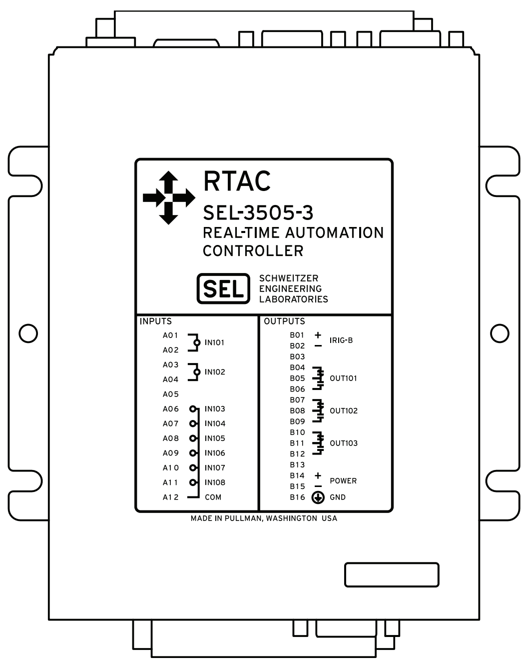


Figure 2.8 SEL-3505-3 Top-Panel, Surface-Mount Drawing

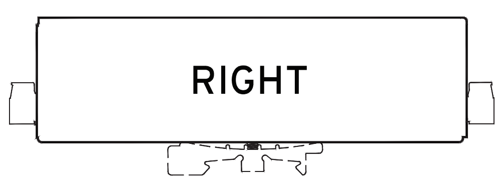


Figure 2.9 DIN-Mount Drawing

Connections

Power

> ⚠**DANGER**
> Contact with instrument terminals can cause electrical shock that can result in injury or death.

NOTE: To comply with IEC HiPot and Impulse standards, use a dc to dc converter instead of connecting directly to the station battery. The auxiliary (power supply) circuit must be connected to a battery (or other external power supply meeting application requirements) that is not used for switching inductive loads and will provide the required hold-up time.

Connect power from a correctly rated voltage source to the **POWER** terminals on the rear panel with the proper polarity. Applying voltages that are outside the specifications for the unit can cause permanent damage to the unit. See Power Supply under *Options on page 1.4* for a list of voltage ratings. The **POWER** terminals are isolated from chassis ground. Use 16–14 AWG (1.5–2.5 mm^2) size wire to connect to the **POWER** terminals (see *Figure 2.10*). Terminal labels for power, digital inputs, and digital outputs are located on the top cover of the SEL-3505-3, as shown in *Figure 2.8*.

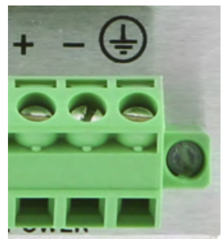


Figure 2.10 Power Connections

Grounding (Earthing)

Connect the ground terminal labeled with the ground symbol on the rear-panel to a rack frame or switchgear ground for proper safety and performance. Use 14 AWG (2.5 mm^2) wire less than 2 m (6.6 ft) in length for the ground connection. Do not remove the protective earth connection when the equipment is energized.

Digital Input

NOTE: SEL recommends that you use 20-14 AWG (0.5-2.0 mm^2) wire of sufficient current capacity and voltage rating for the application.

Connect as many as eight digital inputs to the front connection of the SEL-3505-3. **IN101** and **IN102** are isolated inputs whereas **IN103–IN108** share a common return through the pin labeled **COM** (see *Figure 2.8* and *Figure 2.11*).

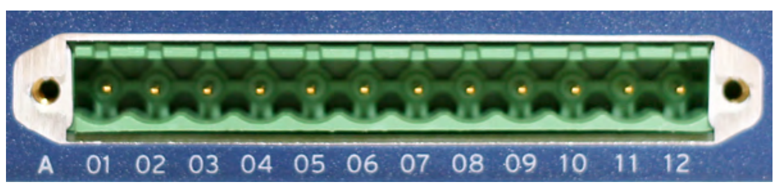


Figure 2.11 Front Panel Digital Inputs

Digital Output

NOTE: SEL recommends that you use 20-14 AWG (0.5-2.0 mm^2) wire of sufficient current capacity and voltage rating for the application.

Connect as many as three digital outputs to the rear connection of the SEL-3505-3. Each output has a normally open and normally closed connection option (see *Figure 2.8* and *Figure 2.12*).

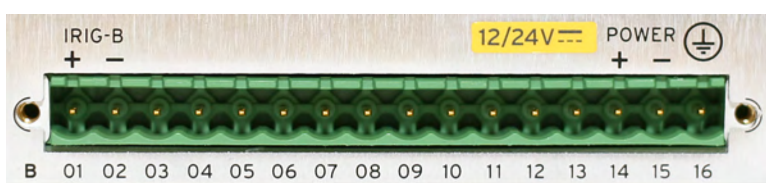


Figure 2.12 Rear-Panel Digital Outputs

Communications Ports

All web access, settings changes, and ODBC connections use either the SEL-3505 Ethernet ports, the front USB Type B port, or on an SEL-3505 with the optional modem port, using Point-to-Point Protocol (PPP). You can use the supplied USB Type B cable for initial configuration of RTAC web settings.

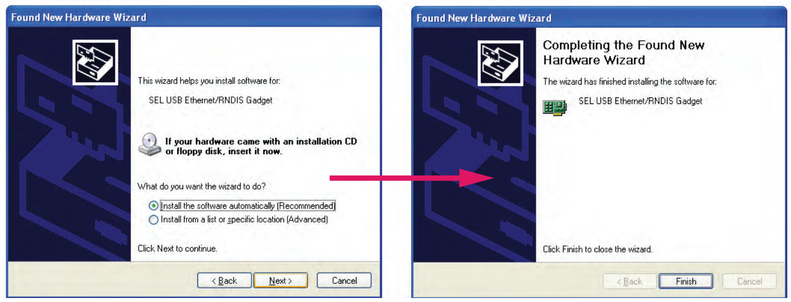
NOTE: Never connect two RTACs via USB to one PC.

NOTE: USB ports are intended for programming only, not continuous use.

NOTE: Do not use USB cables longer than three meters.

The ACSELERATOR RTAC® SEL-5033 Software installation will place the required USB driver on your PC. Once Microsoft® Windows® detects the USB connection, the driver will be installed automatically. Plug the USB cable into the SEL-3505 and into your PC. If you are prompted to connect to Windows Update, select **No, not at this time** and press **Next**. Follow the automatic install prompts, if any, using the Windows Device Installation Wizard to install the SEL USB driver.

After completing this step, you can access the SEL-3505 web interface through the USB cable using IP address 172.29.131.1. See the *SEL-5033 Software Instruction Manual, Section 7: Security and Account Management* to configure the SEL-3505 password using the web interface.



The SEL-3505 has two independent Ethernet ports. Default IP addresses for the SEL-3505 are as follows:

ETH 1: 192.168.1.2

ETH 2: 192.168.2.2

USB: 172.29.131.1 (not configurable)

As described in the document *Getting Started With the SEL-3505/SEL-3505-3*, enter the correct IP address that corresponds to the PC to RTAC connection in the address bar of a web browser. Use the resulting RTAC web interface to configure user RTAC user accounts, network settings and other configuration parameters described in the ACSELERATOR RTAC® SEL-5033 Instruction Manual. Use the same IP address in the ACSELERATOR RTAC software to send project settings to your RTAC, as described in the ACSELERATOR RTAC SEL-5033 Instruction Manual.

NOTE: Do not use serial cables longer than 10 meters.

The SEL-3505 has four nonisolated serial ports with the two bottom ports software selectable as EIA/232 or EIA-485/EIA-422, and the two top ports fixed as EIA-232. The SEL-3505-3 has three nonisolated serial ports that are software selectable as EIA-232 or EIA-485/EIA-422. You can configure any serial protocol on the SEL-3505 to use any of these serial ports. See *Table 2.1* for the pin-out of the SEL-3505 serial ports. See *Figure 2.13* and *Figure 2.14* for typical EIA-485 two-wire and four-wire connections schematics.

Table 2.1 Nonisolated Female DB-9 Ports (Sheet 1 of 2)

EIA-232	EIA-485/EIA-422
Pin 1: N/C (SEL-3505), +5 V (SEL-3505-3)	Pin 1: N/C (SEL-3505), +5 V (SEL-3505-3)
Pin 2: RXD	Pin 2: –RXD
Pin 3: TXD	Pin 3: –TXD
Pin 4: +IRIG-B	Pin 4: +IRIG-B
Pin 5: GND	Pin 5: GND
Pin 6: –IRIG-B	Pin 6: –IRIG-B
Pin 7: RTS	Pin 7: +TXD

Table 2.1 Nonisolated Female DB-9 Ports (Sheet 2 of 2)

EIA-232	EIA-485/EIA-422
Pin 8: CTS	Pin 8: +RXD
Pin 9: GND	Pin 9: GND

Table 2.2 Port Characteristics

Port	Port Interface	Cables
USB-B	USB Type B to USB Type A	C664 (no longer than three meters)
PORT 1–PORT 4 (serial SEL-3505)	Ports 1, 2 EIA-232/EIA-485 (Nonisolated) Ports 3, 4 EIA-232	C234A, 273A, and C387 are popular selections.[a] Twisted pair cables such as C698, or C697, are strongly recommended for EIA-485 installations to reduce noise susceptibility.
PORT 1–PORT 3 (serial SEL-3505-3)	EIA-232/EIA-485 (Nonisolated)	
IRIG-B Input (SEL-3505)	Front compression type	C962
IRIG-B Input (SEL-3505-3)	Rear compression type	C962
IRIG-B Output	Pins 4 and 6 in serial ports	C273A and C273R are popular selections.
ETH 1 and ETH 2	Optional 10/100BASE-T, 100BASE-FX, 100BASE-LX10	C627 (RJ45 for Copper) Industrial cables such as C627R are recommended for reduced noise susceptibility.

[a] Do not use serial cables longer than 10 meters.

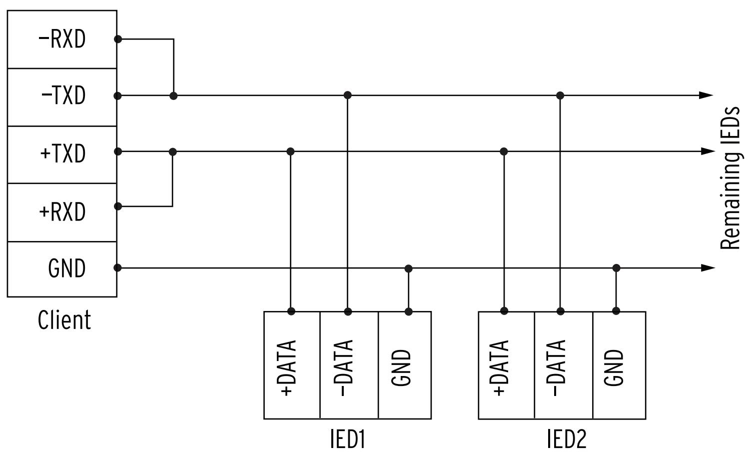


Figure 2.13 EIA-485 Typical Two-Wire Connection

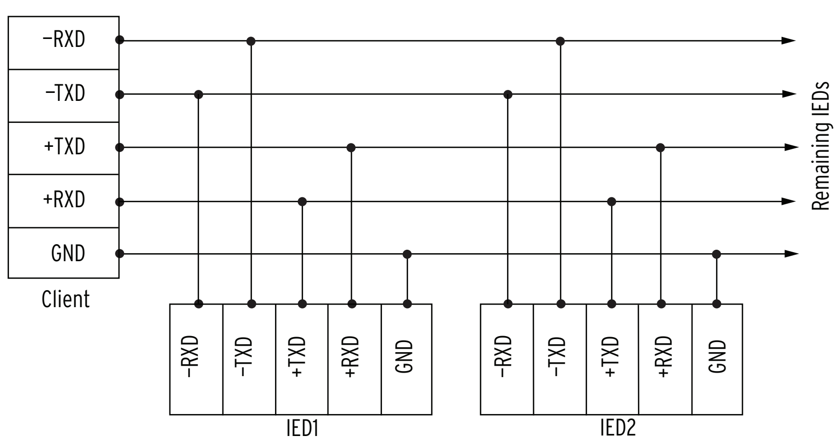


Figure 2.14 EIA-485 Typical Four-Wire Connection

Modem Usage

You can configure the optional 56 kbps internal modem as a dial-up serial connection through the SEL-3505 web interface by clicking on **Interface > PPP_01 > Edit**, then choosing **Disable Interface**. The modem port is now accessed as PORT 5 in the ACSELERATOR RTAC software. You can use this port for dial-up connections or dial-out connections for the DNP3 protocol.

PPP Communication

Point-to-Point Protocol (PPP) provides Internet protocol communication among devices over dial-up modems. Through PPP, you can perform all operations such as project download, web page access, etc. through a remote dial-up connection. You can configure the SEL-3505 to communicate with PPP protocol using the SEL-3505 web interface on an optional front-panel modem port. To use PPP with the SEL-3505, you must also configure your local PC for PPP.

NOTE: Ethernet communications using PPP occur over a dial-up connection, so they will be noticeably slower than a direct Ethernet or USB connection.

To configure PPP on the SEL-3505, perform the following steps:

Step 1. In the SEL-3505 web interface, click on **Interface** under the **Network** heading.

Step 2. Click on the **Edit** button under **PPP_01**. See *Figure 2.15*.

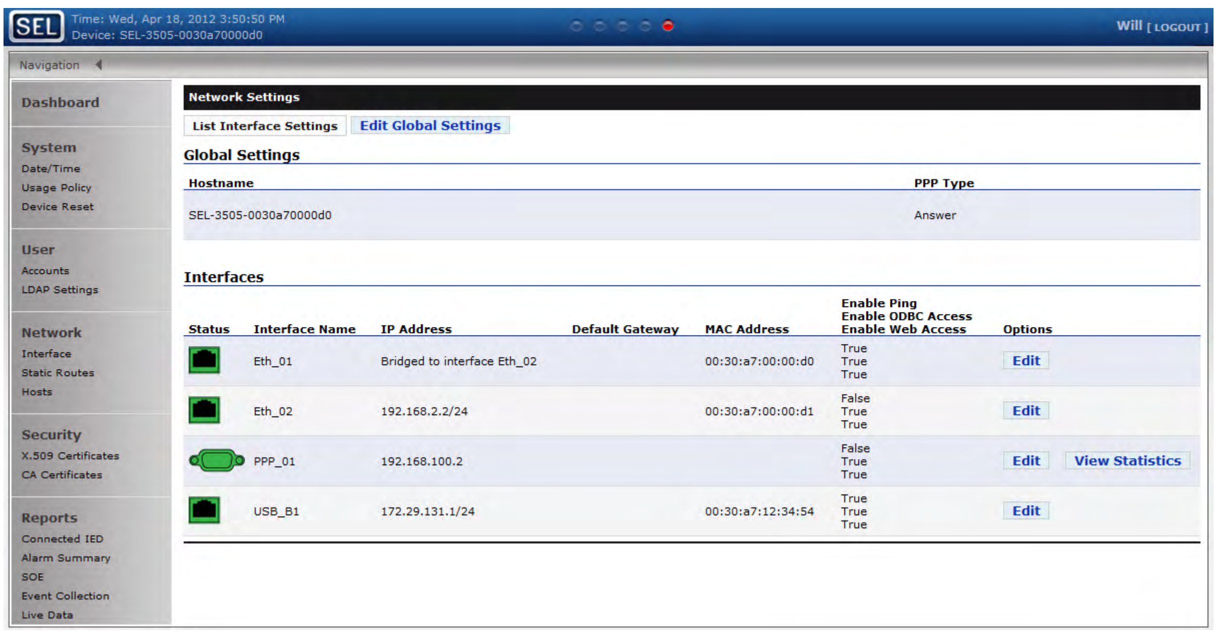


Figure 2.15 Network Settings

Step 3. Configure the Interface PPP_01 settings to enable the interface. See *Table 2.3*.

Table 2.3 PPP Configuration Parameters (Sheet 1 of 3)

Setting	Description
Mode	Configure PPP to answer incoming calls or originate outgoing calls.
Asyncmap	A 32-bit hex bitmap, represented as hex digits, identifying which control characters must be avoided (replaced by a two-character sequence) to prevent their interpretation by link equipment. Each bit in the map corresponds to ASCII characters. To form the map, perform the following: Step 1. Write 32 0s in a row. The right-most bit corresponds to ASCII NULL, the left-most bit represents ASCII 31.

Table 2.3 PPP Configuration Parameters (Sheet 2 of 3)

Setting	Description
	Step 2. Set each bit = 1 to correspond to the ASCII control character that needs to be avoided (start counting at 0). Step 3. Group the 32 bits in sets of four. Step 4. Convert each four-bit group into a hex number. Step 5. Enter the map as 0x followed by the group of eight hex characters. For example, to avoid XON and XOFF, enter the following:

ASCII Code	Dec	Meaning	Keyboard Strokes
DC1	17	XON	**<Ctrl+Q>**
DC3	19	XOFF	**<Ctrl+S>**

Setting	Description
	Bitmap=00000000000010100000000000000000 Asyncmap = 0X000A0000 By default, the map is set to 0xffffffff, which indicates avoidance of all control characters. However, each avoided character results in transmission of two characters, so changing the map can increase performance to a degree.
Maximum Receive Unit	Maximum receive packet size
Maximum Transmit Unit	Maximum transmit packet size
PPP Compression	Select zero or more compression methods.
BSD	Use the BSD compress scheme to request the peer compress packets.
Deflate	Use the Deflate scheme to request the peer compress packets.
Authentication Methods	Select zero or more authentications methods for this PPP session.
PAP	Password Authentication Protocol. After establishment of a link, the initiator sends a plaintext IP/password pair until the authenticator indicates acceptance of this pair.
CHAP	Challenge-Handshake Authentication Protocol. Once a connection exists, the authenticator occasionally issues a challenge to the peer. The peer responds with a value derived from a one-way hash function, which the authenticator checks for validity. Failure to authenticate results in connection termination.
MSCHAP	Microsoft version of CHAP
MSCHAP-V2	Uses CHAP algorithm 0x81 in LCP option 3, which provides for a 16-octet challenge.
EAP	Extensible Authentication Protocol. Runs over the link layer and is suitable for when multiple authentication mechanisms are in place.
Local IP Address	Enter the local IP address of the PPP interface. If you select Get Local IP Address from remote, the remote will provide the IP address or the connection drops.
Remote IP Address	Enter the IP address of the remote device. If PPP interface is in Originate mode, you must select **Get Remote address from remote** to allow the PPP connection to accept the address directly from the remote.
Request DNS Servers	If checked, the SEL-3505 will ask the peer for as many as two DNS servers. If the peer supplies DNS servers, these will be added to the system for the duration of the connection.

Table 2.3 PPP Configuration Parameters (Sheet 3 of 3)

Setting	Description
Link Idle Timeout	If enabled, this is the timeout in seconds during which the link can be idle before disconnecting. Range = 1–3600 seconds.
Maximum Connection Time	If enabled, this is the maximum time in seconds during which this link can stay enabled. Range = 1–300 seconds.
Modem Init String	String sent to the modem prior to originating or answering any calls. If the modem rejects the string, the system uses a default string.
Remote Phone Number	Enter at least one and as many as three valid remote phone numbers for the PPP connection.
Authentication Username	The user name for outgoing connections.
Authentication Password	The password for outgoing connections.
Dialout Method	On-Demand connects as necessary, Persistent remains connected.
Redial Holdoff Timeout	The number of seconds between redial attempts. When the SEL-3505 is in Originate mode, it will dial the configured phone number to attempt to establish a PPP connection. If dial is unsuccessful, the holdoff timeout will delay the next dial out attempt. During this holdoff time, the SEL-3505 is still in dial-out mode and will not listen for incoming calls. A value of 0 results in no delay between redial attempts.

Once the SEL-3505 successfully establishes a PPP connection with a peer, the interface will operate more slowly but with the same functionality as for any of the other Ethernet interfaces on the SEL-3505. Click on **View Statistics** to monitor PPP transmit and receive statistics. The statistic values are view-only from the web interface and are unavailable to the logic engine or RTAC projects.

Input

The dc voltage SEL-3505 optoisolated inputs are not polarity dependent. Refer to *Section 1: Introduction and Specifications* for optoisolated input ratings and *Figure 2.10* for terminal assignments. Configure contact inputs under Contact I/O in ACSELERATOR RTAC. You can change the name of the input points, create an alias tag name for the points, and configure pickup and dropout delays.

The SEL-3505 also has a built-in light sensor that populates the system tag Light_Sensor_Measurement to indicate the level of activity it detects. The light sensor provides an analog value that increases as the ambient light level increases. You can use the light sensor to detect when the cabinet door is opened by comparing the light sensor reading when the cabinet door is closed to the present reading. A reading higher than the ambient reading indicates the door is opened.

The SEL-3505 has a built-in three-axis accelerometer with readings available in the project system tags. You can use the Accelerometer_X, Accelerometer_Y and Accelerometer_Z axis system tags in user logic to determine if any unusual vibrations are occurring at the unit, such as a plant system upset or manual tampering with the cabinet. See SEL application guide AG2015-07, *Using the SEL-3505 Accelerometer and Light Sensor for Physical Security Monitoring* for more details.

Output

Refer to *Section 1: Introduction and Specifications* for output contact ratings and *Figure 2.12* for terminal assignments. Configure contact outputs under Contact I/O in ACSELERATOR RTAC. You can change the name of the points, create alias tag names for the points, and initialize the status values. The RTAC will use the initialized values until the actual value is populated at run-time.

Field Serviceability

You can upgrade the SEL-3505 firmware and custom programming in the field or remotely over Ethernet. Self-tests provide status indication of errant conditions which may occur in the SEL-3505. You can map one or a combination of these or any other status indications to the alarm contact to create a diagnostic alarm (OUT101).

Real-Time Clock Battery Replacement

The only field replaceable component is the real-time clock battery which cannot be recharged. A lithium battery powers the clock (date and time) if the external power source is lost or removed. The battery is a 3 V lithium coin cell, Rayovac BR1632 or equivalent. At room temperature (25°C), the battery will operate nominally for 10 years. When the device is powered from an external source, the battery experiences a low self-discharge rate. Thus, battery life may extend well beyond 10 years.

To replace the real-time clock battery, you can disassemble the SEL-3505, including removing the top circuit board, or, you can remove the battery with a pair of plastic-tipped tweezers or insulated tip needle-nosed pliers. To remove the battery without removing the boards:

⚠DANGER
Disconnect or de-energize all external connections before opening this device. Contact with hazardous voltages and currents inside this device can cause electrical shock resulting in injury or death.

Step 1. De-energize the device and remove it from the rack or panel.

Step 2. Follow *SEL-3505 Disassembly* instructions but do not remove the top circuit board.

Step 3. Looking from the side of the unit adjacent to the USB connector, locate the battery holder on the bottom board.

Step 4. Carefully insert plastic-tipped tweezers or other insulated tool and pull the battery out of the holder.

⚠CAUTION
The device contains components sensitive to Electrostatic Discharge (ESD). When working on the device with the front panel removed, work surfaces and personnel must be properly grounded or equipment damage may result.

Step 5. Insert the new battery into the holder with the insulated tool, taking care to place the battery with the positive (+) side facing up.

Step 6. Follow *SEL-3505 Reassembly* instructions, then reinstall all connectors.

Step 7. Energize the unit and set the device date and time.

⚠CAUTION
There is danger of explosion if the battery is incorrectly replaced. Replace only with Rayovac no. BR1632 or equivalent recommended by manufacturer. See Owner's Manual for safety instructions. The battery used in this device may present a fire or chemical burn hazard if mistreated. Do not recharge, disassemble, heat above 100°C or incinerate. Dispose of used batteries according to the manufacturer's instructions. Keep battery out of reach of children.

If you do not have a tool with insulated tips, you can remove the top board for easy access to the battery by following these steps:

Step 1. Follow *SEL-3505 Disassembly* instructions to expose the bottom circuit board.

Step 2. Locate the battery clip (holder) on the board.

Step 3. Carefully remove the battery from beneath the clip. Properly dispose of the old battery.

Step 4. Install the new battery with the positive (+) side facing up.

Step 5. Follow *SEL-3505 Reassembly* instructions, then reinstall all connectors.

Step 6. Energize the unit and set the device date and time.

Jumpers

Table 2.4 SEL-3505 Configurable Jumper Positions

Jumper	Position
JMP 1	A Reset device to factory defaults C Override login authentication

SEL-3505 Disassembly

⚠DANGER
Disconnect or de-energize all external connections before opening this device. Contact with hazardous voltages and currents inside this device can cause electrical shock resulting in injury or death.

⚠CAUTION
The device contains components sensitive to Electrostatic Discharge (ESD). When working on the device with the front panel removed, work surfaces and personnel must be properly grounded or equipment damage may result.

To disassemble the SEL-3505, perform the following steps.

Step 1. De-energize the device and remove it from the rack or panel.

Step 2. Remove the four retaining screws (two on each side) and plug-in connectors.

Step 3. Remove the top panel by rocking the front up and away from the chassis, then sliding it away from the back.

Note: If you slide it directly back without lifting the front first, you may damage the heat sink Sil-Pad® insulators that are attached to the top panel.

Step 4. Only if you need to remove the top circuit board, remove the four retaining screws that attach the top circuit board to the bottom stand-offs.

Step 5. Only if you need to remove the top circuit board, gently pry straight up on the side of the top circuit board to separate the 50-pin header located on the side of the unit from the top board. Take care to not bend the pins.

SEL-3505 Reassembly

⚠DANGER
Disconnect or de-energize all external connections before opening this device. Contact with hazardous voltages and currents inside this device can cause electrical shock resulting in injury or death.

⚠CAUTION
The device contains components sensitive to Electrostatic Discharge (ESD). When working on the device with the front panel removed, work surfaces and personnel must be properly grounded or equipment damage may result.

To reassemble the SEL-3505, perform the following steps:

Step 1. If you removed it, carefully replace the top circuit board, ensuring the 50-pin header engages the top mating connector.

Step 2. If you removed them, replace the four retaining screws to secure the top circuit board to the bottom stand-off connectors.

Step 3. Replace the top panel by inserting the rear connectors through the rear of the top panel while holding the front of the top panel away from the circuit board.

Note: Do not slide the panel straight onto the unit, because this may scrape the heat sink Sil-Pad insulators off of the top panel.

Step 4. Once the rear connectors are through the top panel holes, rock the front of the top panel down so that the heat sink Sil-Pad insulators that are located on the top panel land on the corresponding IC chips on the circuit board.

Step 5. Reinstall all screws and connectors.

Section 3
Factory Reset

Override Login

NOTE: ODBC and other tools may still prompt you for a password when the RTAC password override is enabled. You can enter any text for a password in those situations.

⚠DANGER
Disconnect or de-energize all external connections before opening this device. Contact with hazardous voltages and currents inside this device can cause electrical shock resulting in injury or death.

⚠CAUTION
The device contains components sensitive to Electrostatic Discharge (ESD). When working on the device with the front panel removed, work surfaces and personnel must be properly grounded or equipment damage may result.

The SEL-3505 has a jumper that allows you to override the requirement to use a user password. SEL recommends using this jumper only when necessary to restore a unit when the administrative password is forgotten. For security reasons, you should not operate the SEL-3505 without a password.

To override login user name/password verification, perform the following steps:

Step 1. Follow *SEL-3505 Disassembly on page 2.12* to expose the top circuit board.

Step 2. Locate the reset jumper (JMP1).

Step 3. Place a jumper across the C position in the jumper block.

Step 4. Apply power to the unit. Wait until the ENABLED LED is on.

Step 5. Log on to the SEL-3505 via the web interface with user name **Edison** and no password. Edison is an administrative account that is only active when a jumper is in the C position. Previously configured user accounts will remain intact.

Step 6. Make any user account changes you need to make.

Step 7. Turn off the SEL-3505 and remove the jumper.

Step 8. Follow *SEL-3505 Disassembly on page 2.12*, reinstall all connectors and energize the unit.

Reset to Factory Settings

You can reset the SEL-3505 to factory-default settings via the web interface or by using a hardware jumper. All programming, user accounts, logs, etc., will be lost and the procedure is not reversible.

Perform the following steps to reset the SEL-3505 to factory-default settings via the web interface.

Step 1. Log on to the web interface.

Step 2. Click on **Device Reset**.

Step 3. Check **Reset To Factory Default Settings** (see *Figure 3.1*).

Step 4. Click **Submit**.

Figure 3.1 Factory Reset via Web Interface

⚠DANGER
Disconnect or de-energize all external connections before opening this device. Contact with hazardous voltages and currents inside this device can cause electrical shock resulting in injury or death.

⚠CAUTION
The device contains components sensitive to Electrostatic Discharge (ESD). When working on the device with the front panel removed, work surfaces and personnel must be properly grounded or equipment damage may result.

Perform the following steps to reset the SEL-3505 to factory defaults using the internal jumper only as a last attempt to recover from a failed state.

Step 1. Follow *SEL-3505 Reassembly on page 2.12* to expose the top circuit board.

Step 2. Locate the reset jumper (**JMP1**).

Step 3. Place a jumper across the **A** position in the jumper block.

Step 4. Apply power to the unit. Wait until the **ENABLED** LED is on before proceeding to the next step.

Step 5. Turn off the SEL-3505 and remove the jumper.

Step 6. Follow *SEL-3505 Reassembly on page 2.12*, reinstall all connectors and energize the unit.

Appendix A

Firmware and Manual Versions

Firmware

Determining the Firmware Version in Your Device

Appendix A in the ACSELERATOR *RTAC® SEL-5033 Software Instruction Manual* lists the firmware versions, a description of modifications, and the instruction manual date code that corresponds to firmware versions.

Instruction Manual

The date code at the bottom of each page of this manual reflects the creation or revision date. *Table A.1* lists the instruction manual release dates and a description of modifications. The most recent instruction manual revisions are listed at the top.

Table A.1 Instruction Manual Revision History (Sheet 1 of 3)

Revision Date	Summary of Revisions
20210720	Section 1 ➤ Updated *Radiated and Conducted Emissions* in *Specifications*.
20200224	Section 1 ➤ Updated *Specifications*.
20190216	Appendix A ➤ Updated for firmware version R144-V2.
20180629	Section 1 ➤ Updated *Time-Synchronization Source* to include additional PTP configuration information.
20180330	Section 1 ➤ Updated *Specifications*.
20170505	Section 1 ➤ Added File Transfer Protocol to *Communications Protocols*. ➤ Updated *Specifications*. Appendix B ➤ Added *RTAC Web Upgrade Procedure*.

Table A.1 Instruction Manual Revision History (Sheet 2 of 3)

Revision Date	Summary of Revisions
20170220	**Preface** ➤ Updated *Safety Information*. **Section 1** ➤ Updated *Options*. ➤ Updated *Specifications*. **Section 2** ➤ Updated *Device Placement*. ➤ Updated *Connections*.
20170109	**Section 1** ➤ Updated *Automation and Control*. ➤ Updated *Communications Protocols*. ➤ Updated *Specifications*.
20160624	**Section 1** ➤ Updated *Features* and *Applications* for client/server and PTP information. **Appendix B** ➤ Added point-release information.
20160115	**Appendix A** ➤ Corrected reference to *ACSELERATOR RTAC SEL-5033 Software Instruction Manual*.
20150904	**Section 1** ➤ Revised for addition of IEC 61850 MMS server. ➤ Updated *Specifications*.
20150417	**Section 1** ➤ Added reference to the exe-GUARD® security feature. ➤ Updated *Features*. ➤ Updated *Models, Options, and Accessories*. ➤ Added figure references. ➤ Updated *Specifications*. **Section 2** ➤ Updated *Device Placement*. ➤ Updated *Connections*. **Section 3** ➤ Removed power source scale information under *Self-Test*. ➤ Updated *Using Online Debug*. ➤ Updated *Figure 3.4: Forced Values Window* and *Figure 3.6: Watch Window Detail* (was 3.5) and added *Figure 3.5: Create a Watch Window*. **Appendix B** ➤ Added Chrome™ to browsers in *Step 11* of *Upgrade Procedure*.
20150126	**Preface** ➤ Updated *Safety Information*.
20140714	**Section 1** ➤ Updated Automation and Control in Features. ➤ Updated Specifications. **Section 3** ➤ Updated *Run-Time Diagnostics* in *Self-Test*. ➤ Added *Tag Cross-References* in *Using Online Debug*.

Table A.1 Instruction Manual Revision History (Sheet 3 of 3)

Revision Date	Summary of Revisions
20140616	General ➤ Updated manual throughout to include SEL-3505-3. Section 2 ➤ Included 12–24 and 24–48 Vdc power supply options. Appendix B ➤ Added project conversion to firmware update instructions.
20130628	Section 1 ➤ Added protocols to Data Concentration and Protocol Conversion in *Automation and Control* in *Features*. ➤ Added protocols to *Communications and Protocols* table in *Features*. ➤ Updated *Figure 1.5: System Control and Synchrophasor Data Concentration*. ➤ Updated *Figure 1.8: Automate and Integrate With Communication and Logic*. ➤ Updated *Specifications*. Section 2 ➤ Updated *Communications Ports* in *Connections*. ➤ Updated *Figure 2.8: EIA-485 Typical Two-Wire Connection* and *Figure 2.9: EIA-485 Typical Four-Wire Connection*. ➤ Added *Modem Usage* in *Connections*. ➤ Updated steps in SEL-3505 *Disassembly* and SEL-3505 *Reassembly* in *Field Serviceability.* Section 3 ➤ Updated *Force Tags* in *Using Online Debug*. ➤ Updated *Table 3.1: Self-Test System Tags*. ➤ Updated *Troubleshooting* table. ➤ Updated *Figure 3.8: Factory Reset via Web Interface*.
20130117	Section 1 ➤ Updated to remove references to modulated IRIG signals. ➤ Changed references to digital output to reference solid state relay. ➤ Updated *Specifications* to include digital output and also fiber specifications. Section 2 ➤ Corrected reference to which of the two serial ports have EIA-232/EIA-485 software select ability. ➤ Corrected reference to PIN 1 power, which is not supported. ➤ Clarify jumper settings for password override and system reset.
20121121	General ➤ Updated manual throughout for greater clarity in text and figures.
20120720	➤ Initial version.

This page intentionally left blank

SCHWEITZER ENGINEERING LABORATORIES, INC.
2350 NE Hopkins Court • Pullman, WA 99163-5603 U.S.A.
Phone: +1.509.332.1890 • Fax: +1.509.332.7990
selinc.com • info@selinc.com

ACSELERATOR RTAC®
SEL-5033 Software

Instruction Manual

20210713

SEL SCHWEITZER ENGINEERING LABORATORIES

PM5033-01

Table of Contents

Section 1: Getting Started

Section 2: Communications

Section 3: Tag Processor

Section 4: Engineering Access

Section 5: Web HMI and Reports

Section 6: Compare ACSELERATOR RTAC Configurations

Section 7: Security and Account Management

Section 8: Time Synchronization

Section 9: Custom Logic

Section 10: Extensions

Section 11: Testing and Troubleshooting

Appendix A: Firmware and Manual Versions

Appendix B: IEC 61131-3 Programming Reference

Appendix C: Firmware Upgrade Instructions

Appendix D: Configure Dynamic Disturbance and Fault Recording Systems

Appendix E: EtherCAT Overview

This page intentionally left blank

List of Figures

List of Tables

This page intentionally left blank

SECTION 1

Getting Started

Overview and Features

The ACSELERATOR RTAC SEL-5033 Software is an intuitive, easy-to-use application designed to configure the SEL Real-Time Automation Controller (RTAC) family of products, including the SEL-2240 Axion®. The main principles behind using the software involve the following:

- ➤ Creating a project for each RTAC or SEL-2240 system. An SEL-2240 system can encompass more than one node. You will need one ACSELERATOR RTAC project for each SEL-2241 RTAC module you use.
- ➤ Inserting and configuring various protocols for devices that connect to IP and serial communications ports on an RTAC or I/O modules for the SEL-2240. Each device name is user configurable with a maximum of 100 characters.
- ➤ Mapping tags (data) from intelligent electronic devices (IEDs) or other data sources to data recipients (remote clients) via the Tag Processor. A tag is a 100 maximum character limit name that specifies a data structure (such as a binary input) or piece of data within that structure (such as the Boolean value).
- ➤ Writing logic and math statements that create outputs based on tag value inputs. An example would be ORing several binary inputs to create one binary value.
- ➤ Writing logic programs/functions in the following IEC 61131 languages:
  - ➢ Structured Text
  - ➢ Continuous Function Block
  - ➢ Ladder Logic Diagram
- ➤ Creating a device configuration to use as a template for other similarly configured devices.

NOTE
The ACSELERATOR RTAC SEL-5033 Software configures all devices that are included in the RTAC family, including the following:

SEL-3505	SEL-3505-3	SEL-3530	SEL-3530-4
SEL-3532	SEL-3555	SEL-2240 Axion	

Throughout this manual, RTAC refers to all devices in the RTAC family. Any unique instructions for a particular model will be clarified.

Getting Started

After launching ACSELERATOR RTAC, provide the login and database information necessary to access the default ACSELERATOR RTAC database and begin configuring your RTAC project. The login and database fields identify which ACSELERATOR RTAC database you are using. The software saves this set of information as a connection record with a unique connection name (in the following example, this name is RTAC Default Connection). Initially, you can leave all the fields in the login screen at their default values.

The ACSELERATOR RTAC software can use alternative connection records to access different project databases. To access another database, create a unique name in the **Connection Name** field, fill out the **Server** name, **Port** number, **Database** name, **User Name**, and **Password**. If the ACSELERATOR RTAC software finds the database, and if the username and password are correct for that database connection, it will create a new connection record automatically. This login connection name will then be the default connection the next time you start ACSELERATOR RTAC. Select any previously used connection records by using the drop-down list button on the **Connection Name** field. Once you have chosen a connection record name, the software will populate all the other fields (except the password) automatically in the login screen. Enter the password necessary to log in to the connection you have selected.

Initially, two user accounts exist for the default database. One user account has a username of admin and a password of TAIL. The second user account has a username of engineer and a password of OTTER. The administrative user has full access to all software features, including adding/deleting user accounts and unlocking projects that are locked for editing by another user. The engineer account user can change his or her own password but cannot unlock locked projects. SEL recommends that you change default account usernames and passwords immediately to secure your application. Until you secure the application, the software will display a warning indicating that the database is not secure. For more information regarding the updating of accounts, see *Section 7: Security and Account Management*.

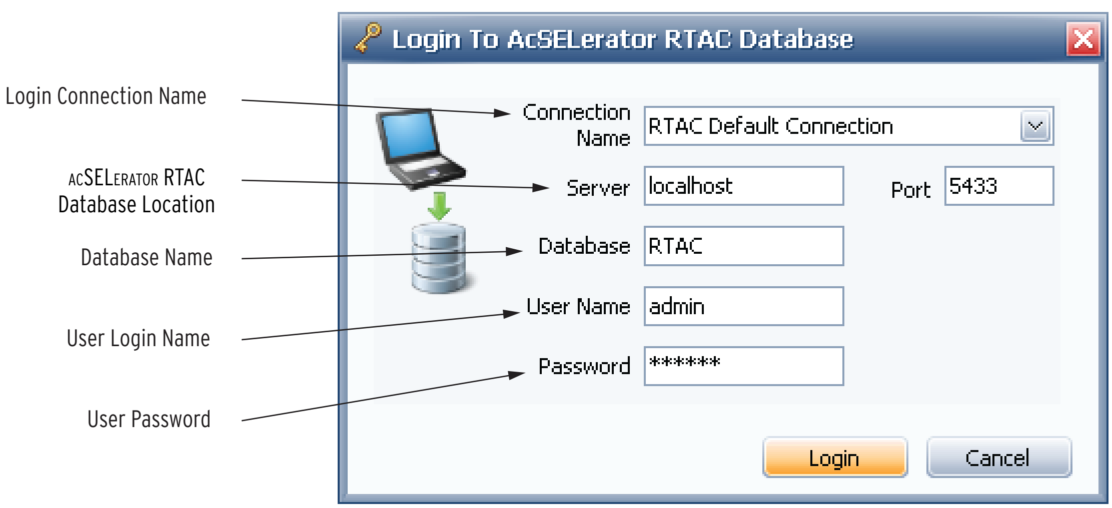


Figure 1.1 ACSELERATOR RTAC Login Screen

Use the start page that appears after you log in to the ACSELERATOR RTAC database to view existing projects or create new ones.

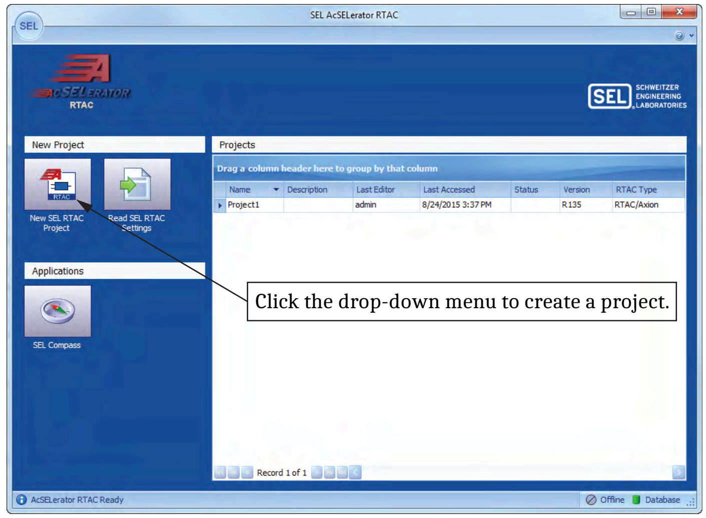


Figure 1.2 ACSELERATOR RTAC Start Page

Start Page Sections

New Project Section

Select **New RTAC Project** to create a new ACSELERATOR RTAC project.

Select **Read RTAC Project** to read the project from a previously configured RTAC into a new project. You must be connected to an RTAC to read its project. See *Save and Download a Project on page 10* for more information.

Applications Section

Click **SEL Compass** to update ACSELERATOR RTAC software.

Messages Section

View updated device definitions. The Messages heading is hidden when no messages are available.

Projects Section

View the list of existing projects in the current database.

Open an existing project by double-clicking on the project.

Back up all projects to a .zip file. Restore all projects from a .zip file.

Right-click on a project to perform the following actions:

- Rename the project.
- Convert the project to different firmware or to a different RTAC type.
- Import a project to a file.
- Export a project from a file.

Creating a New Project

A project contains communications settings, tags, and logic for a particular RTAC configuration. You can create a new project or use a project you saved previously as a starting point for a new project.

Click the **New SEL RTAC Project** icon in the start page or select **New** from the menu in the SEL application button to access the project creation dialog box.

To quickly create a new default project for the latest RTAC version, select the **Create** button shown in *Figure 1.3* to open the project view. The **RTAC Firmware Version** pane provides a method to create projects for earlier RTAC versions. ACSELERATOR RTAC selects the latest firmware version by default. If you select a previous version, ACSELERATOR RTAC will remember that selection the next time you create a new project.

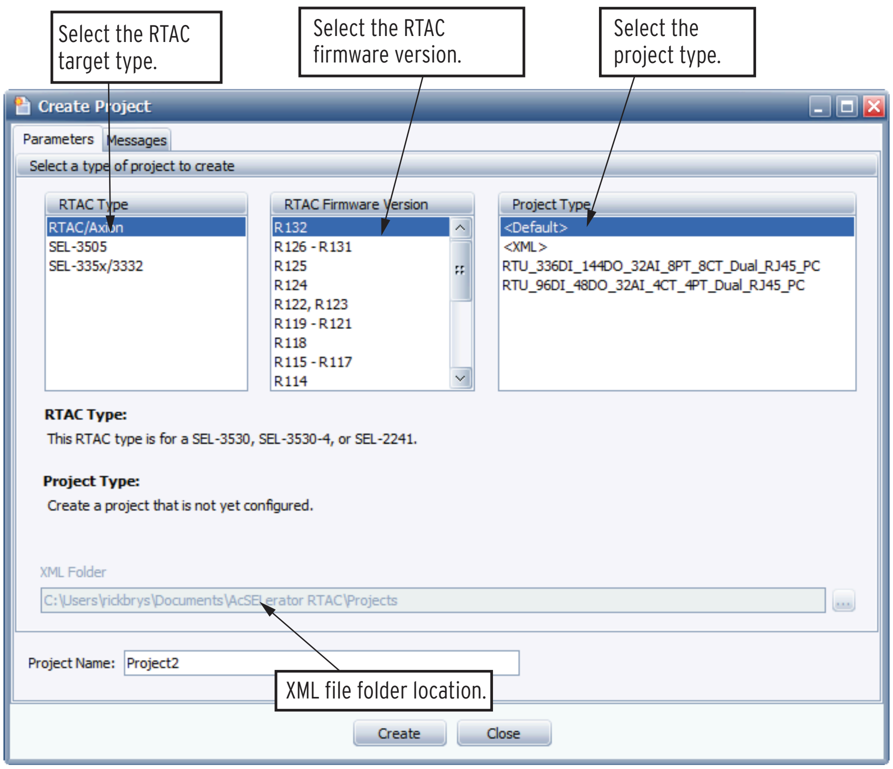


Figure 1.3 ACSELERATOR RTAC Project Creation Interface

The default ACSELERATOR RTAC project template includes an RTAC (system tags and diagnostics), but no pre-configured client or server protocol connections. Select the default project category and default project template to create and open a default project. If necessary, enter a custom project name. If you have previously exported items from an RTAC project in XML format, you can select **XML** under **Project Type** to create a new project using those XML files.

ACSELERATOR RTAC also provides some Axion example projects in the Project Type column that include common settings for certain types of applications. Click on an RTU project type name from the list to see a project description. Provide a custom project name, if necessary, and select Create to open the example in the project view for editing. Refer to *RTU Project Examples on page 239* for more information.

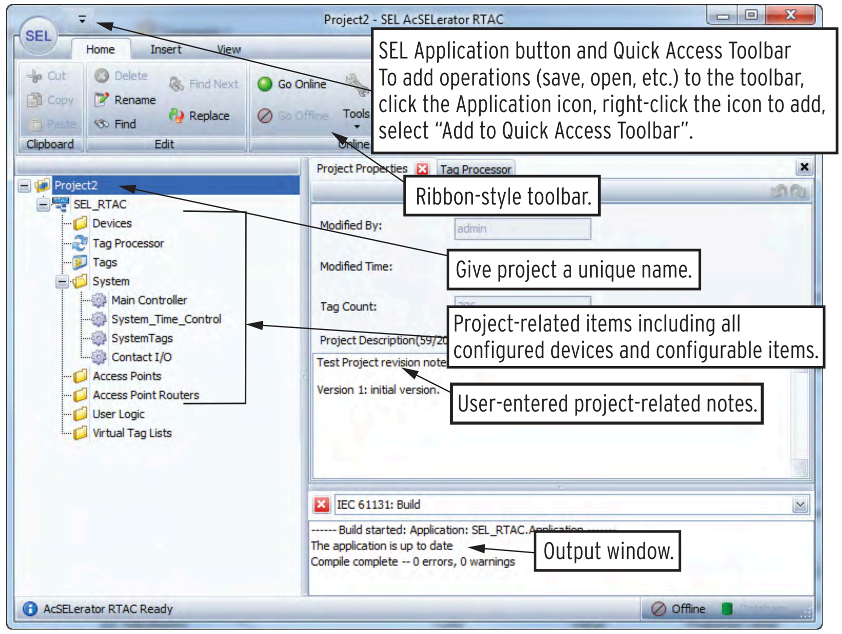


Figure 1.4 Project View

Navigate this screen by using the ribbon menu and the configurable **Quick Access Toolbar** at the top. The ribbon menu has four tabs:

➤ **Home**: Perform standard editing functions such as copy, paste, find, replace, etc. Also contains online tools including online/offline buttons, communications monitoring (Comm Monitor), and the Tools menu for user logic debugging options.

➤ **Insert**: Add a new intelligent electronic device (IED) or remote polling device, add new SEL-2240 I/O modules, save or retrieve a custom device configuration, add an access point router, or create custom IEC 61131-3 user logic.

➤ **View**: Open/close window panes, view reports, return all docked windows, and change skins for personal preference.

➤ **Tag Processor**: Append, insert, and delete lines in the Tag Processor as well as copy, paste, and delete items. This menu is only visible when the Tag Processor window is in focus.

The following descriptions are for project items in the left window pane of *Figure 1.4*. Right-click the project name to rename it. Right-click on **SEL_RTAC** to add any of the following features. You can also add folders in the device tree from the right-click menu. Organize devices and connections by dragging/dropping them into the custom folders as needed.

➤ **Project2**: The configurable name of the current project. This must be a unique name in a given database.

➤ **SEL_RTAC**: The name of the device in the project.

➤ **Devices**: A blank folder where you can place devices configured for this project (IEDs, remote client connections, etc.).

➤ **Tag Processor**: Tag mapping tool used to map data from IEDs to polling client devices.

➤ **Tags**: View of all alias tags. When online, view of task timers, task controller blocks, and program organizational unit (POU) pins.

- **System**:
  - **Main Controller**: Set cycle time, watchdog time, and execution order of tasks or program organizational units (POUs). For example, alter the execution order to ensure that all defined protocols collect data before a custom logic program uses the data.
  - **System_Time_Control**: Configure device time-source information such as IRIG-B, PTP, and NTP. Configure UTC time and Daylight-Saving Time settings.
  - **SystemTags**: Configure and view all system (non-device or user logic) tags here. Open **SystemTags** when online to view and/or force present values of all system tags. Also configure the C37.118 Data_Rate or rate of data transmission. *IEEE C37.118 Synchrophasors on page 98* explains this further.
- **Contact I/O**: Configure RTAC contact inputs and outputs, including tag names, pickup and dropout delays, and default values.
- **EtherCAT I/O Network**: Visible once you have inserted an Axion node from the insert menu. Configure and view SEL-2240 nodes and modules. View network messages in Online mode.
- **Access Points**: A blank folder where you can place names of access point configurations, which are used with access point routers in creating transparent connections.
- **Access Point Routers**: A blank folder where you can place names of transparent connection configurations.
- **User Logic**: A blank folder where you can place user-defined logic blocks. This includes IEC 61131-3 programs, functions, function blocks, user-defined global variable lists (GVLs), and data types.
- **Virtual Tags**: A blank folder where you can place virtual tag lists. Virtual tags are global variables that you can use anywhere in an RTAC project. You can create as many as 30 virtual tag lists.

Task Cycle Time

The RTAC executes all tasks each task cycle time interval to which they are assigned. Configured task cycle times that are evenly divisible into one second are synchronized to execute at the top of each second of the system time. For example, if the processing cycle is configured as 100 ms, the first task cycle will be at the top of the second. Subsequent task cycles are executed at 100 ms intervals until the top of the next second, at which time the next processing cycle is executed. Configured task cycle times that are not evenly divisible into one second are not executed at the top of the second but are still locked to system time. By default, all tasks run in the main task cycle interval.

Perform the following steps to assign certain tasks to the automation task cycle so that they can run independently and at a faster interval than the main tasks.

Step 1. From the **Main Controller** tab, click the task in the task order list.

Step 2. Click the **Move To** button to move the task from the main task cycle to the automation task cycle.

Step 3. Click the **Automation** tab to configure the automation task cycle parameters.

With the present software, you cannot directly map tags from a task that is in the automation task cycle to a task that is in the main task cycle. Each tag value, control, and logical operation cannot update faster than the task cycle time. While online with the RTAC, you can view various task timers, as well as an overall task timer, in ACSELERATOR RTAC by clicking on **Tags** under the device tree. The task timers indicate how long, in microseconds, the tasks are taking to run. You should configure the main task cycle time to be at least 140 percent of the overall task timer. This provides enough time for the tasks to run, with added time for the RTAC web interface and HMI access. If you configure the RTAC main task cycle time less than what is actually required to run all the defined tasks, the RTAC will run over the task cycle time. For example, if you configure the RTAC to run every 20 milliseconds, but the total task timer indicates the RTAC tasks are using 50 milliseconds to complete, the RTAC will not drop tasks, but will use time from subsequent task cycles to ensure each task runs to completion. The RTAC will therefore not complete every task in the configured 20 milliseconds, and web access performance will be degraded.

The automation task timers are not contained in the **Tags** tab. To view the task times for tasks that are located in the automation task, click **Tools** on the **Home** ribbon while online. Select **Watch Values** and open a watch window. In the watch window, type the following:

SEL_RTAC.Application.Automation_SystemTotal_task_timer

This contains the automation task timers. Use this information to adjust the automation processing cycle. The information regarding adjusting the main task cycle for optimal performance applies to adjusting the automation task cycle as well.

NOTE
If an SEL-2245-4 or SEL-2245-42 CT/PT module is the source of any C37.118 data, the task cycle time will lock to the configured Nominal Frequency. The task cycle time will be 20 ms for 50 Hz and an average of 16.667 ms for 60 Hz. The 16.667 ms average is achieved by running the task cycle time at 17 ms for two cycles and then 16 ms for one cycle.

A rising edge change triggers control operations in the RTAC per cycle interval. This implies a control value that is high must drop low and become high again for a second control operation. Because the RTAC only updates values each cycle-time interval, the control state change must occur each interval. For example, a change in control state from high to low to high during one cycle-time interval will not trigger a new control value because only the last value (high) will be stored. It is possible, therefore, to configure the cycle-time interval in such a way to cause the RTAC to miss rapid control changes. As a rule, set the cycle-time interval such that control operations are not executed at a rate greater than half the cycle-time interval.

Startup Time

Initial startup time of the RTAC will vary depending on the version of the RTAC used. The initial startup time includes initializing all of the operating system, exe-Guard®, and initialization of communication links among other system tasks. This corresponds to when the RTAC ENABLED LED turns green and does not include the time to establish communication to downstream devices. Fully establishing communication with downstream devices requires additional time based on the configured communications parameters and can be adversely impacted if the RTAC is configured with a high burden. *Table 1.1* shows expected initialization times for each RTAC type.

Table 1.1 Typical RTAC Startup Time

RTAC Type	Time to Enable
SEL-3530, SEL-3530-4, SEL-2241	~2.5 minutes
SEL-3555	~45 seconds

Application Button

Click the **SEL Application** button to execute file and database functions.

Add frequently used functions by clicking on the **SEL Application** button, right-clicking on the function you want added, and selecting **Add to Quick Access Toolbar** from the pop-up menu.

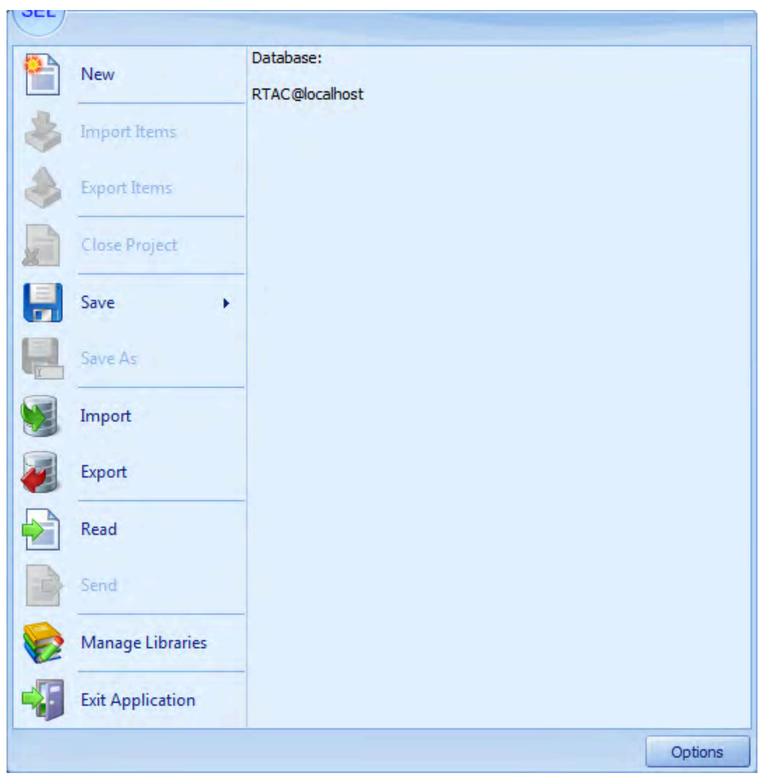


Figure 1.5 Application Button and Quick Access Toolbar

NOTE

User-created names must conform to IEC 61131-3 standards: only use letters, numbers, and underscore; always alert with a letter; and never have two consecutive underscores. If a name is invalid, a will appear beside it.

- **New**: Create a new project for an RTAC.
- **Import Items**: Import individual project items from previously exported XML files. This is only available while a project is open.

- **Export Items**: Save selected project items as XML files. This is only available while a project is open.
- **Close Project**: Close current project. This will not close ACSELERATOR RTAC.
- **Save/Save As**: Save current project.
- **Import**: View or edit projects that you exported previously. This is only available from the start page.
- **Export**: For the selected configuration, generate a configuration file that can be imported into any ACSELERATOR RTAC database. This is only available from the start page.
- **Read**: Read an existing configuration from a connected RTAC. This is only available from the start page.
- **Send**: Send a compiled project to a connected RTAC.
- **Manage Libraries**: Install previously created libraries for use in ACSELERATOR RTAC projects.
- **Exit Application**: Close ACSELERATOR RTAC. Be certain that you save and close projects before exiting.

Docking Windows

From the Project View, you can open configuration windows by clicking devices in the device tree. Most of these windows can be undocked from ACSELERATOR RTAC and placed anywhere on your computer workspace. To do this, click the top bar, which bears the name of the window in focus, and drag it anywhere on your workspace. See *Figure 1.6*.

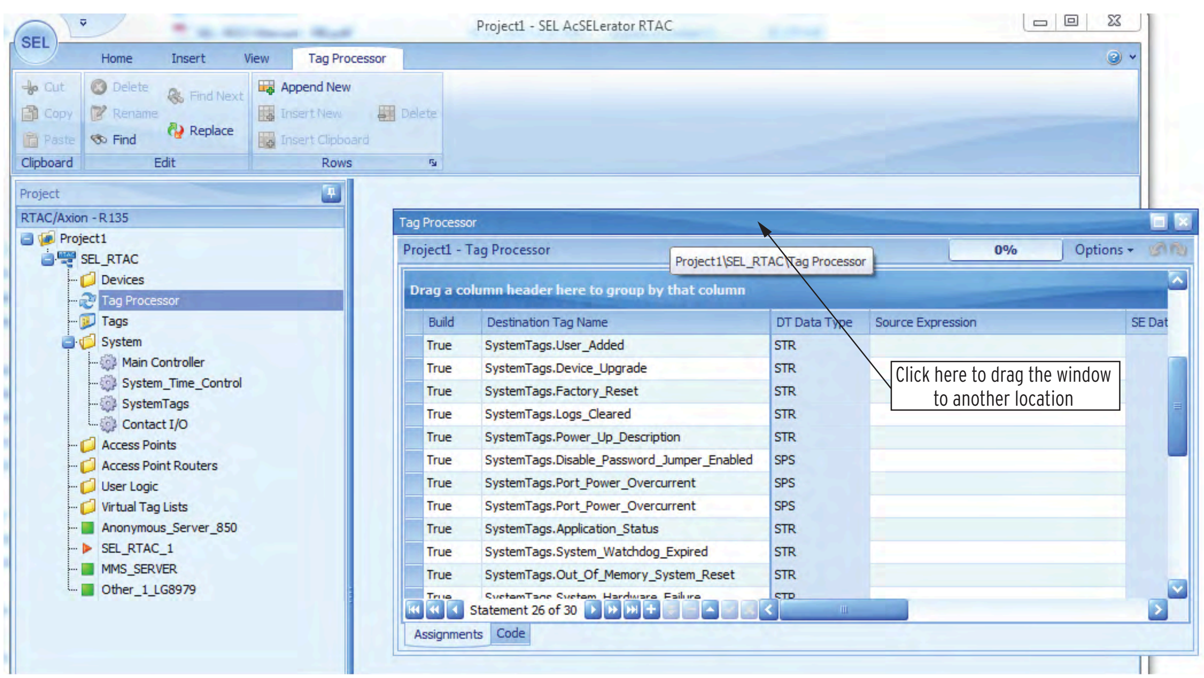


Figure 1.6 Docking Windows

Each window will retain the project and window names. To return the window, close the window by clicking the X in the upper right-hand corner or click **View > Reset Window Layout** to restore all windows to your project. See *Figure 1.7*.

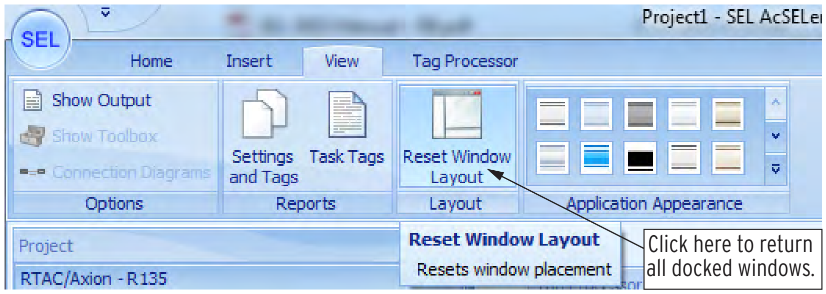


Figure 1.7 Reset Docked Windows

Key Shortcuts

As with most Microsoft Windows-based programs, there are a number of keystroke shortcuts you can use to copy, paste, find, and perform other operations. From the **Home** ribbon, or within the SEL action button, hover your mouse over the operation to display a tooltip window that describes the operation and the keystroke shortcuts.

Use **<Ctrl+Alt+click>** in structured text programs to highlight, copy, paste, and delete columns of text. The editor replicates anything you type in a highlighted column in every row of that column.

Save and Download a Project

After completing the RTAC configuration, save and download the project into the RTAC. Before downloading the project, you must first set up a user account on your RTAC through the web interface. See *Section 7: Security and Account Management* for details.

Step 1. Ensure proper physical connection with either Ethernet or USB to the RTAC. See *Section 2: Installation* in the *SEL-2240 Instruction Manual* for physical connection requirements.

Step 2. Click the **Save** icon, accessible either from the **Quick Access Toolbar** or by clicking the **SEL Application** button, to save and compile the project. Alternatively, you can press **<Ctrl+S>** to save the project.

NOTE
The save operation first checks the project for syntactic errors before storing the project in the database. The normal save operation does not, however, check for tags that are referenced in the automation and main task. This is referred to as crosstask checking and is done only when you select the save with cross-task checking option, or when you send the project to the target.

Step 3. Check the output window for compile errors or warnings. Resolve errors before continuing. Double-click on the error line in the output window to have the program direct you to the item causing the error.

Step 4. Click the Send icon to download the new configuration and logic to the RTAC.

Step 5. Enter the RTAC IP address in the RTAC Address field, username, and password then press **Next**.

You can use the same default connection name for each time you connect to an RTAC. Alternatively, you can create a unique connection name for each unique IP address of each RTAC residing on the network. To create a new connection, type a unique connection name and IP address, enter the username and password, then press **Login**. The connection information is stored in the ACSELERATOR RTAC database. The next time you attempt to go online with a project, the most recently used connection information is displayed. You can also select other previously used connections from the **Connection Name** drop-down menu.

Note that the RTAC and ACSELERATOR RTAC software usernames and passwords are not necessarily the same. The RTAC web interface maintains the RTAC username and password information. See *Section 7: Security and Account Management* for more information on password management.

Step 6. Optional: Click the **Options** tab and indicate if you want to send the complete project or only IEC 61131 logic changes.

NOTE
If a protocol is disabled during a settings download through custom logic or a direct transparent connection, it will not re-enable until after the RTAC restarts with the newly downloaded settings.

You can do quick iterative testing of IEC 61131 logic by selecting **Send only custom IEC 61131 logic**. This option does not reset the RTAC and thus allows a form of online editing of your custom logic. Other settings, such as those in protocol configurations, are not sent to the RTAC when this is selected, so it is important to resend the entire project when you are finished testing IEC 61131 logic changes.

When you are satisfied with the IEC 61131 logic, always select **Send Complete Project** and resend the entire project. This puts the RTAC into service and allows you to read the entire project out of the RTAC at a later date if necessary.

Step 7. When the download is complete, click **Finish**. Upon completion of RTAC programming, the RTAC automatically enters online mode.

Step 8. (This step only applies to projects that include SEL Axion I/O modules and EtherCAT protocol.)

After the download completes and the project starts running, the RTAC will automatically configure and start the EtherCAT network. No user selections will be required during the configuration process. All other protocols and user logic in the RTAC will begin to run as soon as the project download is complete.

Monitor the configuration status by watching the messages in the software status bar (bottom of software window). Configuration will take 10–15 seconds per power coupler and I/O module on the network. A network with 60 modules can take 12 minutes to fully configure. Configuration times will be longest during the initial project downloading sequence. Subsequent project changes will configure more quickly.

The sequence of configuration messages are as follows:

➤ "Beginning EtherCAT boot sequence..."
➤ "Reading EtherCAT settings..."
➤ "Addressing EtherCAT network..."
➤ "Verifying EtherCAT network..."

Step 9. While online, you can view data, force values, and perform other online functions, as necessary.

Step 10. After you finish viewing data, etc., press ● to go offline. Note that all logic and configuration will continue to run and that the RTAC application is now in service.

If any tag values remain forced when you attempt to go offline, a dialog box will ask if you want to unforce them. If you go offline because of inactivity, the values are automatically unforced and returned to their original state.

Go online anytime by selecting the ● icon. Upon your selecting the online icon, the RTAC software compares the currently open project with that in the RTAC. If these projects are the same, the RTAC will go online immediately. If the projects differ, ACSELERATOR RTAC will prompt you to send the currently open project to the RTAC.

See the testing and troubleshooting instructions for your RTAC model for online options such as forcing of data values, creating tag watch windows, and changing tag display modes.

Connection Directory

Starting in software version 1.31.147.xxxx, the connection directory will be preserved when a software upgrade occurs. The connection directory is now saved in the following folder location:

\ProgramData\Application Data\SEL\AcSELerator\RTAC\Connections

There are two files in this directory that save connections:

➤ SEL3530.csv: This file contains IP addresses to which RTAC projects can be sent. This list is accessible when sending RTAC projects.
➤ RTAC.csv: This file contains IP addresses of remote RTAC databases to access with ACSELERATOR RTAC. This list is accessible when logging into the RTAC database.

Connections can be added through ACSELERATOR RTAC or by directly editing the CSV files. These CSV files can be copied and moved between installations of ACSELERATOR RTAC.

Upload Projects From the Web Interface

Starting with firmware version R136, the RTAC supports a project-upload feature for the web interface. This feature allows you to send previously exported project files to the RTAC even when the ACSELERATOR RTAC software is not installed on the computer. This function is available for Administrator, Engineer, and HMI Operator user accounts. See *Section 7: Security and Account Management* for more information on web-interface user-account privileges.

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **HTTPS://** followed by the RTAC Ethernet address.

Step 4. Log into the RTAC web interface.

Step 5. Select **Project Upload**.

Step 6. Select **Browse**, locate the desired project file, and click **Open**.

Step 7. Select **Upload**. Note that once successfully uploaded, the project is loaded into the internal memory of the RTAC as an inactive project.

Step 8. Once uploaded, the project name appears in the list. Click **Activate** to apply the settings to the connected RTAC.

NOTE
If you have created a custom password for the exported project, the RTAC will prompt you to enter the password before it is enabled in the RTAC. See Section 7: Security and Account Management for more details on encrypting passwords.

If there is insufficient storage space on the RTAC to store an uploaded file, the error message, "ERROR: Insufficient storage space," will display.

An attempt to upload a file with an extension other than .exp results in the software displaying the error message, "ERROR: Invalid Project File."

You can use the applicable buttons to rename or delete projects that have been uploaded to the **Project Upload** page. To organize the project list, click the **Create Directory** button to create a new folder. Add existing projects in the list to the directory by left-clicking and dragging the project file icon  into the folder.

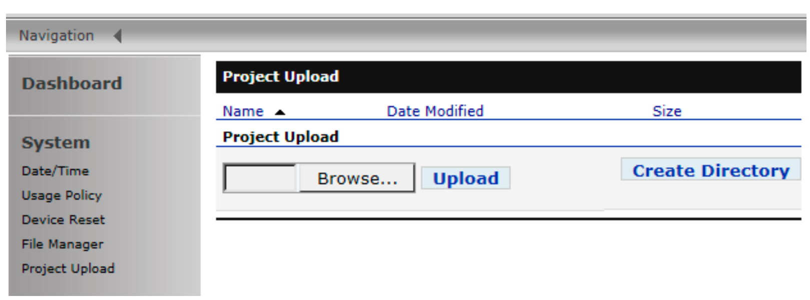


Figure 1.8 Project Upload Screen

Starting in R147, projects uploaded via the web interface can be activated with advanced settings that may be embedded with the uploaded project.

Open/Edit an Existing Project

You can open and edit an existing project by double-clicking on the project name in the ACSELERATOR RTAC start page. You can also open and edit projects not in the database that are presently in an RTAC, or projects that have been previously exported.

Use the **Read SEL RTAC Project** button on the start page to retrieve an existing project from a working RTAC.

Step 1. Ensure proper physical connection with either Ethernet or USB to the RTAC. See the installation instructions for your RTAC module for physical connection requirements.

Step 2. Click the **Read SEL RTAC Project** button. The login screen will appear.

Step 3. Enter correct RTAC login information and press **Next**.

Step 4. Enter a unique project name.

Step 5. Click **Next**.

Step 6. Download will commence. Download time will vary depending on project size.

Step 7. Press **Finish** when done. Download is now complete, and you can view or edit the project.

Use the import/export functions to save/retrieve an existing project.

Step 1. After you have created and saved a project, click the **Export** icon from the **SEL Application** button. Note that all projects must be closed before you can use the export/import functions.

Step 2. Create a unique name and press **Save**. The software will save the selected project as a file with extension .EXP. You can use email, USB memory stick, etc. to send this file to any PC running the same version of ACSELERATOR RTAC software.

Step 3. To open an exported file, select the **Import** icon from the **SEL Application** button.

Step 4. Browse and select the correct file name with extension .EXP and click **Open**. The software imports the project information in the .EXP file into the ACSELERATOR RTAC configuration database in the local machine.

Creating and Augmenting Projects With XML

Exporting Programs to XML

You can export individual POUs such as custom function blocks, client and server POUs, and tag processors to an Extensible Markup Language (XML) format. You can later import these individual XML files into existing projects or group them together to generate a new project.

NOTE

Manually editing the exported XML files can create settings that are not valid in an RTAC project. Be careful not to place invalid combinations of XML files in the same directory. For example, copying and pasting a program XML file would create an invalid condition because the program name is the same in both XML files even though the file names differ. The RTAC software ensures that all configured POUs have unique names.

To export your POUs to XML, perform the following:

Step 1. In an open project, select **Export Items** from the **SEL Application** button or right-click the desired POU and select **Export Items**.

Step 2. Select the items you want to export from the resulting pop-up window.

Step 3. Click the ... button choose a directory in which to store the exported features.

Step 4. Press **Export**.

The software exports each selected item into individual XML files and places the files in the export location directory you selected.

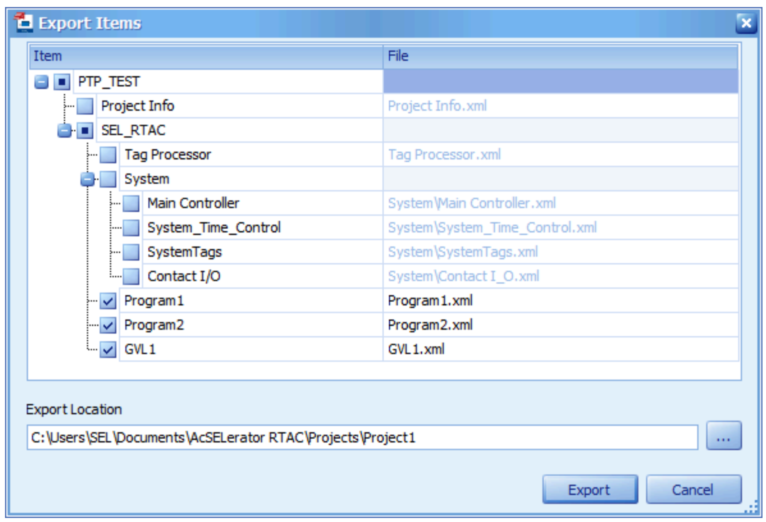


Figure 1.9 Export to XML Window

Alternatively, you can select one or more POUs from the project list, right-click, and select **Export Items**. The resulting pop-up window indicates with check marks all POUs you selected.

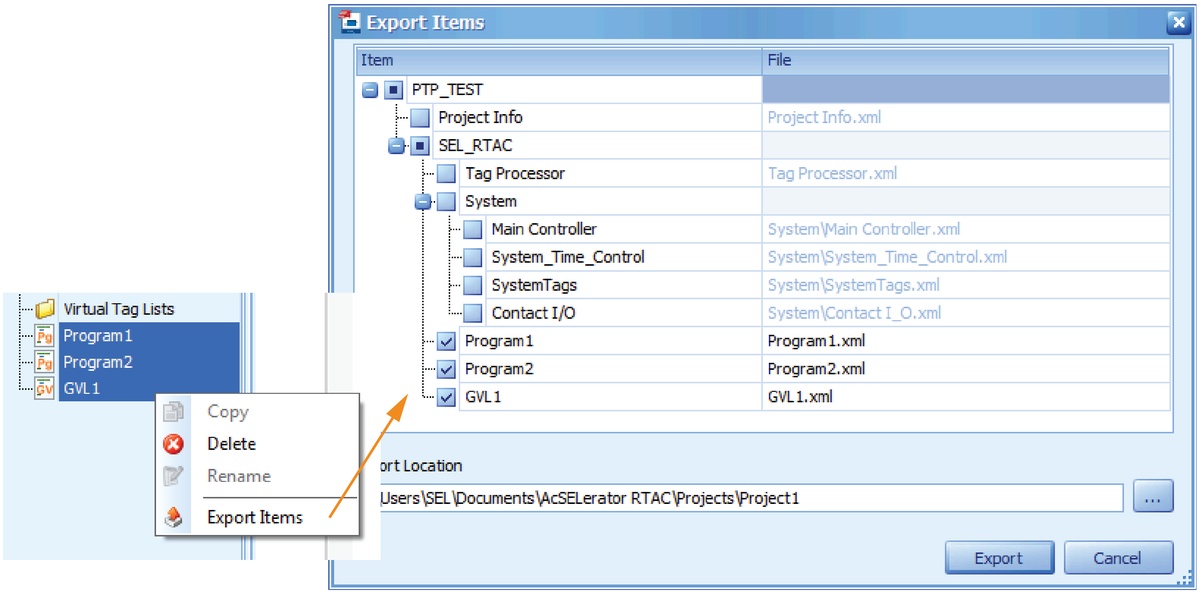


Figure 1.10 Export Selection to XML

Using Windows Explorer, you can create custom folders and place combinations of these XML files as necessary for future projects. You do not need to have every object from a project in the folder. Use only the objects you want. For example, in *Figure 1.10*, Project1 contains two custom IEC 61131 programs and a Global Variable List (GVL1) to use in multiple future projects. You can export each of the items you selected into a folder for use as a base for future projects that need those items.

To use the exported XML, you can either add individual XML files to an existing project or create a new project from exported XML files.

Modify Existing Projects With XML

ACSELERATOR RTAC software allows you to import existing XML projects. Perform the following steps to import XML POUs directly to an existing project.

Step 1. From an open project, select **Import Items** from the **SEL Application** button.

Step 2. Choose the folder that contains the XML file you want. See *Exporting Programs to XML on page 14* for instructions on generating the necessary XML files.

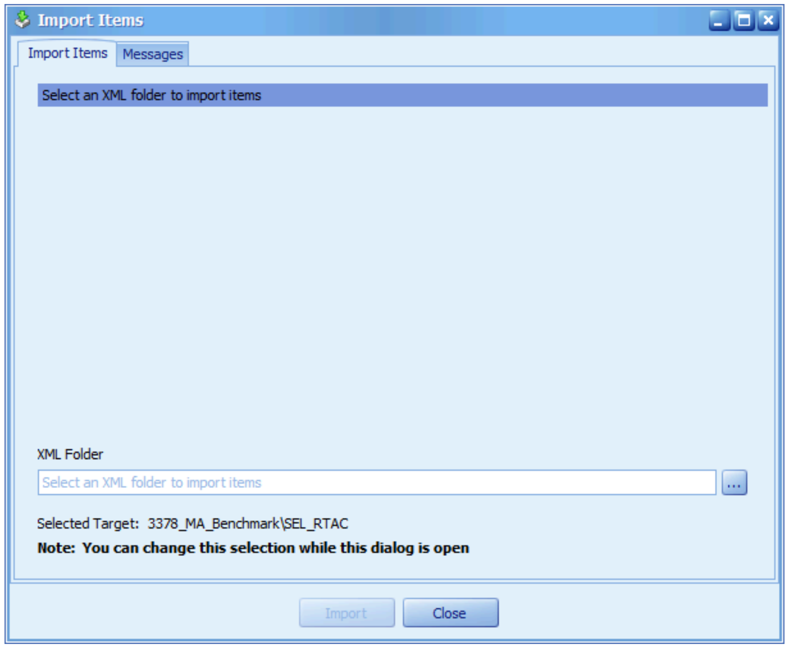


Figure 1.11 Import XML Window

Once you select the folder, the software displays all XML files available for importing.

Step 3. Select one or more of the available XML files and click **Import**.

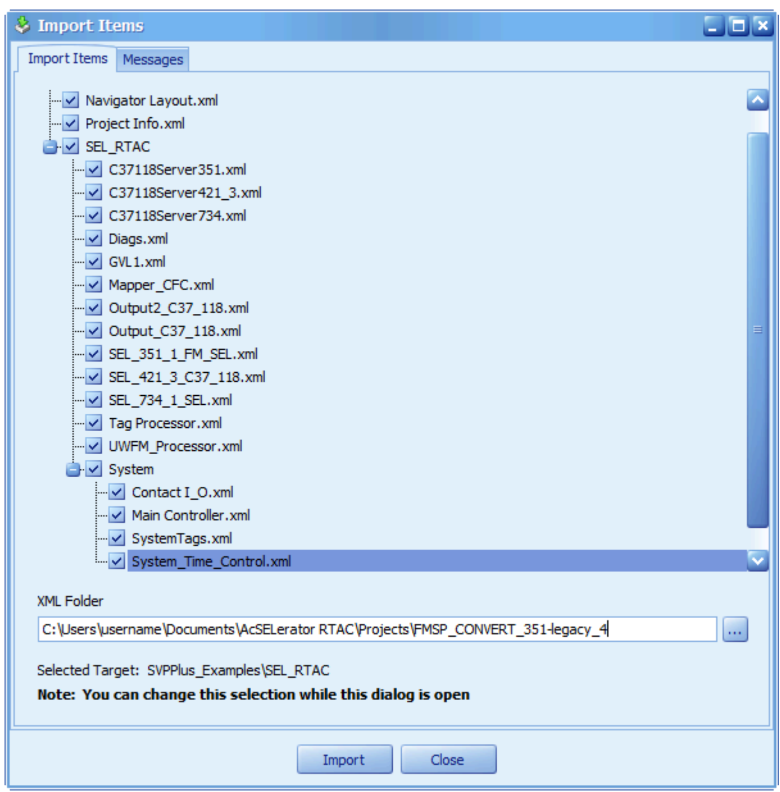


Figure 1.12 Import XML Items

Creating a New Project From XML

To use the exported XML files to generate a new project, click the **New SEL RTAC Project** icon while on the start page or click **New** from the **SEL Application** button. Select **XML** under **Project Type**, browse to the folder you previously created with the XML files, and click **Create**. The software uses all settings within the XML files in your selected folder to create a new project.

Backup/Restore Projects

The Backup Projects feature allows you to create a zipped file of all projects presently in your RTAC database.

Step 1. From the ACSELERATOR RTAC start page, right-click anywhere in the **Projects** area.

Step 2. Select **Backup Projects**.

Step 3. Browse to the location in which you want to save the archive.

Step 4. Name the file and press **Save**.

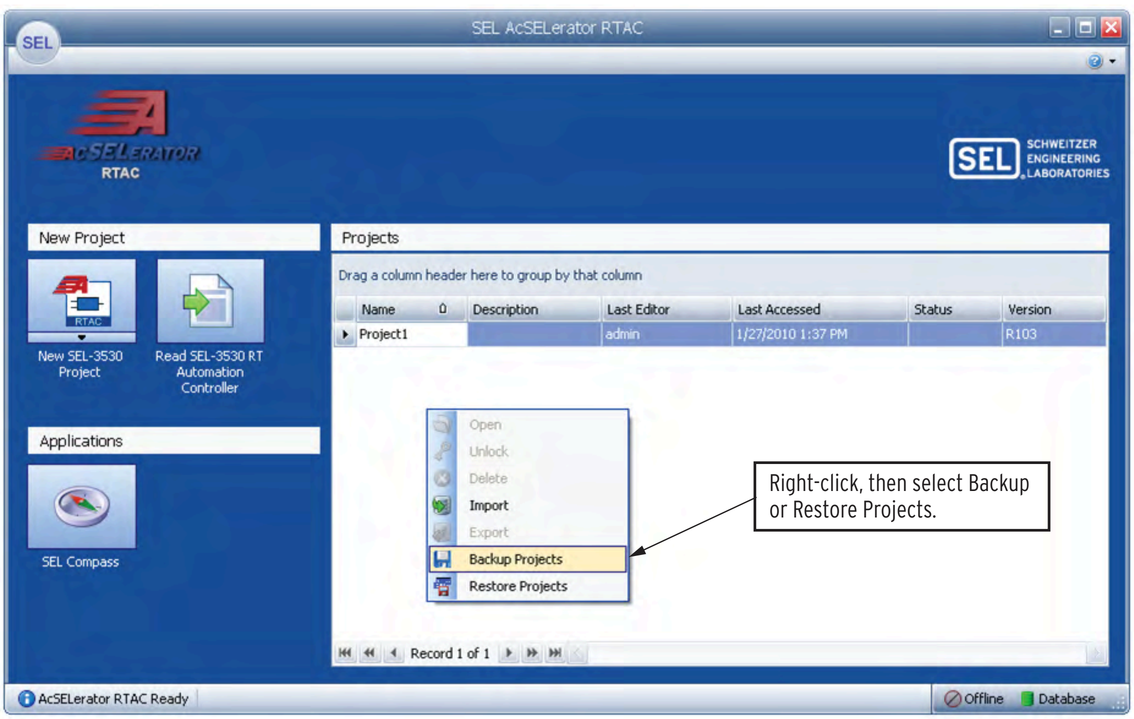


Figure 1.13 Project Backup and Restore

ACSELERATOR RTAC will create a zipped file of all projects. You can email this zipped file, store it on external media, etc. To restore the projects, use the same version of ACSELERATOR RTAC that you used to create the projects:

Step 1. From the ACSELERATOR RTAC start page, right-click in the **Projects** area.

Step 2. Select **Restore Projects**.

Step 3. Browse to the location of the archive .zip file.

Step 4. Select the .zip file and click **Open**.

ACSELERATOR RTAC restores all projects that are part of the archive, without overwriting existing projects. The restore process will resolve any name conflicts by appending a .0, .1, etc. to the end of the restored project names. For example, restore a project named MyProject when there is already a project with that name in the ACSELERATOR RTAC database. The software renames the restored project to MyProject.0. Restoring the project again will retain the two existing projects and create a newly restored project with the name MyProject.1.

Converting Older Projects

Use the project backup utility before updating ACSELERATOR RTAC to a newer version. Once the update is complete, restore projects, as necessary. You can use the projects as they are, or you can upgrade the database of the older projects to the latest database schema to be compatible with the latest RTAC firmware. The new schema can contain extra settings fields or data tags in the database definition file (DDF).

To convert an older project, follow these steps:

Step 1. Enter the **Convert SEL RTAC Project Settings** tool by doing either of the following:

- ➤ Right-click a project from the list of projects on the start page, then select **Convert Projects > By Project Type** or **By Project Version**.
- ➤ From the **SEL** button, select **Options > Convert Project Settings > OK**.

Step 2. Check the tab at the top of the conversion tool (by version or by type) to ensure use of the correct conversion mode.

Step 3. Select the project to convert.

Step 4. Change the **Convert?** field from **No** to **Yes**.

Step 5. If converting by type, change the **Converted Type** to the type of RTAC to which you want to convert.

Step 6. Click **Convert**.

The project will now have the new features from the latest database schema or, if converted to another type, those that can be loaded into the RTAC type to which it is converted. Open the converted project and ensure that the new schema settings are the values you need.

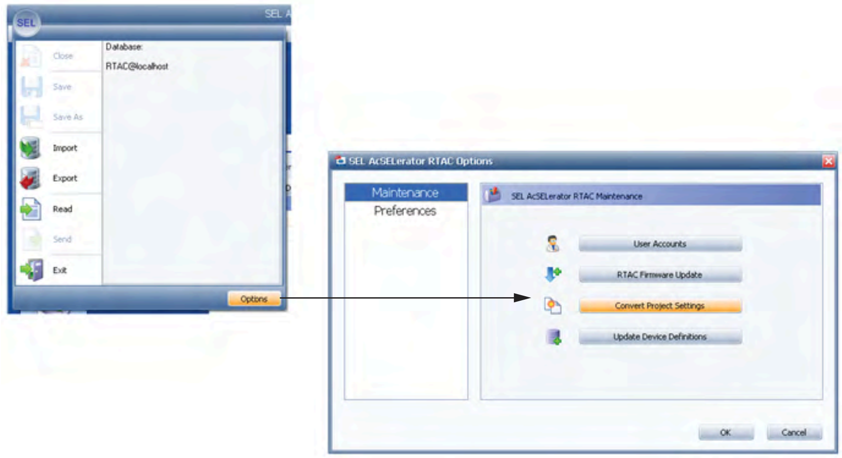


Figure 1.14 Convert a Project

RTAC Database on a Shared Server

In the standard configuration, the ACSELERATOR RTAC program installs and accesses the RTAC database on the local computer. As an alternative, you can configure one computer on a local area network (LAN) as a shared RTAC database server. In this configuration, each PC on the LAN can access RTAC projects from a common database that resides on the common server.

Perform the following steps to create and use a common RTAC project database on a shared server:

Step 1. Ensure the server machine and all computers that will access this machine are all visible on the same LAN. You can verify this by using the Ping command from the CMD prompt on one of the computers to ping each of the computers on the network.

Step 2. Install ACSELERATOR RTAC on the PC that is designated as the server. You must install ACSELERATOR RTAC software to install the RTAC database on that machine.

Step 3. Use Notepad or a similar text editing software to open the file postgresql.conf. This file is located at C:\Documents and Settings \All Users\Application Data\SEL\AcSELerator\RTAC\Database \data.

Step 4. On approximately line 56, change the following:

```
#listen_addresses = localhost # what IP address(es) to listen on;
```

to:

```
listen_addresses = * # what IP address(es) to listen on;
```

Step 5. Save the postgresql.conf file.

Step 6. From within the same directory, open the file pg_hba.conf.

Step 7. Search for and copy this line:

```
host all all 127.0.0.1/32 md5
```

Step 8. Paste the copied line directly after the line you just copied and make the following change:

```
host all all 10.202.13.0/24 md5
```

Note that the address 10.202.13.0/24 is just an example IP address. In place of 10.202.13.0/24, enter the address and CIDR subnet to match the address of the RTAC database server computer.

Step 9. Save the pg_hba.conf file.

Step 10. Restart the server computer.

Step 11. On each client machine, install ACSELERATOR RTAC software.

Step 12. On a client computer, start ACSELERATOR RTAC software.

Step 13. In the login dialog, you can optionally change the Connection Name to something that reflects a server connection. Typing a new Connection Name creates a connection record, as this section described previously, and will store the Server information for that connection.

Step 14. Change the Server field to be the IP address or hostname of the server computer but leave the Port field = 5433 and the Database field = RTAC. See *Figure 1.15*.

Step 15. Log in as normal. Note that the username and password login credentials you use in ACSELERATOR RTAC to access the server are stored on the server machine.

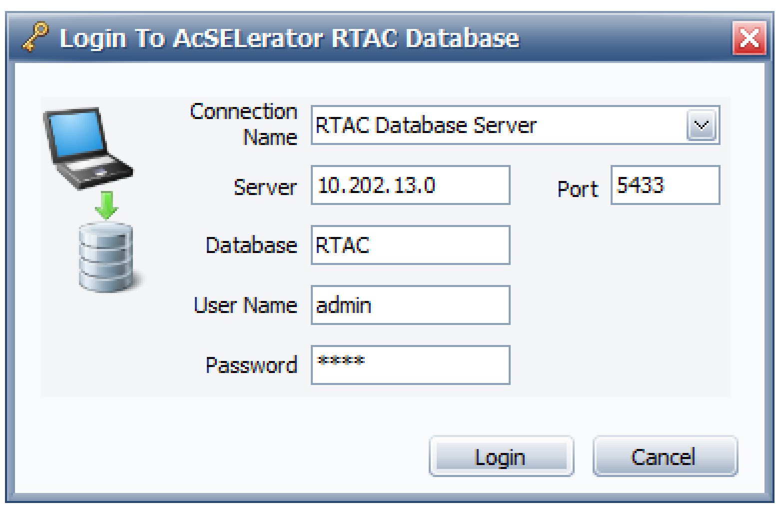


Figure 1.15 Logging In to Database Server

Software Install, Uninstall, and Backup

Startup Switches

You can modify the shortcut to ACSELERATOR RTAC to include various startup runtime switches. These switches provide features not available in the software in normal run mode. To modify the shortcut, right-click the **AcSELerator RTAC** shortcut and select **Properties**. Modify the line labeled **Target** to include the needed switch by adding */switchname* to the right of the quoted path name. Save the shortcut and run it to start ACSELERATOR RTAC using the new switch.

Change File Storage Location

ACSELERATOR RTAC uses the **My Documents** folder in Windows for program-related file storage. If your My Documents folder is located on a network drive instead of your local computer, modify the ACSELERATOR RTAC shortcut to include the startup switch */documents*, as shown in the following example:

> "C:\Program Files (x86)\SEL\AcSELerator\RTAC\RTAC.exe" /documents *I:\documents\location\MyDocuments*

where *I:\documents\location\MyDocuments* is the location of the **My Documents** folder.

Clean Install and Uninstall

The clean switch when moved will remove ACSELERATOR RTAC from the PC that it is currently installed on. It will search the following locations on the PC and remove all relevant ACSELERATOR RTAC files and registry entries.

NOTE
The locations shown are for Windows 10 64-bit. Other Windows versions will have similarly named folders, registry locations, and registry services.

- Folders
  - C:\Program Files (x86)\SEL\AcSELerator\RTAC
  - C:\ProgramData\Application Data\SEL\AcSELerator\RTAC
  - C:\Users\YOURUSERNAME\Local Settings\Application Data\SEL\AcSELerator\RTAC
  - C:\Users\Administrator\Local Settings\Application Data\SEL\RTAC.exe_URL_*
  - My Documents\RTAC
- Registry Locations
  - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RTACDB
  - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SEL\SEL-5033

    - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SEL\SEL-5033 DB
    - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SEL\SEL-5033 GAC
- Registry Services
    - RTACDB

Unlike the Documents switch, the clean functionality is not included with the shortcut for the ACSELERATOR RTAC software but rather with the actual setup or uninstall file. When run with the setup file, the clean functionality will remove all parts of the software from the PC before installing the new version of software.

Clean Install

Open the command prompt and browse to the folder where the RTACSetup.exe file is. Run the following command to have the software completely remove any remnants of ACSELERATOR RTAC before beginning to install the software. This will remove any projects that were automatically backed up in the C:\ProgramData\Application Data\SEL\AcSELerator\RTAC folder.

```
.\RTACSetup.exe /Clean
```

Clean Uninstall

Open the command prompt and browse to the folder where the uninstall.exe file is. Run the following command to have the software completely remove any remnants of ACSELERATOR RTAC. This will remove any projects that were automatically backed up in the C:\ProgramData\Application Data\SEL\AcSELerator\RTAC folder.

```
\uninstall.exe /Clean
```

NOTE
The uninstall file typically exists in C:\Program Files (x86)\sel\AcSELerator\RTAC\Installation. If you received the uninstall file from an SEL representative, you may have saved it in a different location. Browse to that location before running this command.

Automatic RTAC Project Backup

When ACSELERATOR RTAC is updated from a previous version all RTAC projects from the previous version are automatically backed up and saved on the local disk in the following location. These projects can be imported into ACSELERATOR RTAC or zipped and have all projects brought back into the software with the restore option.

- C:\ProgramData\SEL\AcSELerator\RTAC\backup_exp

Software Maintenance

Software versions 1.28.xxx and later include a database maintenance option found under **SEL > Options**. Performing maintenance cleans up the database and improves software performance, especially after the software has imported, exported, converted, or deleted numerous projects. You can choose between standard or advanced maintenance; a detailed description is provided for each option, and SEL-5033 provides a recommendation of which type to use (if any).

SECTION 2

Communications

Overview

A core feature of the SEL Real-Time Automation Controller (RTAC) family of products is the ability to provide communications protocol conversion and data concentration among intelligent electronic devices (data sources) and SCADA or other upstream polling devices (data destinations). Use the following steps from within ACSELERATOR RTAC SEL-5033 Software to configure RTAC communications.

Step 1. Insert a device into your project.

Step 2. Specify the protocol you want to use in communicating with the device.

Step 3. Specify whether the connection is a client or a server.

Client and server are synonymous with master and slave in legacy protocols.

- A client connection is one in which the RTAC will be a client that polls the connected device for data.
- A server connection is one in which the RTAC serves data to SCADA or some other remote client polling for data.

Data Tags

Data within the RTAC are referenced by database points called tags that may be associated with client or server protocols, system values, global variables (including virtual tags), or values created in user-defined logic.

Tags are generally either global (visible to every RTAC process) or local (limited to a particular function and unavailable elsewhere in the RTAC software). Most tags, such as system tags, device tags, virtual tags, and tags defined in global variable lists (GVLs) in the IEC 61131-3 user logic are global. Other tags declared in user-defined functions are local to those functions. The RTAC supports a maximum of 25,000 tags, the SEL-3555 supports a maximum of 100,000 tags, and the SEL-3505 supports a maximum of 5000 tags. See *Section 9: Custom Logic* for more information on tag declaration.

Data flows within the RTAC from source tags to destination tags via Tag Processor mapping. Source data are typically from client device (IED) tags. Destination data tags typically are from server devices (HMIs, remote polling clients).

NOTE
Remember that RTAC updates tag values, including controls, once during each cycle-time interval.

1. Use the **Insert** ribbon to build source tags in the RTAC by inserting IED devices as client connections.

   Configure tags with the data type tabs for corresponding devices. These tags are global and are available for use anywhere in the RTAC. Reference these tags by using their corresponding assigned tag names.
2. Use the **Insert** ribbon to build destination tags in the RTAC by inserting devices as server connections.

   Configure tags with the data type tabs for corresponding devices. These tags are global and are available for use anywhere in the RTAC. Source tags typically provide the data values for destination tags.
3. Use the Tag Processor to map source tags to destination tags. Data from the source tag transfers to the destination tag at each RTAC processing cycle. See *Section 3: Tag Processor* for additional details.

If you need global tags that are of the same types as those contained in protocol tasks, insert a virtual tag list. Use GVLs to create standard data type global tags under the IEC 61131-3 ribbon (see *Section 9: Custom Logic*).

Data Structures

The RTAC data points each have a data structure similar to the IEC 61850 standard. This structure provides much more information for each point, including such other attributes as instantaneous value, time stamp, data quality, and dead-banded values. See *Appendix B: IEC 61131-3 Programming Reference* for an in-depth break down of each data type.

You can use data structure attributes for each tag in logic or map these attributes within the Tag Processor. To access each attribute within the structure, enter the tag name followed by a period (.) and the next structure level. ACSELERATOR RTAC concatenates the device name (Relay1) with the protocol (DNP) and the actual variable name (AI_0000). A period (.) after the tag name joins the tag name with the specific attribute (i.e., instMag). In this way, the point value is self-describing. Instead of typing only the point name (AI_0000) in the Tag Processor or other RTAC field to access the instantaneous value of AI_0000, type the fully qualified tag name (Relay1_DNP.AI_0000.instMag).

Example 2.1 Tag Data Structure Breakdown

This example illustrates the structure of an analog input tag name. We are assuming for this example that the RTAC is configured to poll one analog point from an IED named Relay1 using the DNP3 protocol. The default analog input tag name will be:

Relay1_DNP.AI_0000

The instantaneous value for that analog input will be:

Relay1_DNP.AI_0000.instMag

The following figure illustrates all the attributes in this tag's data structure.

AI_0000	MV
instMag	REAL
mag	REAL
range	RANGE_T
q	quality_t
validity	VALIDITY_T
detailQual	detailQual_t
source	SOURCE_T
test	BOOL
operatorBlocked	BOOL
t	timeStamp_t
value	dateTime_t
quality	timeQuality_t
daylight_savings_time	DST_t
UTC_Offset	INT
source	timeSource
db	REAL
zeroDb	REAL
rangeC	rangeConfigReal_t
hhLim	REAL
hLim	REAL
lLim	REAL
llLim	REAL
minVal	REAL
maxVal	REAL

Data structure description:

- **Relay1_DNP** is the device name. This is also the global name for this device. All tags associated with this device will begin with this name.
- **AI_0000** is the IEC 61131-3 variable name.
- **instMag** is the instantaneous value.
- **mag** is the value snapshot after instMag exceeds the dead-band value. It changes each time instMag exceeds the dead band (db). In event-driven protocols such as DNP3, this is the time-stamped dead-banded event value.
- **range** is the result of testing instMag against the rangeC values.
- **q** is a quality structure that contains data quality flags for this point.
- **t** is a time structure that contains the data time stamp as well as time quality information for this point.
- **db** is the dead-band value applied to the analog value instMag.
- **zeroDb** is a zero cutoff dead band. If mag is < zeroDb, mag will be set to 0.
- **rangeC** is a structure that contains user-configurable high and low limits. The range limits include high-high, high, low, low-low alarm limits. You can configure the RTAC to generate an alarm when the mag value exceeds the rangeC limit. The software logs this alarm, and it can then be viewed on the webpage. The type of alarm (high-high, etc.) is stored in the range field.

The following illustrates how to access some of the elements from the data point AI_0000.

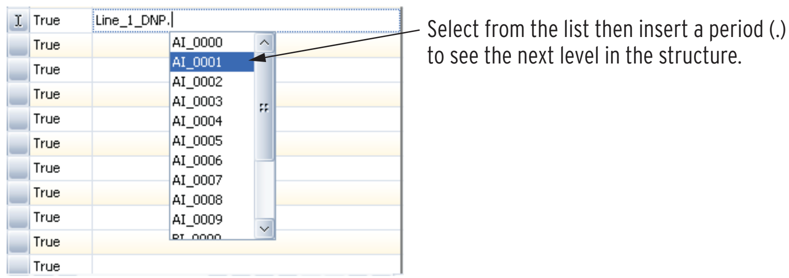
Dead-Banded Value of AI_0000	
Relay1_DNP.AI_0000.mag	Analog event value after the instantaneous value exceeds the dead band.
Time Stamp of AI_0000	
Relay1_DNP.AI_0000.t.value.dateTime	The date/time of the last dead band excursion for this point.
Data Quality of AI_0000	
Relay1_DNP.AI_0000.q.detailQual.outOfRange	A Boolean value that, if true, indicates that the value is presently out of range.

See *Data Types on page 747* for more detail on individual data type structures. Following is an example of various data values in a DNP IED called Relay_1.

Binary Input	
Relay1_BI_0000.stVal	Instantaneous value
Analog Input	
Relay1_AI_0000.instMag Relay1_AI_0000.mag	Instantaneous value Value when the dead band was exceeded
Counter	
Relay1_CNT_0000.actVal Relay1_CNT_0000.frVal	Instantaneous value Present frozen value
Analog Output	
Relay1_AO_0000.oper.setMag	Instantaneous value
DNP Binary Output	
Various DNP control methods for this point Relay1_BO_0000.operPulse.ctlVal Relay1_BO_0000.operLatchOn.ctlVal Relay1_BO_0000.operLatchOff.ctlVal Relay1_BO_0000.operTrip.ctlVal Relay1_BO_0000.operClose.ctlVal	

Autocomplete

The RTAC has an autocomplete tag entry mechanism. As you type a tag, this feature will automatically complete the tag name and the components of the tag structure. The RTAC software finds tag names that closely match what you type. Choose the name or structure element from the resulting list by selecting it with the mouse or arrow keys and pressing **<Enter>**. Next press the period **<.>** to see a list of the next data structure elements.



Create a Client Device Connection

A client device connection defines the protocol, port number, port settings, and a tag list of all data that the RTAC will poll from an IED.

The following steps illustrate, generally, how to configure a client connection to poll data from an IED. Sections that contain protocol-specific configuration information are described later in *Section 2: Communications*.

Step 1. Click the **Insert** ribbon or right-click **SEL_RTAC** in the device tree.

Step 2. Select the type of device (**SEL**, **Fieldbus I/O**, **Other**) from the ribbon menu.

Step 3. Select the device/protocol from the drop-down list within the device type.

For example, to poll a non-SEL IED that uses DNP master protocol, select **Other** for the device type, then **DNP** for the protocol.

Step 4. Select the type of client connection (serial, Ethernet, etc.).

Scroll through the list to see the graphic change with each connection type to show the relationship of communications between the RTAC and the device.

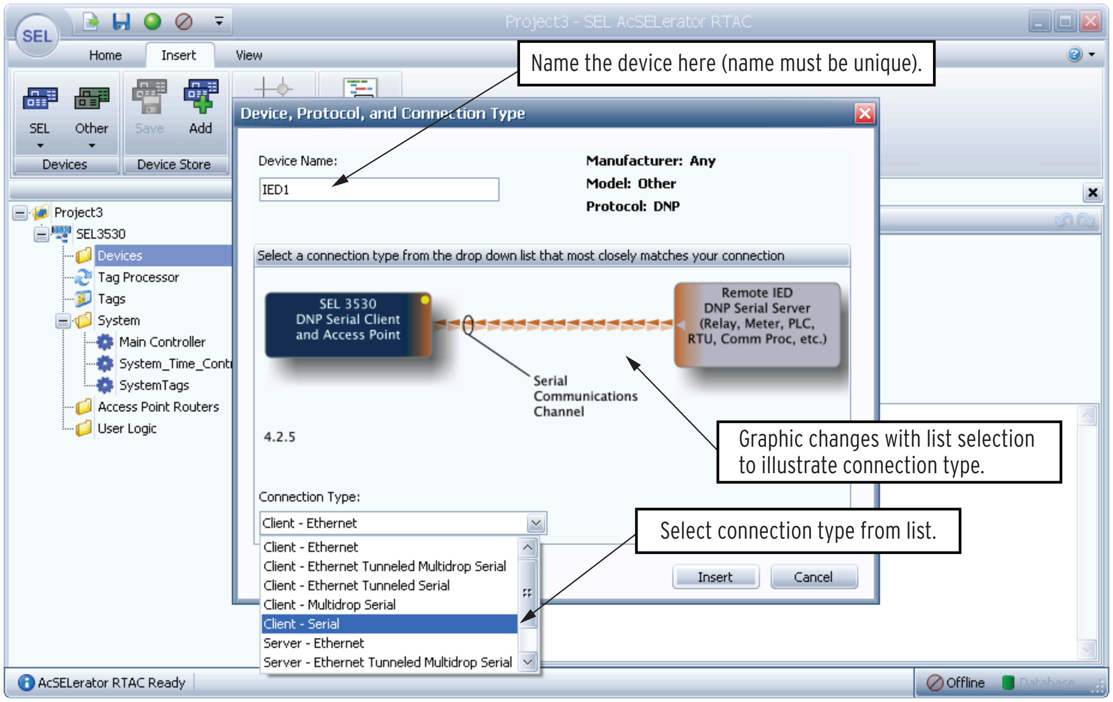


Figure 2.1 Select Serial or IP Client Connection

Step 5. Configure the communications parameters of the connection in the grid view. See *Figure 2.2* as an example.

This view displays automatically. Descriptions of each parameter are on the right of the screen. All configurable communications parameters are listed in this view and may include the following:

a. Port number.

The software assigns the next contiguous port number automatically, or you can choose to assign this number yourself.

b. Data rate, communication time-outs, etc.

c. Protocol-specific IDs, time-synchronization information, etc.

d. **Advanced Settings** check box.

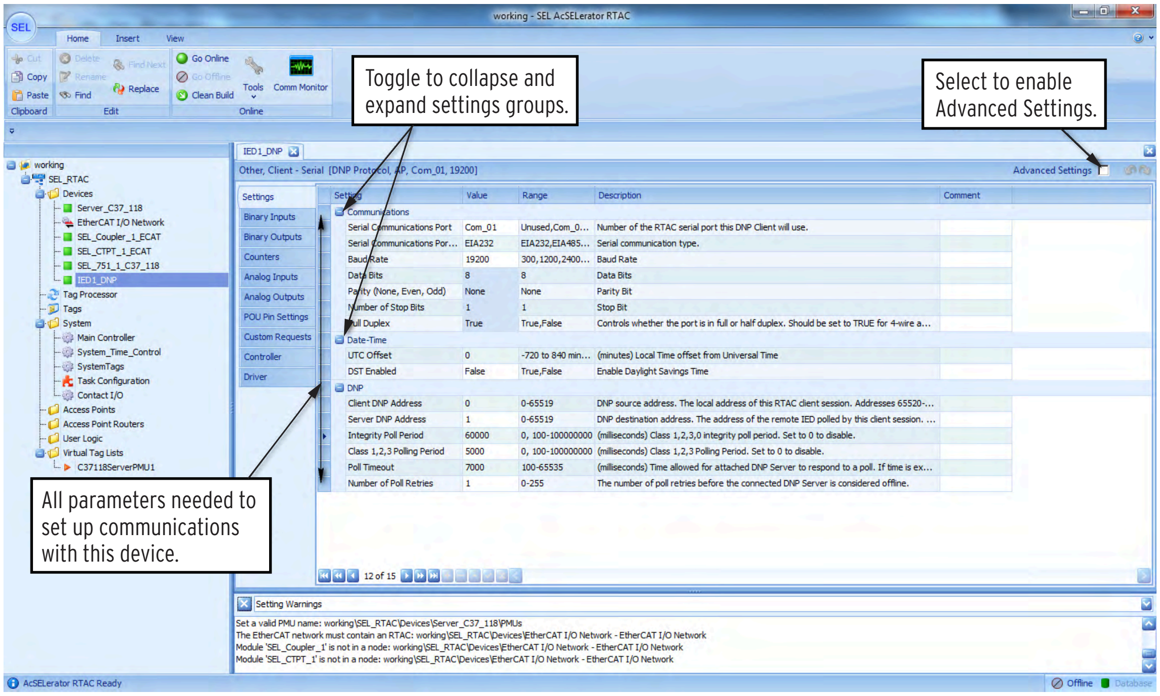


Figure 2.2 Example Serial Client Settings

Step 6. Unless data points are preconfigured, click on each device tag type tab to add the number of data tags (binary, analog, etc.) you want to use on this port.

Client Connection Types

Client—Serial: Use this setting to communicate in a point-to-point, radio, fiber, or other configuration where an EIA-232/EIA-422 connection is necessary. Insert one of these types of connections for each unique EIA-232/EIA-422 communications port on the RTAC that connects to an IED via EIA-232/EIA-422.

Client—Ethernet: Use this setting to communicate over Ethernet to an IED. Insert a new device connection for each device. The combination of the IEDs IP address and the IEDs port must be unique (i.e., two IEDs can have the same IP port number only if those two IEDs have different IP addresses).

Client—Multidrop Serial: Use this setting to enable polling of multiple IEDs on one serial port. Set the port to EIA-485 or EIA-232. If the port is configured as EIA-485, connect all IEDs on an EIA-485 loop to the RTAC on the specified serial port. Configure the port as EIA-232 if you are using multidrop communications equipment (EIA-232/EIA-485 transceiver, fiber modem, radio, etc.) that has an EIA-232 interface. Insert one multidrop direct serial client for each IED connected to that port. Configure each new device with a corresponding unique remote server address, if applicable. Configure each new device on this multidrop chain with the same serial communications port number. This configuration will cause the software to direct all traffic for the configured IEDs through a single configured serial port.

Client—Ethernet Tunneled Multidrop Serial: Use this setting to enable an Ethernet tunnel through a serial port server (SPS). Connect an IED to each port on a serial port server via EIA-232 serial cables. Connect the serial port server to the RTAC LAN. Insert one SPS multidrop Ethernet Tunneled Multidrop Serial connection for each IED connected to the SPS. Configure each of the SPS multidrop clients for the corresponding unique address of each IED and the unique IP port number configured on the SPS. Configure each SPS multidrop client for the same IP address of the SPS. Each poll will go first to the SPS and then to the appropriate port according to your configuration.

Client—Ethernet Tunneled Serial: Use this setting as in the previous configuration to enable Ethernet tunnel through an SPS, but for a point-to-point type connection.

The following steps detail adding client device data:

Step 1. Click the device tag type tab to add and configure points (tags).

Step 2. Click + to add tags (10,000 tag limit or 5000 for SEL-3505, per device tag type).

NOTE
Use Copy **<Ctrl+C>** and Paste **<Ctrl+V>** to populate the columns of Tag names and to duplicate devices.

Creating only the number of necessary tags will help the system run at optimum performance.

Step 3. Change the names of the tags, if necessary.

Step 4. Change other tag-related information as necessary.

Repeat these steps to configure all IED devices. When finished, configure one or more servers to deliver the data to remote client devices.

NOTE
Save the configuration by clicking the Save icon or **<Ctrl+S>**.

Create a Server Device Connection

A server device connection defines the protocol, port number, port settings, and a tag list of all data that a remote client device, communications processor, RTU, etc. will poll from the RTAC.

The following steps illustrate, generally, how to configure a server connection to send data to a polling device. Sections that contain protocol-specific configuration information are described later in *Section 2: Communications*.

Step 1. Select the type of device (**SEL**, **Other**) from the ribbon menu or right-click **SEL_RTAC** in the device tree.

Step 2. Select the protocol.

In the case of an SEL IED, select the device from the drop-down list within the device type ribbon and then select the protocol. For example, select **SEL > 3500 Series > RTAC > DNP**.

Step 3. Select the type of server connection (serial, Ethernet, etc.).

Step 4. Scroll through the list to see the graphic change with each connection type to show the relationship of communications between the RTAC and the device.

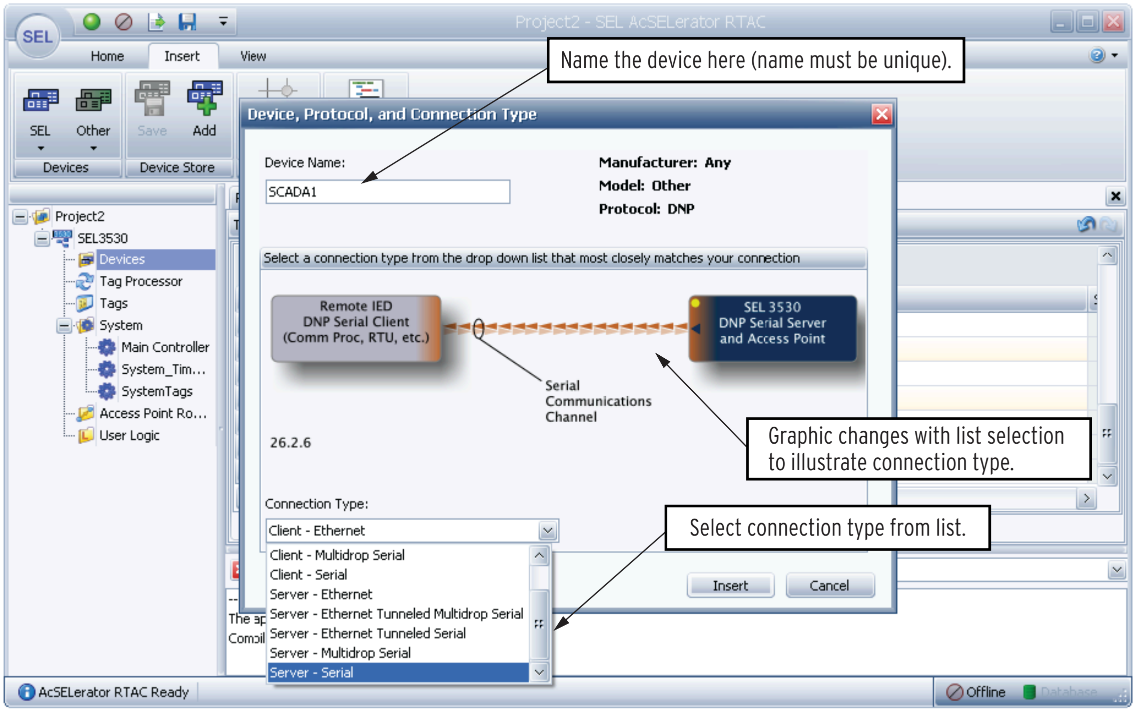


Figure 2.3 Select Serial or IP Server Connection

Step 5. Configure the communications parameters of the connection in the grid view. This view displays automatically.

Descriptions of each parameter are on the right of the screen. All configurable communications parameters are listed in this view and include the following:

a. Port number.

The software assigns the next contiguous port number automatically, or you can choose to assign this number yourself.

b. Data rate, communication time-outs, etc.

c. Protocol-specific IDs, time-synchronization information, etc.

d. **Advanced Settings** check box.

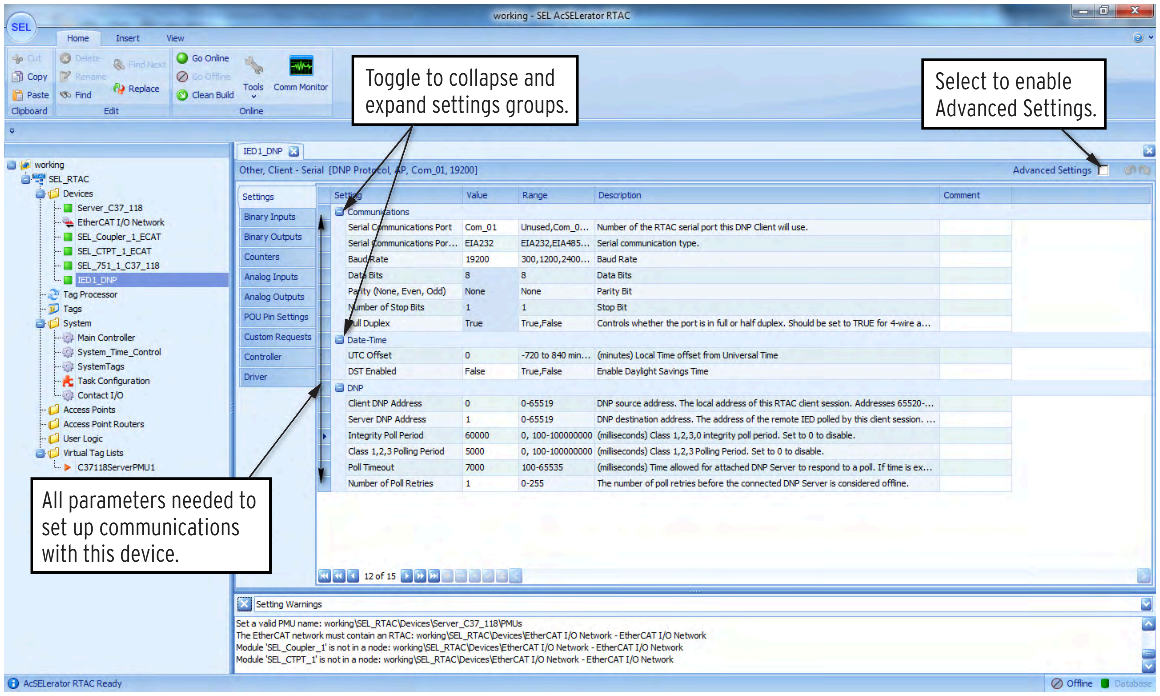


Figure 2.4 Example Serial Server Settings

Server Connection Types

Server—Serial: Use this setting to communicate in a point-to-point, radio, fiber, or other configuration where an EIA-232/EIA-422 connection is necessary to connect to a polling client (master). Insert one of these types of connections for each unique EIA-232/EIA-422 communications port on the RTAC that connects to a master via EIA-232/EIA-422.

Server—Ethernet: Use this setting to communicate over Ethernet to a client polling device. Insert a new server device connection for each remote client device. Configure each server connection with a unique server IP address and port if you set Allow Anonymous DNP IP Clients to False. The combination of the clients IP address and the servers local port must be unique (i.e., two server connections can have the same IP port number only if the two corresponding clients have different IP addresses).

Server—Multidrop Serial: Use this setting to configure the serial port as EIA-485 to allow the RTAC to reside on a multidrop EIA-485 polling loop. Each device on the loop must have a unique ID.

Server—Ethernet Tunneled Multidrop Serial: Use this setting to enable an Ethernet tunnel through a serial port server (SPS) to multiple polling clients. Connect one or more polling client to a serial port server via EIA-232 serial cables. Connect the serial port server to the RTAC LAN. Insert an Ethernet Tunneled Multidrop Serial server connection for each client connected to the SPS. Configure each of the SPS multidrop server connections for the appropriate address of each polling clients and the unique port number they are connected to on the SPS. Configure each SPS multidrop server for the same IP address of the SPS. Each poll reply will go first to the SPS and then to the appropriate port as configured.

Server—Ethernet Tunneled Serial: Use this setting to enable an Ethernet tunnel through a SPS as previously described, except that this is designed for a single-port SPS or point-to-point type connection.

The following steps detail adding server device data:

Step 1. Click the device tag type tab to add and configure points (tags).

Step 2. Click + to add tags (10,000 tag limit or 5,000 for SEL-3505, per device tag type).

Creating only the number of necessary tags will help the system run at optimum performance.

Step 3. Change the names of the tags, if necessary.

Step 4. Change other tag-related information as necessary.

Repeat these steps to configure all server connections. When finished, configure the Tag Processor to populate these server connection tags with actual values.

Using Copy/Paste Shortcuts

Copy a Column of Tag Names

If you have a list of tag names available in a column format (spreadsheet, IED documentation, etc.), you can save time by copying and pasting the entire column.

Step 1. From within ACSELERATOR RTAC, create the number of tags you want from each device tag type, as the client section, server section, and *Section 3: Tag Processor* describe.

Step 2. Highlight and copy the column of names from the source (Microsoft Excel spreadsheet, column from another RTAC device configuration, etc.).

Step 3. Click the first point from *Step 1* that you want to replace. You must highlight the entire tag name cell.

Step 4. Paste the new column onto the highlighted tag name cell. The pasted column of names will replace the default tag names one-for-one.

Step 5. See *Section 3: Tag Processor* for instructions on how you can use a SCADA tag name spreadsheet to also populate source and destination tag names.

Copy/Paste a Device

Use copy/paste to duplicate a configured device. An example would be a duplicate master station port used for testing that must be exactly the same as the main SCADA port.

Use this operation as well for a new device that is similar to or exactly the same as a configured device. In such a case, create a copy of the configured device and use this as a starting point for the new device. This operation will copy names and settings that you can then change in the new device.

To make a copy of a configured device:

Step 1. Right-click the device name.

Step 2. Select **Copy** or press **<Ctrl+C>**.

Step 3. Right-click the **Devices** label and select **Paste** or press **<Ctrl+V>**.

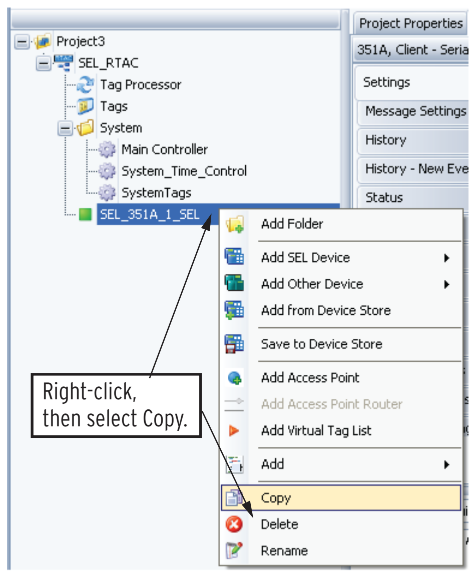
Step 4. Rename the new device to a unique name.

All tag names of the new device will reflect the new device name and the port number of the connection increments automatically for the new device. Custom logic written for that device will not be copied.

Example 2.2 SEL_351A_1 Device Copied to Make SEL_351A_1copy

The following figure illustrates making an exact copy of an SEL_351A device. Notice the duplicated device has the word copy appended to the name to avoid duplicate names.

To avoid confusion, rename SEL_351A_1copy to something unique, such as SEL_351A_2. Right-click, then select **Copy**.



Custom Device Definitions (Device Store)

Use Device Store to save a configured device definition onto your PC. For example, you can store a large IED that you have configured and use frequently and insert this IED later into new projects as necessary. All configuration, points, etc., will remain intact as originally configured except for custom logic.

To create a custom device definition:

Step 1. Configure a device according to the instructions previously in this section.

Step 2. Save your project with **<Ctrl+S>** or by pressing the  icon.

Step 3. Ensure there are no errors in the output window after the save operation.

Step 4. Click the icon under the **Device Store** ribbon or right-click the device configuration you want to save.

Step 5. In the pop-up dialog, provide a unique file name.

Step 6. Browse to the directory in which you want to store your custom devices.

Step 7. Press **Save**.

To use a custom device definition:

Step 1. With a project open, click the icon in the Device Store.

Step 2. In the pop-up dialog box, browse to the directory in which you stored your custom device definitions, select the definition you want to add.

Step 3. Click **Open**.

Step 4. The software will add the device, completely configured as before, into your project.

As with copy/paste, the software will copy names and settings, but you can change these later after creating the copy.

POU Pin Settings (Advanced Usage)

The **POU Pin Settings** tab provides the names, types, and default values of various components of each task for advanced usage. Each pin name has a Visible flag with a configurable value of True or False. Items marked as True will display on the controller logic block. Input pins control task operation, including polling functionality and data collection into the database. Output pins contain real-time task values used by the protocol. You can configure the default value for most input pins. You can use real-time pin values in logic you write in the RTAC.

NOTE
Use caution when making any programming modifications to the Controller Block or to the POU Pin settings. For more information on using the Controller CFC and POU Pin settings, please contact an SEL application engineer.

NOTE
Copying and pasting a device will not copy custom logic in the Controller Block or POU Pin settings.

Controller (Advanced Usage)

The **Controller** tab contains the Continuous Function Chart (CFC) function block for the task. POU Pin Settings that have their Visible tags set to True in the **POU Pin Settings** tab become visible in the Controller. The Controller block provides advanced users the ability to alter the standard functionality of the protocol task. The task function block contains input POU pins on the left side of the CFC block and output POU pins on the right side of the block. You can add CFC logic such as input blocks, output blocks, functions and function blocks to the controller work space as standalone logic or to interface to the controller task function block.

Tags (Overview)

The **Tags** tab for each device is a view of all the tags you have configured for this device connection.

While offline, use the **Tags** tab to view, select, copy, and paste large groups of data points from this server connection to the Tag Processor or spreadsheet. Right-click on the column heading to display options, including a filter, that you can use to restrict information display to only the data you want to see.

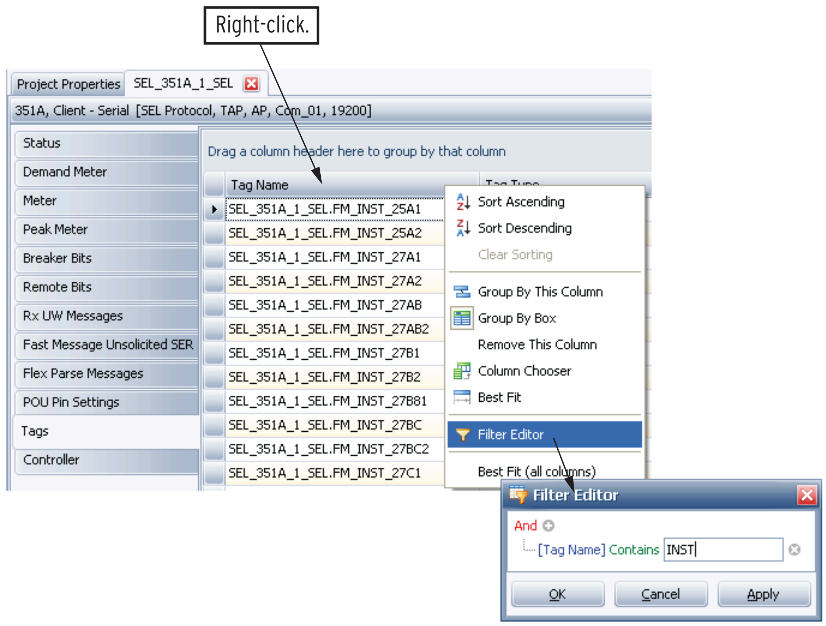


Figure 2.5 Using a Tag Filter

While online, use the **Tags** tab to see the present values of all tags you have defined for this device except for those tags for which you have assigned alias tag names. Alias tag names are part of a different global variable list and are located under the main **Tags** tab in the left-hand side of the ACSELERATOR RTAC menu tab. Expand each tag structure to view all embedded elements such as present value, time, quality, etc. See *Appendix B: IEC 61131-3 Programming Reference* for data type descriptions.

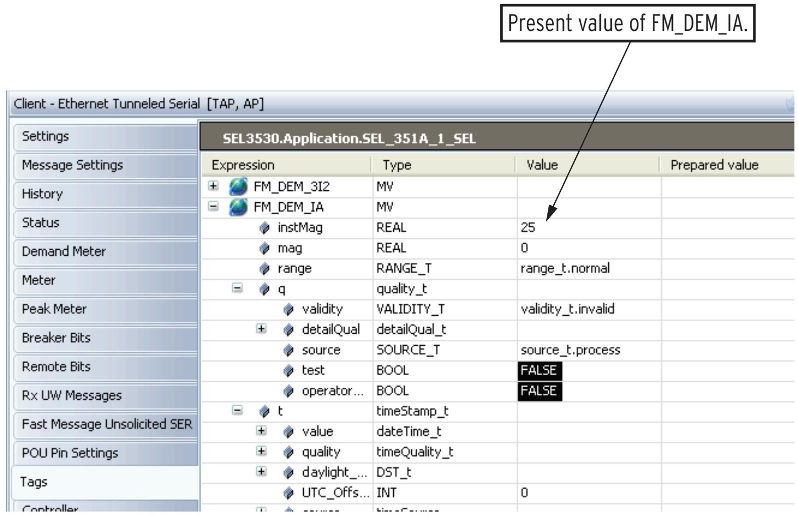


Figure 2.6 View Tags Online

Comm Monitor

Overview

You can use the communication monitor (Comm Monitor) to view data traffic in real time for all serial and Ethernet communications. The Comm Monitor displays communications events, such as data transmission and reception, and UART control signal transitions. The Comm Monitor shows the time of the event, the present state of the UART (when capturing serial communications) and data associated with the event. Data are displayed in raw hexadecimal and ASCII character format.

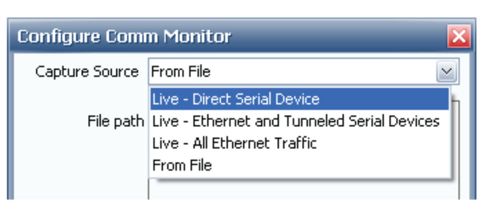
Configure Comm Monitor

From the **Insert** ribbon, add and configure a DNP, Modbus, SEL, or other device to communicate with the RTAC. Once you have completed the configurations, send the project to the RTAC and leave ACSELERATOR RTAC online with the RTAC. Perform the following steps to configure and use as many as 60 instances of the Comm Monitor:

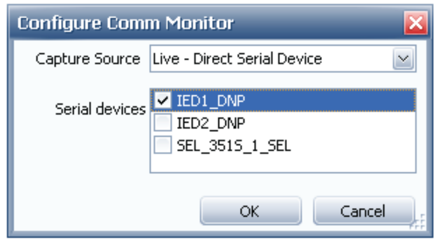
Step 1. Verify that you are online with the RTAC by observing the Online indicator at the bottom right corner of ACSELERATOR RTAC.

Step 2. From the **Home** ribbon menu, click **Comm Monitor**.

Step 3. From the drop-down menu under **Capture Source**, select the type of communications to monitor.

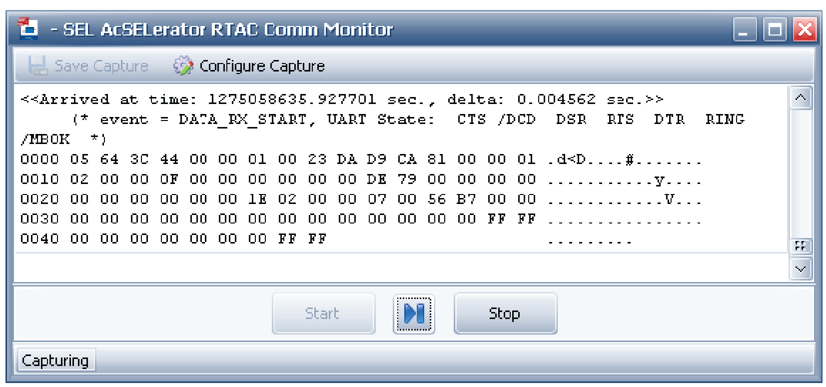


Step 4. ACSELERATOR RTAC will present all available devices of the communications type you selected. Check the box or boxes of the individual devices to monitor and click **OK**.



Step 5. The monitor screen will appear. Click the controls to start, pause, stop, save, and configure the capture.

- **Start**: Comm Monitor clears the display, connects to the configured device, and captures data.
- **Pause**: Stops display scrolling. Press **Resume** to continue displaying present communications.
- **Stop**: Stops data capture. After stopping, you can save the capture. Press **Start** again to clear the display and begin a new capture.
- **Save Capture**: After stopping the capture, you can save the present file in WireShark format.
- **Configure Capture**: Reconfigure the capture session to view data traffic on another device.



Step 6. Once you have captured all the needed data, click **Stop** and then **Save Capture**.

Step 7. Provide a file name and press **Save**.

You can load the saved file into Comm Monitor at any time while on- or offline by selecting **From File** as the Capture Source.

NOTE
You can also view the saved Comm Monitor files with Wireshark. Download WireShark software from WireShark.org

Comm Monitor Components

The Comm Monitor provides detailed information to help you not only see what the protocols are sending "on the wire" but also to provide timing information to help troubleshoot communications problems. For example, the RTAC sends a correct poll to an IED but is configured to time out too quickly because the IED reply is delayed. The IED does not respond reliably, and it appears that the RTAC is transmitting incorrect polls. The Comm Monitor delta time will show how much time has transpired between each communication event displayed and reveal that adding a slight delay in the RTAC configuration resolves the issue.

Figure 2.7 shows an example communication capture of SEL IED traffic. Note the ASCII messages displayed in hexadecimal and in human-readable character form. For example:

Hexadecimal characters:	46	49	44	3D	53	45	4C
ASCII character equivalent:	F	I	D	=	S	E	L

See *Table 2.1* for an explanation of Comm Monitor display field names. *Table 2.2* describes the data field portions of the Comm Monitor.

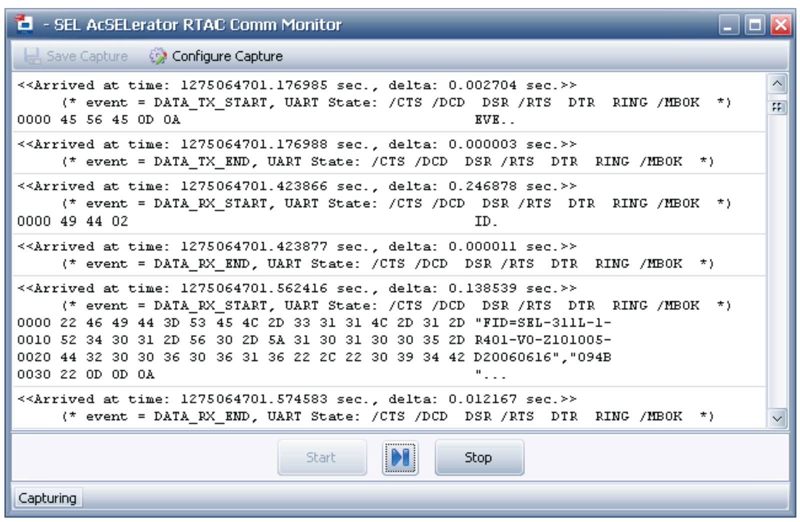


Figure 2.7 Comm Monitor Display Fields

Table 2.1 Comm Monitor Display Fields

Name	Meaning
Arrived at time	The time the communications event occurs in seconds. This does not indicate direction of data flow.
delta	The time difference in seconds between the present communications event and the one immediately preceding it.
event	A description of the type of event. For example, DATA_TX_START is when the UART begins transmitting data.
UART State	Shown for serial data capture, indicates the present state of the UART signals, e.g., CTS, DCD, DSR, etc. Deasserted state indicated by a / beside the state name. See *Table 2.3*.

Table 2.2 Comm Monitor Data Fields

Byte Count (Hexadecimal)	Hex Representation	ASCII Representation
0000 0010 0020 0030	22 46 49 44 3D 53 45 4C 2D 33 31 31 4C 2D 31 2D 52 34 30 31 2D 56 30 2D 5A 31 30 31 30 30 35 2D 44 32 30 30 36 30 36 31 36 22 2C 22 30 39 34 42 22 0D 0D 0A	"FID=SEL-311L-1- R401-V0-Z101005- D20060616","094B " ...

Table 2.3 UART Signals

Name	Meaning
CTS	Clear to send hardware flow control signal.
DCD	Data carrier detected.
DSR	Data set ready.
RTS	Request to send hardware flow control signal.
DTR	Data terminal ready.
RING	Valid only when using a modem. Indicates modem has sent RING to the RTAC.
MBOK	Valid only on MIRRORED BITS communications. Indicates MIRRORED BITS channel receive OK is asserted.

SCADA Protocol Redundancy

SCADA protocol redundancy is available on the DNP server and IEC 60870-5-101/104 server.

Protocol event synchronization between two separate RTAC units is available in RTAC firmware version R136 and later. If two separate RTAC units are configured in a failover scheme and are serving the same data to a single client, you can configure the two units to synchronize which events the client has already confirmed. When the client confirms the reception of an event from a server, that server transmits a UDP message to the other RTAC serving the event information. The second, or failover, RTAC then uses that information to identify which event the client on the first RTAC acknowledged and to acknowledge the same event in its own system.

In this configuration, if a single IED sends the same binary point change with time stamp to two separate RTACs and a client acknowledges the event on Server A, there will be no reporting of that same event to the client when the client begins communicating with Server B. This prevents transmission of duplicate events to the same client. This feature is ideal in a redundancy configuration where multiple RTACs collect the same data from IEDs. The client can then switch communications between the two RTACs without receiving duplicate events.

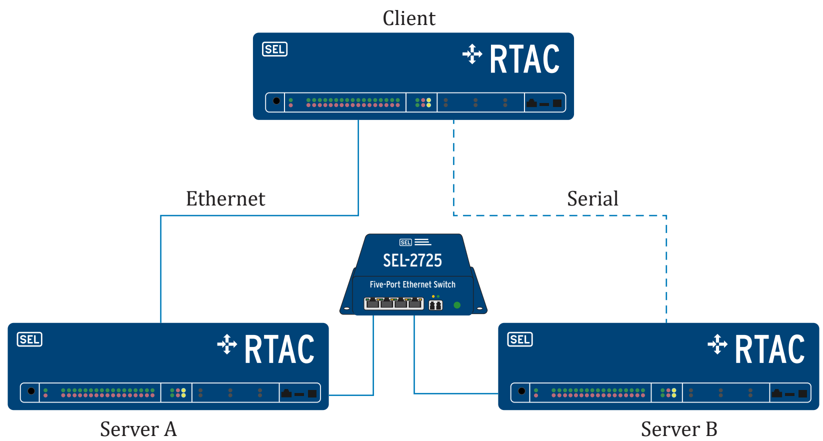


Figure 2.8 SCADA Synchronization

To synchronize two servers, use the Event Synch IP Address and Event Synch IP Port settings. An Ethernet connection between the two RTACs must be available. This can be the same physical Ethernet port that the client and server connection uses, or it can be any other Ethernet interface on the RTAC. The two synchronized RTACs can use any configurable connection method with the SCADA client, including a combination of serial and Ethernet connections.

DNP3

Overview

Configure DNP3 protocol on any of the RTAC serial or Ethernet ports to communicate with such other devices and systems as DNP3 IEDs, communications processors, RTUs, and remote client systems. The RTAC supports a combination of 200 servers and clients. Prior to R148, the SEL-3555 supports a combination of 256 servers and clients. For R148 and later firmware versions, the SEL-3555 supports as many as 500 DNP devices.

This section describes the configuration and use of DNP3 protocol with ACSELERATOR RTAC. The DNP3 Device Profile Document is included at the end of this section. For more DNP3 protocol information, go to *www.dnp.org*.

For a detailed description of client and server concepts and other general information on protocol configuration, see *Overview on page 25*.

DNP Client Configuration

Configure a DNP client connection to communicate via serial or Ethernet to IEDs. The RTAC will poll data from these IEDs and store the data it receives in global variables. Use the Tag Processor to map these data to any protocol, logs, user logic, etc.

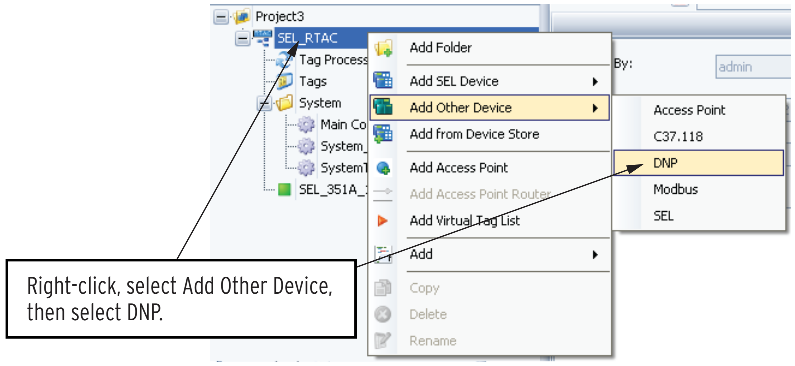


Figure 2.9 Insert DNP Client Device

Give the device connection a unique name and select the connection type, as shown in *Figure 2.1*. Refer to *Client Connection Types on page 31* for a description of each connection type. Note that, in a DNP client connection, the client DNP address setting is the address of the RTAC. The server DNP address setting is the DNP address of the IED being polled.

Settings Tab

The **Settings** tab contains all configurable items necessary for communications. Check the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description. Type any applicable comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Add Client Device Data

The following steps detail adding client device data.

Step 1. Click a device tag type tab to add and configure tags.

Step 2. Click + to add tags (10,000 tag limit or 5000 for SEL-3505, per device tag type).

NOTE
Use Copy (**<Ctrl+C>**) and Paste (**<Ctrl+V>**) to populate columns of Tag names and to duplicate devices.

Creating only the number of necessary tags will help the system run at optimum performance.

Step 3. Change the names of the tags, if necessary.

Step 4. Change other tag-related information as necessary.

Repeat these steps to configure all IED devices. When finished, configure one or more servers to deliver the data to remote client devices.

NOTE
Save the configuration by clicking the **Save** icon or by pressing **<Ctrl+S>**.

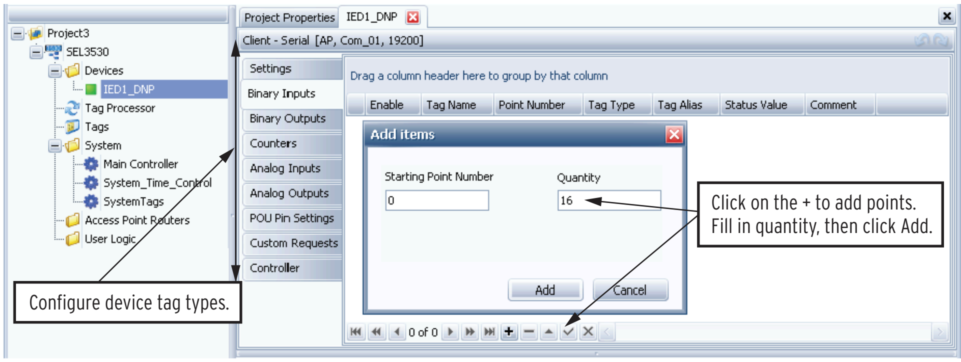


Figure 2.10 Add DNP Client Tags

Device Tag Type Configuration Parameters

Each device tag type has parameters that must be configured correctly to ensure proper system operation.

Table 2.4 Common Device Tag Type Parameters

Parameter	Description	Default
Enable	Set this flag to True to enable processing of this tag. Set this flag to False to disable processing.	True
Tag Name	This parameter describes the tag. You can change this name as necessary.	The default tag name contains the device name and tag type and is numbered 0–*n* tags.
Point Number	This is the point number for which the DNP client will poll. If points are not contiguous in the IED, change these point numbers to match the DNP database for the IED.	Contiguous from 0 to the point count.
Tag Alias	Enter an optional descriptive tag name in this field, and then reference this tag alias anywhere in the RTAC system in place of the actual tag name.	

Table 2.5 Binary Input Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	SPS
Status Value	The initialized value at startup.	False
Comment	Optional user-entered comment field.	

Table 2.6 Binary Output Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. In DNP, there are five binary output operations and one status.	Control operations is operSPC. Status is SPS.
Status Value	The initialized state at startup.	False
Control Value	The initialized control value at startup.	False
Control Model	Defines if the control is a single pulse or persistent (latch) type control.	Defaults are set according to control type and cannot be changed.
Number of Pulses	Defines the number of pulses the control will issue each time this point is controlled.	1

Parameter	Description	Default
On Pulse Dur	Defines the duration for an active pulse.	
Off Pulse Dur	Defines the duration for an inactive pulse. Although you can configure the pulse-off duration, the pulse-off command in DNP is not supported in the RTAC. A pulse-off command is treated as a pulse-on command.	
Comment	Optional user-entered comment field.	

Table 2.7 Counter Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	BCR
Actual Value	The initialized running value at startup.	0
Frozen Value	The default frozen value the tag will have before the RTAC initializes the tag upon startup.	
Comment	Optional user-entered comment field.	

Table 2.8 Analog Input Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	MV
Inst Magnitude	The initialized instantaneous value at startup.	
Magnitude	The initialized dead-banded value at startup.	
Dead Band	The number of units of change necessary to move the Inst Magnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag. An excursion of this dead band will generate a time-stamped DNP event.	
Zero Dead Band	The number of units at or below which Magnitude is forced to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	
Comment	Optional user-entered comment field.	

Table 2.9 Analog Outputs

Parameter	Description	Default
Set Magnitude	The commanded value of the analog output.	0

The other analog output settings are identical to those for analog inputs.

Custom Requests

Create as many as 15 optional custom requests per DNP client session. Add a request by clicking the + icon. Expand the custom request headings to configure the object, poll period, and broadcast request, and to add function codes and all other configurations related to the custom DNP request.

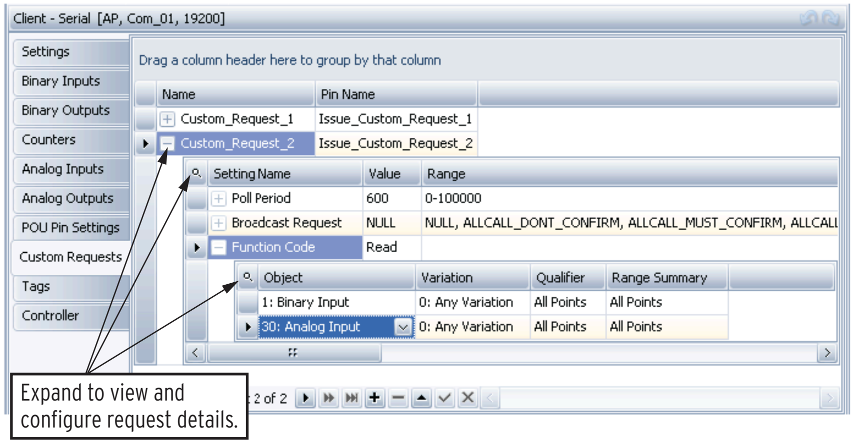


Figure 2.11 DNP Client Custom Requests

DNP Server Configuration

Configure a DNP server connection to communicate via serial or Ethernet to DNP polling clients (HMIs, communications processors, etc.). Create binary input, analog input, and counter tags in a DNP Server Shared Map as placeholders for data that the clients will poll. Configure the DNP server to use the configured shared map. Each DNP server can use a unique map, or can share the same map. Although two or more DNP servers can share a map, each server will maintain its own event queue. You can create a maximum of 100 shared DNP server maps in a project. Map data to those tags by using the Tag Processor from client IED or other source tags in the RTAC database.

Insert a unique DNP Server connection for each DNP client connected to the RTAC. Although sharing DNP server connections simultaneously with one or more polling clients is not permitted, you can configure various failover scenarios in the RTAC (see *DNP Server Failover on page 48*).

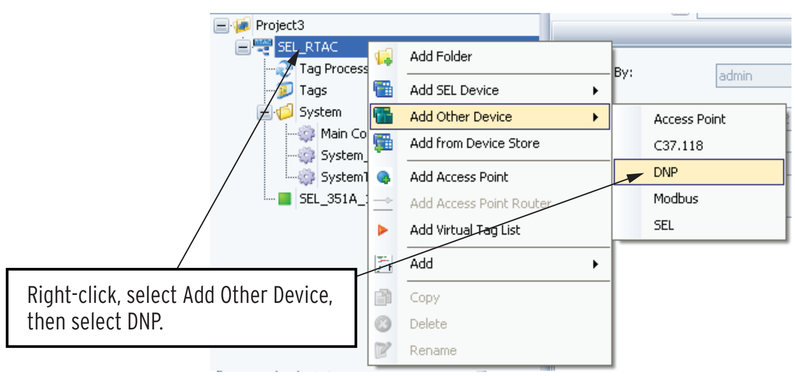


Figure 2.12 Insert DNP Server Device

Give the device connection a unique name and select the type of connection as shown in *Figure 2.3*. Note in DNP server, the Server DNP Address is the DNP address of the RTAC. The Client DNP Address is the address of the device polling the RTAC. You can configure multiple serial DNP servers to use the same serial port to create a virtual multidrop environment within the RTAC. Each server monitors traffic on the one configured serial port and responds to messages that contain the correct DNP address for that server. Each server maintains its own event queue.

The **Settings** tab contains all configurable items for communications. Note that you must enter a DNP Server Shared Map name in the settings. This map contains all of the tags that this DNP server uses. You insert and configure a DNP shared map under Tag Lists in the **Insert** tab. Check the Description column for details on each configuration item. Move the slider or hover your cursor over the description to see the entire description text. Type any necessary comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

DNP Server Failover

You can configure DNP servers to operate under the following failover scenarios:

DNP/IP: Multiple clients can poll one RTAC server if they are configured in a failover configuration so as to poll the RTAC one at a time. In this configuration, the DNP server maintains one DNP event queue. Events reported to one client are not reported in the next reply, even if the poll comes from a different client. If you set Anonymous DNP IP Clients = TRUE, the RTAC will allow any DNP client that can connect to the RTACs network to poll this DNP server, as long as the DNP IP addresses are correct. If you set Allow Anonymous DNP IP Clients = False, you can configure as many as 10 Client IP Addresses, separated by commas. In this configuration, the RTAC will reply to any client that has one of the configured IP addresses.

DNP Serial and Tunneled Serial: You can configure a DNP server to communicate DNP serial or tunneled serial over Ethernet with a redundant dial-up, leased line, etc., serial port with the following steps. Insert a DNP serial server, select **Advanced Settings**, then scroll down and configure the Failover Type to be another serial port or a tunneled connection. Configure the respective serial port or IP information for the failover port. You can configure as many as 10 TCP addresses, separated by commas. The DNP server monitors both the primary serial port and the failover Ethernet or serial port as redundant communication paths, and responds to polls from the respective interface. Because the DNP task is monitoring and responding to both interfaces, there are no failover delays or risks of missing queued events. The DNP server maintains one event queue regardless of how many failover ports are configured.

DNP Event Reporting

You can configure event data storage in the advanced settings in each DNP server. Analog, binary input, and counter data each have separate event queues that are configurable in size from 1–10,000 entries, for a maximum of 30,000 events per DNP server. By default, each event buffer is configured as a one-to-one correlation with the number of defined tags in the DNP server. As such, each buffer entry contains only one (the most recent) event with time stamp for each configured tag. Configure the binary, analog, and counter buffer sizes to match the number of tags defined for the respective types. Alternatively, you can configure each event buffer type as SOE and configure the size as many as 10000 entries. When a buffer type is configured as SOE, the RTAC places all events for a tag type in the respective buffer until the buffer size is exceeded. On a buffer overflow, the oldest events shall be maintained in the buffer and new events discarded. In SOE mode, the buffer may contain many events from a few tags or a few events from many tags.

Unsolicited event reporting can introduce a challenge for half-duplex communications in collision avoidance. If two devices on a multidrop line attempt to talk at the same time, a collision will occur. If they hold off the exact amount of time and then retry, they will experience another collision. You can configure the Transmit Minimum Delay and Transmit Maximum Delay in each RTAC to create a window of random transmit delays for unsolicited reporting. With these parameters configured, the two RTACs will retry at different random times, therefore avoiding the chances of another communication collision.

Add Server Device Data

The following steps detail adding server device data.

Step 1. Under the **Insert** tab, select **Tag Lists**, then DNP Server Shared Map.

This will insert a configurable tag list for a DNP server. You can configure multiple DNP servers to use this same shared tag list.

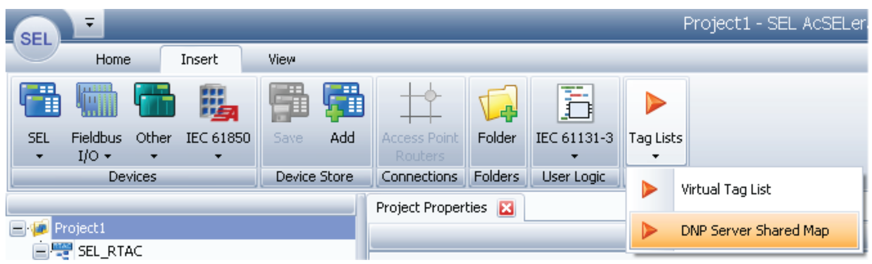


Figure 2.13 Add DNP Server Map

Step 2. Click a device tag type tab to add and configure tags.

Step 3. Click + to add tags (10,000 tag limit or 5000 for SEL-3505, per device tag type).

Creating only the number of necessary tags will help the system run at optimum performance.

Step 4. Change the names of the tags, if necessary.

Step 5. Change other tag-related information as necessary.

Repeat these steps to configure all server tags. When finished, configure the DNP server to use this custom map by selecting the MapName under settings. Although more than one DNP server can use the same map, each DNP server will manage separate event queues. Configure the Tag Processor to populate these server connection tags with actual values.

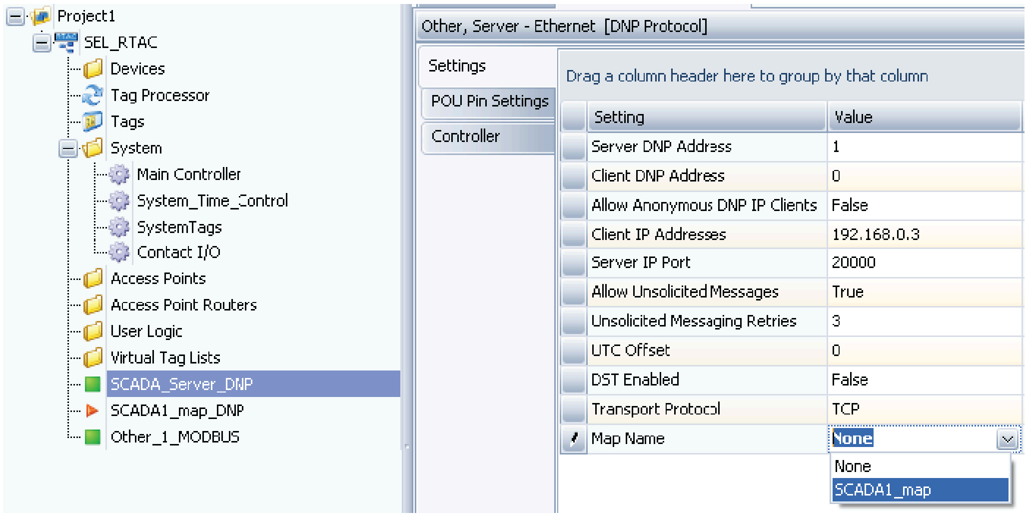


Figure 2.14 Select DNP Shared Map

Device Tag Type Configuration Parameters

Each device tag type has parameters that must be configured correctly to ensure proper system operation. Configuration columns in the DNP Shared Map have the same meaning as in DNP clients (described previously) except for the addition of fields that have special meaning in DNP servers. Use these fields to alter the default object variations returned to a polling DNP client.

Server Dial-Out

You can configure a DNP serial server device to dial out automatically to report unsolicited data. Insert a DNP serial server connection and select **Advanced Settings**. *Figure 2.15* shows these settings that are enabled when Modem Connected = TRUE. Notice the configuration supports an optional second phone number in case the RTAC does not get an answer when using the first phone number.

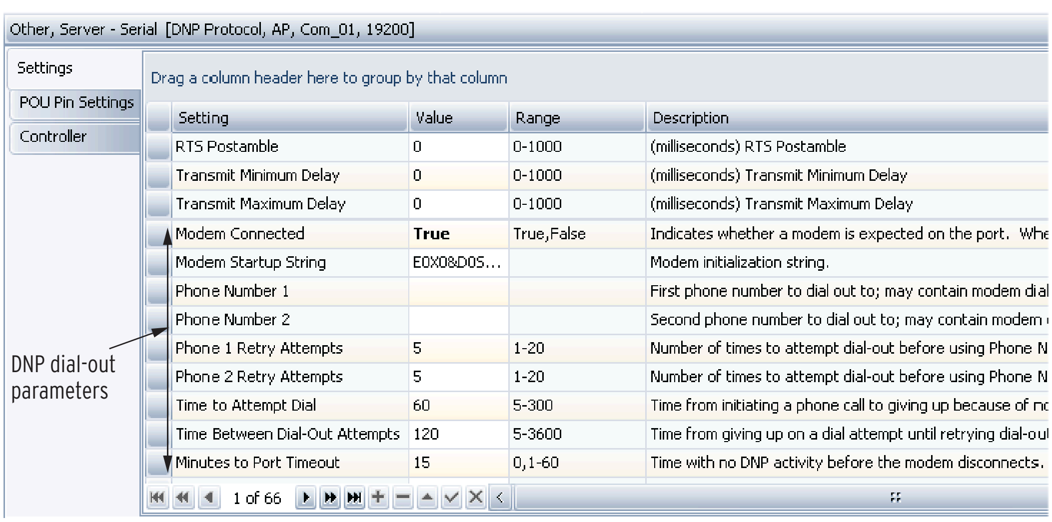


Figure 2.15 DNP Server Dial-Out

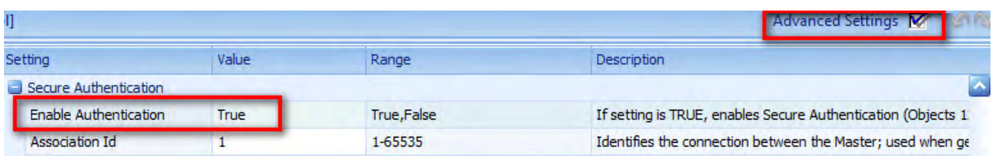
Secure Authentication

In firmware versions R143 and later, the DNP server supports Secure Authentication Version 5. Select the **Advanced Settings** check box in the DNP server to enable this feature. DNP Secure Authentication uses cryptographic hashing techniques for a challenge-and-response infrastructure that serves to authenticate critical commands and services. IEEE 1815 (DNP standard) defines commands and services that are critical and which must be challenged.

When Secure Authentication is enabled, the commands and services that can be challenged are configurable. This will cause the RTAC to issue a challenge any time the RTAC receives one of the configured function codes. By default, read commands do not prompt challenges. This can be configured to your desired configuration list. When Secure Authentication is enabled it will cause additional bandwidth consumption and delays in processing received messages that are challenged.

How to Configure Secure Authentication

Step 1. Under **Advanced Settings**, set **Enable Authentication** to True.



Step 2. Configure the appropriate options for the client connection under **Secure Authentication** in the **Settings** tab. Most values, by default, are configured for interoperability with most connections.

Step 3. When Secure Authentication is enabled, two new tabs appear on the DNP server: **Secure Event/Statistic Parameters** and **User Settings**. The **Secure Event/Statistic Parameters** tab allows for configuration of statistics about the secure authentication connection usage. The **User Settings** tab includes additional configuration parameters.

Step 4. In the **User Settings** tab, the user and update key information is configured. By default, there is a user defined with the role of **singleuser**. The DNP standard defines these roles and the capabilities each connection has with the DNP session. These roles are specific to DNP3 and have no effect on user accounts in the RTAC system or on other functionality outside of the DNP3 protocol. The singleuser default user contains the superset of permissions contained in the various roles. *Table 2.10* shows the permissions of each user role for DNP services.

Table 2.10 DNP Services User Role Permissions

Commands	User Roles							
	VIEWER	**OPERATOR**	**ENGINEER**	**INSTALLER**	**SECADM**	**SECAUD**	**RBACMNT**	**SINGLEUSER**
CONFIRM	X	X	X	X		X	X	X
READ	X	X	X	X		X	X	X
WRITE		X						X

Commands	User Roles							
	VIEWER	OPERATOR	ENGINEER	INSTALLER	SECADM	SECAUD	RBACMNT	SINGLEUSER
SELECT		X						X
OPERATE		X						X
DIRECT_OP		X						X
DIRECT_OP_ NOACK		X						X
FRZ		X						X
FRZ_NOACK		X						X
FRZ_CLEAR		X						X
FRZ_ CLEAR_ NOACK		X						X
COLD_ RESTART		X						X
WARM_ RESTART		X						X
ENABLE_ UNSOL		X	X					X
DISABLE_ UNSOL		X	X					X
ASSIGN_ CLASS			X					X
DELAY_ MEASURE		X	X					X
RECORD_ CURRENT_ TIME		X	X					X

Step 5. These user types only apply to the DNP session and do not allow for any use with other functionality of the RTAC. The columns Transfer data files, Change config, Change security config, change code, and local login do not have any functional capabilities on the RTAC. For this reason, SEL recommends using either the default USER or VIEWER role for read-only communications sessions and OPERATOR for reading and sending controls. The username must match the username that the client uses in the DNP communications session.

Step 6. Once a user is selected, configure the update key. The update key will be the symmetric key from the user on the DNP client. If there are multiple users, the update key can be the same or different between users. This is primarily dependent upon how the users on the clients are configured.

Step 7. The user public key is optional. The user public key is used to remotely change the update key through use of the asymmetric method for that user. If there is no user public key entered, the update key must be updated through the RTAC configuration software.

Changing the Update Key for DNP Secure Authentication

The DNP protocol offers a mechanism for remotely changing the update key. How often the update key needs to be changed depends on the security practices and policy of the implementers. There are two methods by which to change the update key: manually through the configuration software (The DNP standard does not specify a specific process for this) or by using DNP.

Through the Configuration Software

Once the new public key for the client has been obtained, open the RTAC settings and update the public key. This can be done locally at the unit or remotely if a connection to the RTAC is available.

The RTAC software will validate that the entered text contains a public key and conforms to valid privacy enhanced mail (PEM) format content.

Through DNP

When using DNP, there are two options for changing the update key: symmetric or asymmetric.

- **Symmetric**: In this manner, the RTAC configuration will need to have the Authority Certification Key, which is configured in the **DNP Settings** tab under the enable DNP Secure Authentication option. The RTAC will verify that the Authority Certification Key matches what the client sends when an update key change is attempted through the symmetric process.
- **Asymmetric**: In this manner, the RTAC configuration will need to have the User Public Key that the client uses, and the client will need to have RTAC active x.509 certificate public key. This allows the client and the RTAC to authenticate and change the update key.

Remote User Management

The DNP standard allows for the protocol to dynamically manage users for secure authentication. An authority can be set up that will authenticate that the client is capable of adding, removing, or changing a user. The client and the authority cannot have the same credentials. In order for the RTAC to accept user management changes via DNP, the RTAC must have the public key of the authority. This public key is added to the CA certificates in the RTAC web interface.

POU Pin Settings

See *POU Pin Settings (Advanced Usage) on page 37* for a description of the **POU Pin Settings** tab.

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

DNP3 Device Profile Document

The following DNP3 Device Profile displays only selections relevant to the RTAC DNP3 implementation.

Table 2.11 Server (Outstation)

Parameter	Value
Vendor name	Schweitzer Engineering Laboratories
Device name	RTAC
Highest DNP request level	Level 3
Highest DNP response level	Level 3
Device function	Outstation
Notable objects, functions, and/or qualifiers supported	Analog Dead-Band Objects (object 34)
Maximum data link frame size transmitted/received (octets)	292
Maximum data link retries	Configurable, range 0–15
Requires data link layer confirmation	Configurable by setting
Maximum application fragment size transmitted/received (octets)	2048
Maximum application layer retries	None
Requires application layer confirmation	When reporting event data
Data link confirm time-out	Configurable
Complete application fragment time-out	None
Application confirm time-out	Configurable
Complete application response time-out	None
Executes control WRITE binary outputs A	Always
Executes control SELECT/OPERATE	Always
Executes control DIRECT OPERATE	Always
Executes control DIRECT OPERATE-NO ACK	Always
Executes control count greater than 1	When pulse count > 1
Executes control Pulse On	Always
Executes control Pulse Off	Never
Executes control Latch Off	Always
Executes control Latch On	Always

Parameter	Value
Executes control Queue	Never
Executes control Clear Queue	Never
Reports binary input change events when no specific variation requested	Only timetagged
Reports time-tagged binary input change events when no specific variation requested	Binary input change with time
Sends unsolicited responses	Configurable with unsolicited message enable settings. Increases retry time (configurable) when a maximum retry setting is exceeded.
Sends static data in unsolicited responses	Never
Default counter object/variation	Object 20, Variation 6
Counter rollover	32 bits
Sends multifragment responses	Yes

Table 2.12 DNP Server (Slave) Object

Obj	Var (*default)	Description	REQUEST Func Codes (dec)	REQUEST QualCodes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
1	0	Binary Input—All Variations	1,22	0,1,6,7,8,17,28		
1	1	Binary Input	1	0,1,6,7,8,17,28	129	0,1,17,28
1	2*	Binary Input With Status	1	0,1,6,7,8,17,28	129	0,1,17,28
2	0	Binary Input Change —All Variations	1	6,7,8		
2	1	Binary Input Change Without Time	1	6,7,8	129,130	17,28
2	2*	Binary Input Change With Time	1	6,7,8	129,130	17,28
2	3	Binary Input Change With Relative Time	1	6,7,8	129,130	17,28
10	0	Binary Output—All Variations	1	0,1,6,7,8		
10	2*	Binary Output Status	1	0,1,6,7,8	129	0,1
12	0	Control Block—All Variations				
12	1	Control Device Output Block	3,4,5,6	17,28	129	echo of request
12	2	Pattern Control Block	5,6	7		
12	3	Pattern Mask	5,6	0,1		
20	0	Binary Counter—All Variations	1,7,8,9,10,22	0,1,6,7,8,17,28		
20	1	32-Bit Binary Counter	1,7,8,9,10	0,1,6,7,8,17,28	129	0,1,17,28
20	2	16-Bit Binary Counter	1,7,8,9,10	0,1,6,7,8,17,28	129	0,1,17,28
20	5	32-Bit Binary Counter Without Flag	1,7,8,9,10	0,1,6,7,8,17,28	129	0,1,17,28
20	6*	16-Bit Binary Counter Without Flag	1,7,8,9,10	0,1,6,7,8,17,28	129	0,1,17,28
21	0	Frozen Counter—All Variations	1,22	0,1,6,7,8,17,28		

Obj	Var (*default)	Description	REQUEST Func Codes (dec)	REQUEST QualCodes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
21	1	32-Bit Frozen Counter	1	0,1,6,7,8,17,28	129	0,1,17,28
21	2	16-Bit Frozen Counter	1	0,1,6,7,8,17,28	129	0,1,17,28
21	5	32-Bit Frozen Counter With Time of Freeze	1	0,1,6,7,8,17,28	129	0,1,17,28
21	6*	16-Bit Frozen Counter With Time of Freeze	1	0,1,6,7,8,17,28	129	0,1,17,28
21	9	32-Bit Frozen Counter Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
21	10	16-Bit Frozen Counter Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
22	0	Counter Change Event —All Variations	1	6,7,8		
22	1	32-Bit Counter Change Event Without Time	1	6,7,8	129,130	17,28
22	2*	16-Bit Counter Change Event Without Time	1	6,7,8	129,130	17,28
22	5	32-Bit Counter Change Event With Time	1	6,7,8	129,130	17,28
22	6	16-Bit Counter Change Event With Time	1	6,7,8	129,130	17,28
23	0	Frozen Counter Event —All Variations	1	6,7,8		
23	1	32-Bit Frozen Counter Event Without Time	1	6,7,8	129,130	17,28
23	2	16-Bit Frozen Counter Event Without Time	1	6,7,8	129,130	17,28
23	5	32-Bit Frozen Counter Event With Time	1	6,7,8	129,130	17,28
23	6*	16-Bit Frozen Counter Event With Time	1	6,7,8	129,130	17,28
30	0	Analog Input—All Variations	1,22	0,1,6,7,8,17,18		
30	1	32-Bit Analog Input	1	0,1,6,7,8,17,28	129	0,1,17,28
30	2	16-Bit Analog Input	1	0,1,6,7,8,17,28	129	0,1,17,28
30	3	32-Bit Analog Input Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
30	4*	16-Bit Analog Input Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
30	5	Short Floating Point Analog Input (32 bit)	1	0,1,6,7,8,17,28	129	0,1,17,28
32	0	Analog Change Event —All Variations	1	6,7,8		
32	1	32-Bit Analog Change Event Without Time	1	6,7,8	129,130	17,28

Obj	Var (*default)	Description	REQUEST Func Codes (dec)	REQUEST QualCodes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
32	2	16-Bit Analog Change Event Without Time	1	6,7,8	129,130	17,28
32	3	32-Bit Analog Change Event With Time	1	6,7,8	129,130	17,28
32	4*	16-Bit Analog Change Event With Time	1	6,7,8	129,130	17,28
32	5	Short Floating Point Analog Change Event	1	6,7,8	129,130	17,28
32	7	Short Floating Point Analog Change Event With Time	1	6,7,8	129,130	17,28
34	0	Analog Dead Band—All Variations	1	0,1,6,7,8,17,28		
34	1*	16-Bit Analog Dead Band	1,2	0,1,6,7,8,17,28	129	0,1,17,28
34	2	32-Bit Analog Dead Band	1,2	0,1,6,7,8,17,28	129	0,1,17,28
34	3	Short Floating Point Dead Band	1,2	0,1,6,7,8,17,28	129	0,1,17,28
40	0	Analog Output Status —All Variations	1	0,1,6,7,8		
40	1	32-Bit Analog Output Status	1	0,1,6,7,8	129	0,1,17,28
40	2*	16-Bit Analog Output Status	1	0,1,6,7,8	129	0,1,17,28
40	3	Short Floating Point Analog Output Status (32 bit)	1	0,1,6,7,8	129	0,1,17,28
41	0	Analog Output Block —All Variations				
41	1	32-Bit Analog Output Block	3,4,5,6	17,28	129	echo of request
41	2	16-Bit Analog Output Block	3,4,5,6	17,28	129	echo of request
41	3	Short Floating Point Analog Output Block (32 bit)	3,4,5,6	17,28	129	echo of request
50	0	Time and Date—All Variations				
50	1*	Time and Date	1,2	7,8 (index=0)	129	07 (quantity=1)
50	3	Time and Date (Last Recorded Time)	2	7 (quantity=1)	129	
51	1	Time and Date CTO			129	07 (quantity=1)
51	2*	Unsynchronized Time and Date CTO			129	07 (quantity=1)
52	1	Time Delay Coarse			129	07 (quantity=1)
52	2	Time Delay Fine			129	07 (quantity=1)
60	1	Class 0 Data	1,22	6,7,8		
60	2	Class 1 Data	1,20,21,22	6,7,8		
60	3	Class 2 Data	1,20,21,22	6,7,8		
60	4	Class 3 Data	1,20,21,22	6,7,8		

Obj	Var (*default)	Description	REQUEST Func Codes (dec)	REQUEST QualCodes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
80	1	Internal Indications	1 2	0,1 1 (index 4,7)		
NA	NA	No Object	13,14,23,24			

Table 2.13 DNP Client (Master) Object

Obj	Var	Description	REQUEST Func Codes (dec)	REQUEST Qual Codes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
1	0	Binary Input—Any Variation	1	0,1,6,7,8,17,28		
1	1	Binary Input—Packed Format	1	0,1,6,7,8,17,28	129	0,1,17,28
1	2	Binary Input—With Flags	1	0,1,6,7,8,17,28	129	0,1,17,28
2	0	Binary Input Event —Any Variation	1	6,7,8		
2	1	Binary Input Event—Without Time	1	6,7,8	129,130	17,28
2	2	Binary Input Event— With Absolute Time	1	6,7,8	129,130	17,28
2	3	Binary Input Event —With Relative Time	1	6,7,8	129,130	17,28
10	0	Binary Output—Any Variation	1	0,1,6,7,8,17,28		
10	2	Binary Output— Output Status With Flags	1	0,1,6,7,8,17,28	129	0,1,17,28
11	1	Binary Output Event —Status Without Time			129,130	17,28
11	2	Binary Output Event —Status With Time			129,130	17,28
12	1	Binary Command—Control Relay Output Block (CROB)	3,4,5,6	17,28	129	echo of request
20	0	Counter—Any Variation	1,7,8,9,10	0,1,6,7,8,17,28		
20	1	Counter—32-bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
20	2	Counter—16-bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
20	5	Counter—32-bit Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
20	6	Counter—16-bit Without Flag	1	0,1,6,7,8,17,28	129	0,1 17,28
21	0	Frozen Counter—Any Variation	1	0,1,6,7,8,17,28		
21	1	Frozen Counter—32-bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
21	2	Frozen Counter—16-bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
21	5	Frozen Counter—32-bit With Flag and Time of Freeze	1	0,1,6,7,8,17,28	129	0,1,17,28
21	6	Frozen Counter—16-bit With Flag and Time of Freeze	1	0,1,6,7,8,17,28	129	0,1,17,28
21	9	Frozen Counter— 32-bit Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28

Obj	Var	Description	REQUEST Func Codes (dec)	REQUEST Qual Codes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
21	10	Frozen Counter—16-Bit Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
22	0	Counter Event—Any Variation	1	6,7,8		
22	1	Counter Event—32-Bit With Flag	1	6,7,8	129,130	17,28
22	2	Counter Event—16-Bit With Flag	1	6,7,8	129,130	17,28
22	5	Counter Event—32-Bit With Flag and Time	1	6,7,8	129,130	17,28
22	6	Counter Event—16-Bit With Flag and Time	1	6,7,8	129,130	17,28
23	0	Frozen Counter Event—Any Variation	1	6,7,8		
23	1	Frozen Counter Event—32-Bit With Flag	1	6,7,8	129,130	17,28
23	2	Frozen Counter Event—16-Bit Without Flag	1	6,7,8	129,130	17,28
23	5	Frozen Counter Event—32-Bit With Flag and Time	1	6,7,8	129,130	17,28
23	6	Frozen Counter Event—16-Bit With Flag and Time	1	6,7,8	129,130	17,28
30	0	Analog Input—Any Variation	1	0,1,6,7,8,17,28		
30	1	Analog Input—32-Bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
30	2	Analog Input—16-Bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
30	3	Analog Input—32-Bit Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
30	4	Analog Input—16-Bit Without Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
30	5	Analog Input—Single-prec flt-pt With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
32	0	Analog Input Event—Any Variation	1	6,7,8		
32	1	Analog Input Event—32-Bit Without Time	1	6,7,8	129,130	17,28
32	2	Analog Input Event—16-Bit Without Time	1	6,7,8	129,130	17,28
32	3	Analog Input Event—32-Bit With Time	1	6,7,8	129,130	17,28
32	4	Analog Input Event—16-Bit With Time	1	6,7,8	129,130	17,28
32	5	Analog Input Event—Single-prec flt-pt Without Time	1	6,7,8	129,130	17,28
32	7	Analog Input Event—Single-prec flt-pt With Time	1	6,7,8	129,130	17,28

Obj	Var	Description	REQUEST Func Codes (dec)	REQUEST Qual Codes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
40	0	Analog Output Status —Any Variation	1	0,1,6,7,8,17,28		
40	1	Analog Output Status —32-Bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
40	2	Analog Output Status —16-Bit With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
40	3	Analog Output Status—Single-prec flt-pt With Flag	1	0,1,6,7,8,17,28	129	0,1,17,28
41	1	Analog Output—32-Bit	3,4,5,6	17,28	129	echo of request
41	2	Analog Output—16-Bit	3,4,5,6	17,28	129	echo of request
41	3	Analog Output—Single-prec flt-pt	3,4,5,6	17,28	129	echo of request
42	1	Analog Output Event —32-bit Without Time			129,130	17,28
42	2	Analog Output Event —16-bit Without Time			129,130	17,28
42	3	Analog Output Event —32-bit With Time			129,130	17,28
42	4	Analog Output Event —16-bit With Time			129,130	17,28
42	5	Analog Output Event—Single-prec flt-pt Without Time			129,130	17,28
42	7	Analog Output Event—Single-prec flt-pt With Time			129,130	17,28
50	1	Time and Date—Absolute Time	1,2	7 (Qty = 1)	129	7 (Qty = 1)
50	3	Time and Date—Absolute Time at Last Recorded Time	2	7 (Qty = 1)	129	7 (Qty = 1)
51	1	Time and Date CTO—Absolute Time, synchronized			129,130	7 (Qty = 1)
51	2	Time and Date CTO—Absolute Time, unsynchronized			129,130	7 (Qty = 1)
52	1	Time Delay—Coarse			129	7 (Qty = 1)
52	2	Time Delay—Fine			129	7 (Qty = 1)
60	1	Class Objects—Class 0 Data	1	6		
60	2	Class Objects—Class 1 Data	1 20,21	6,7,8 6		
60	3	Class Objects—Class 2 Data	1 20,21	6,7,8 6		
60	4	Class Objects—Class 3 Data	1 20,21	6,7,8 6		

Obj	Var	Description	REQUEST Func Codes (dec)	REQUEST Qual Codes (hex)	RESPONSE Func Codes (dec)	RESPONSE Qual Codes (hex)
80	1	Internal Indications —Packed Format	1 2	0,1 1 (index 4,7)	129	0,1
NA	NA	No Object (function code only)	13,23 14,24			

Although the RTAC is classified as a DNP3 Level 3 client, it will support the items indicated in the previous tables in automated or custom requests. It is the user's responsibility to create custom requests that the connected DNP3 server can parse.

Modbus

This section describes the configuration and usage of the Modbus protocol with ACSELERATOR RTAC. The RTAC supports Modbus RTU, Modbus/TCP, and Modbus serial tunneled over Ethernet. Modbus ASCII is not supported at this time. Configure the Modbus protocol on any of the RTAC serial or Ethernet ports to communicate with Modbus IEDs, communications processors, RTUs, HMI systems, etc. For a detailed description of client and server concepts and other general information on protocol configuration, see *Overview on page 43*.

Supported Modbus Function Codes

- Read Coil Status, function code 01h
- Read Discrete Inputs, function code 02h
- Read Holding Registers, function code 03h
- Read Input Registers, function code 04h
- Write Single Coil, function code 05h
- Write Single Holding Register, function code 06h
- Write Multiple Coils, function code 0Fh
- Write Multiple Holding Registers, function code 10h
- Report Client/Server ID, function code 11

Important: Register offsets often used with Modbus devices such as 30000 or 40000 are for reference to that device only and are not actually used in the Modbus protocol. As such, these references are not used in the RTAC Modbus protocol setup.

NOTE
Register addressing in RTAC Modbus is 0 based. For example, the RTAC Modbus message register address will be 0 if the register address in the ACSELERATOR RTAC project is configured as 0.

Modbus Client Configuration

Configure as many as 120 (in the SEL-3530, SEL-3530-4, SEL-2241, and SEL-3532) or 256 (in the SEL-3555 and SEL-3560) concurrent Modbus client or server connections to communicate via serial or Ethernet to IEDs. The RTAC will poll data from these IEDs and store the data in its database. Use the Tag Processor to map these data to server connections, logs, user logic, etc.

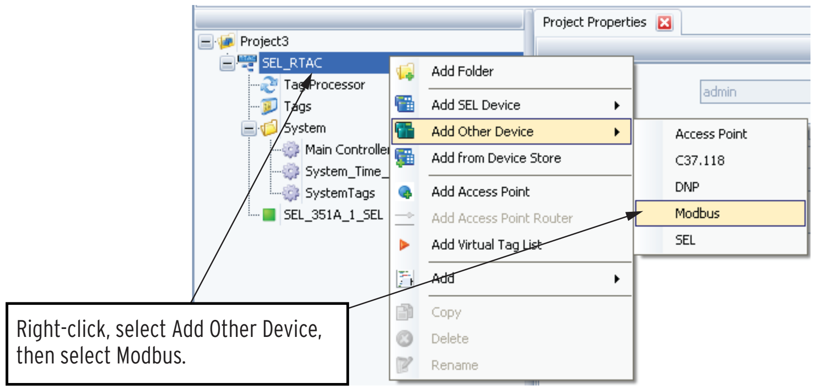


Figure 2.16 Insert Modbus Client

Give the device connection a unique name, and select the type of connection as shown in *Figure 2.1*.

Settings Tab

Upon your selecting the **Settings** tab, the RTAC displays all configurable items for communication. Check the **Description** column for details on each configuration item. Move the slider bar, or hover your mouse over a truncated description to see the entire text of a description. Type any necessary comments in the blank column to the far right.

Select the **Advanced Settings** check box to enable configuration of advanced settings.

Select device data settings tabs such as **Coils**, **Discrete Inputs**, etc., to add and configure Modbus tags.

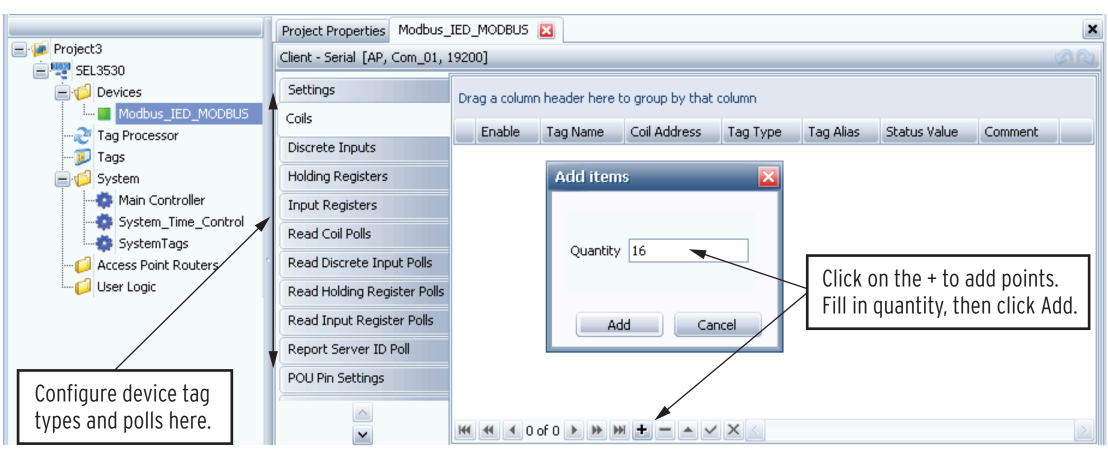


Figure 2.17 Add Modbus Client Tags

Add Client Device Data

The following steps detail adding client device data.

Step 1. Click on a device tag type tab to add and configure tags.

Step 2. Click + to add tags (10,000 tag limit or 5000 for SEL-3505, per device tag type).

NOTE
Use Copy (**<Ctrl+C>**) and Paste (**<Ctrl+V>**) to populate columns of Tag names and to duplicate devices.

Creating only the number of necessary tags will help the system run at optimum performance.

Step 3. Change the names of the tags, if necessary.

Step 4. Change other tag-related information as necessary.

Repeat these steps to configure all IED devices. When finished, configure one or more servers to deliver the data to remote client devices.

Device Tag Type Configuration Parameters

Data and poll types have parameters that you must configure correctly to ensure proper system operation. ACSELERATOR RTAC will gray out configuration fields that do not apply to a specific device tag type. The common input parameters are configurable in all device tag types. The following tables list specific device tag type configuration parameters.

Table 2.14 Common Input Parameters

Parameter	Description	Default
Enable	Tag processing enable. This flag must be True to enable processing of this tag. Setting this flag to False disables processing.	True
Tag Name	Configurable name that contains the device name and point description by default. You can change this name as necessary.	
Tag Type	The data type of this tag. See *Data Types on page 747* for more details.	
Tag Alias	An optional descriptive tag name. If you use a Tag Alias, reference this alias, instead of the actual tag name, in the RTAC.	
Status Value	The default value for the tag prior to initialization of the tag at startup.	False
Comment	Optional user-entered comment field.	

Configure how many coils the RTAC will poll from the IED.

Table 2.15 Coils

Parameter	Description	Default
Coil Address	The bit address of the coil you want to operate.	Contiguous numbering from 0–n, where n = number of coils – 1.

Configure how many discrete inputs the RTAC will poll from the IED.

Table 2.16 Discrete Inputs

Parameter	Description	Default
Input Address	The discrete input address.	Contiguous numbering from 0–*n*, where *n* = number of coils – 1.

Configure how many holding registers the RTAC will poll from the IED.

Table 2.17 Holding Registers

Parameter	Description	Default
Register Address Start	The starting address of each holding register in the polled Modbus IED.	Contiguous numbering from 0–*n*, where *n* = number of inputs – 1. If Variation = 32 bits, the consecutive numbering will skip every other register to allow two registers per tag.
Register Address Stop	A noneditable field that indicates the ending register for this tag.	The value of Register Address Start + 0 if Variation = 16 bit. If Variation = 32 bit, the value is Register Address Start + 1.
Variation	Definition of the holding register size, type, sign, and configuration. You can swap the order of high and low bytes or registers by using the variation type. Variation types for single registers are as follows: LSB = least significant byte first; MSB = Most significant byte first; Variation types for multiple registers that are combined into one tag are as follows: LSR=Least significant register first; MSR=Most significant register first.	
Bit Reference	Definition for each bit in a holding register. This parameter is applicable only when holding register type = MDBC.	
Control Value	The initialized control value before the RTAC database is initialized at startup.	
Set Magnitude	The default instantaneous value that the RTAC will write to the IED in a write holding register message.	
Inst Magnitude	The default instantaneous value prior to initialization of the tag at startup.	
Magnitude	The default dead-banded value prior to initialization of the tag at startup.	
Dead Band	The number of units of change necessary to move the Inst Magnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag. An excursion of this dead band will cause the RTAC to generate a time-stamped event. Setting the dead band to 0 causes an event with every instantaneous magnitude the RTAC receives.	
Zero Dead Band	The number of units at or below which the Magnitude is forced to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value allowed for this point.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	

Parameter	Description	Default
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point.	

Configure how many input registers the RTAC will poll from the IED.

Table 2.18 Input Registers

Parameter	Description	Default
Register Address Start	The starting address of each holding register in the polled Modbus IED.	Contiguous numbering from 0–*n*, where *n* = number of inputs – 1. If Variation = 32 bits, the consecutive numbering will skip every other register to allow two registers per tag.
Register Address Stop	A noneditable field that indicates the ending register for this tag.	The value of Register Address Start + 0 if Variation = 16 bit. If Variation = 32 bit, the value is Register Address Start + 1.
Variation	Definition of the input register size and configuration. You can swap the order of high and low bytes or registers by using the variation type.	
Bit Reference	Definition for each bit in an input register. Applicable only when input register type = SPS.	
Number of Bytes	The number of bytes to read for this tag. Applicable only when Tag Type = STR.	
Inst Magnitude	The default instantaneous value prior to initialization of the tag at startup.	
Magnitude	The default dead-banded value prior to initialization of the tag at startup.	
Dead Band	The number of units of change necessary to move the Inst Magnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag. An excursion of this dead band will generate a time-stamped event. Setting the dead band to 0 causes an event with every instantaneous magnitude the RTAC receives.	
Zero Dead Band	The number of units at or below which the Magnitude is forced to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value allowed for this point.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point.	

Parameter	Description	Default
Inst Angle	For a CMV data type, the default instantaneous angle value prior to initialization of the tag at startup. This is only available in Modbus Server.	
Angle	For a CMV data type, the default dead-banded angle value prior to initialization of the tag at startup. This is only available in Modbus Server.	
String Value	Default string value before the RTAC is initialized on startup. Applicable only when Tag Type = STR.	

Set up the number of read coil polls you want to use. If the coil status addresses you have configured are noncontiguous, you can break up the polling scheme into multiple polls. Each message can poll for a group of contiguous coils. Configure the following fields for each coil status poll.

Table 2.19 Read Coil Polls

Parameter	Description	Default
Poll Number	Consecutive field indicating the poll entry.	Sequential from 1–*n*, where *n* = total number of polls.
Starting Address	Starting coil address for that poll. Remember that the RTAC uses 0 relative addressing.	
Quantity	The number of coil points read for that poll operation.	

Read Discrete Input Polls

The setup for Read Discrete Input Polls is the same as for Read Coil Polls.

Read Holding Register Polls

The setup for Read Holding Register Polls is the same as for Read Coil Polls.

Read Input Register Polls

The setup for Read Input Register Polls is the same as for Read Coil Polls.

Report Server ID Poll

The setup for Report Server ID Poll is similar to that for Holding Registers configuration. This poll will return the following three values:

- Modbus Server ID
- Run Indicator Status as defined by the Modbus IED vendor
- Server ID Data as defined by the Modbus IED vendor

GE Relay Event Collection

Event Collection

The RTAC can automatically collect COMTRADE event records and Sequential Events Recorder (SER) data from connected GE relays by using the Modbus protocol. To collect COMTRADE event records, perform the following steps:

Step 1. Insert a Modbus client and configure communications settings as needed to communicate with the GE relay.

Step 2. Click on **POU Pin Settings** and set **Enable_Event_Collection** to True.

Step 3. Under the **Settings** tab, adjust the Polling period between requests to locate New Events as needed. The RTAC uses this value to periodically check for new events from the relay.

The RTAC will poll the relay periodically and collect any new events that are present.

SER Logging

You can configure the RTAC to automatically collect SER data from a GE relay connected via Modbus and log that SER data into the RTAC SOE log. For each Modbus client, set the POU pin setting **Enable_ASCII_SER_Logging_Enabled** to True. Set the **ASCII SER Logging Poll Period** under the **Settings** tab to control how often the SER data are polled. To identify which device the SER came from, you can enter a unique device identifier by using the **Device GUID** setting. When the RTAC receives the events, it logs the SERs in the RTAC SOE log with the optional GUID name and a time stamp that is constructed from the response of the GE relay. If needed, convert the SER time stamp to the RTAC time reference with the **Adjust ASCII SER Logging Timestamps** setting.

Modbus Server Configuration

Configure Modbus Server or client connections to communicate via direct serial or Modbus TCP to polling masters (SCADA, HMIs, communications processors, etc.). You can configure the connection with a fixed client IP address, or as Anonymous, which means any client on the RTAC network can poll for Modbus data. However, the Modbus server only allows one client connection at one time and is designed for a "connect and stay connected" type of communications scheme. Create tags in a Modbus Server as placeholders for data that the RTAC will return to a polling client. Use the Tag Processor to map data to these tags from client IED tags or other tags in the RTAC database.

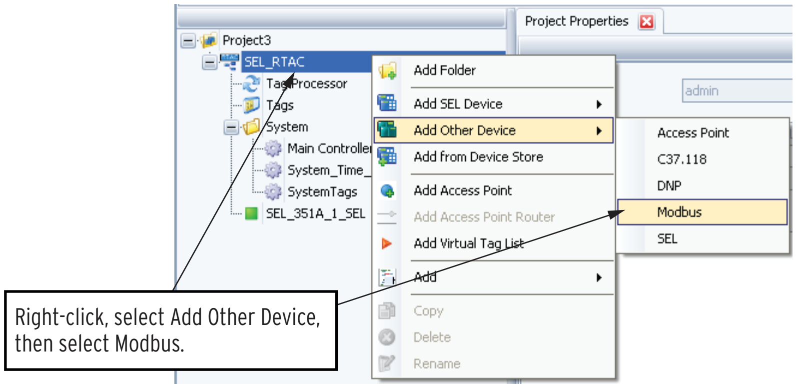


Figure 2.18 Insert Modbus Server

Give the device connection a unique name, and select the type of connection as shown in *Figure 2.3*.

The **Settings** tab contains all configurable items for communications. Check the Description column for details on each configuration item. Move the slider, or hover your cursor over the description to see the entire description text. Type any necessary comments in the Comment column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

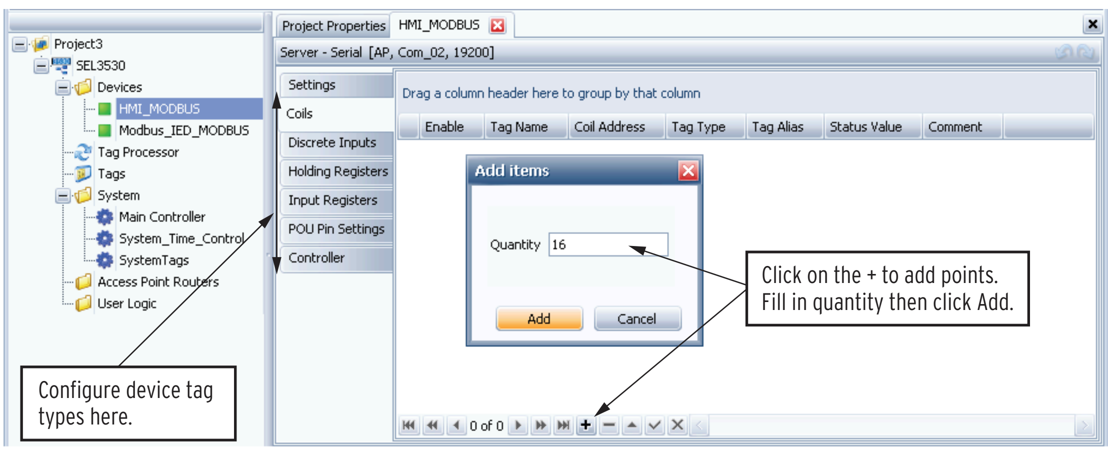


Figure 2.19 Add Modbus Server Tags

Add Server Device Data

The following steps detail adding server device data.

Step 1. Click on a device tag type tab to add tags.

Step 2. Click + to add tags (10,000 tag limit or 5000 for SEL-3505, per device tag type).

Creating only the number of tags necessary will optimize system performance.

Step 3. Change the names of the tags, if necessary.

Step 4. Change other tag-related information as necessary.

Repeat these steps to configure all server connections. When finished, configure the Tag Processor to populate these server connection tags with actual values.

Device Tag Type Configuration Parameters

Each device tag type has parameters that you must configure correctly for proper Modbus responses. Configuration columns have the same meaning as in Modbus client configuration (see *Modbus Client Configuration on page 62*) except for the addition of fields that have special meaning in Modbus server configuration. Use these fields to manipulate data returned in Modbus server configuration. For example, you can change the device tag type, value limits, etc.

Bit reference also differs between Modbus server and client configurations. In Modbus client configuration, the bit reference specifies the bit in the data stream that will populate the client tag. In Modbus server configuration, the bit reference specifies the bit in the server tag that returns to a polling client or master. Populate the specified bits in a server tag through values in the Tag Processor.

POU Pin Settings

See *POU Pin Settings (Advanced Usage) on page 37* for a description of the **POU Pin Settings** tab.

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

Example 2.3 Packed Status Bits From an IED Input Register

This example illustrates how to access individual status bits from a 16-bit packed input register. You can access individual bits by using the bit reference either in the protocol point setup or in the Tag Processor. This example uses the Modbus Client protocol tag configuration.

Step 1. On the **Insert** ribbon, select **Other** and then **Modbus**.

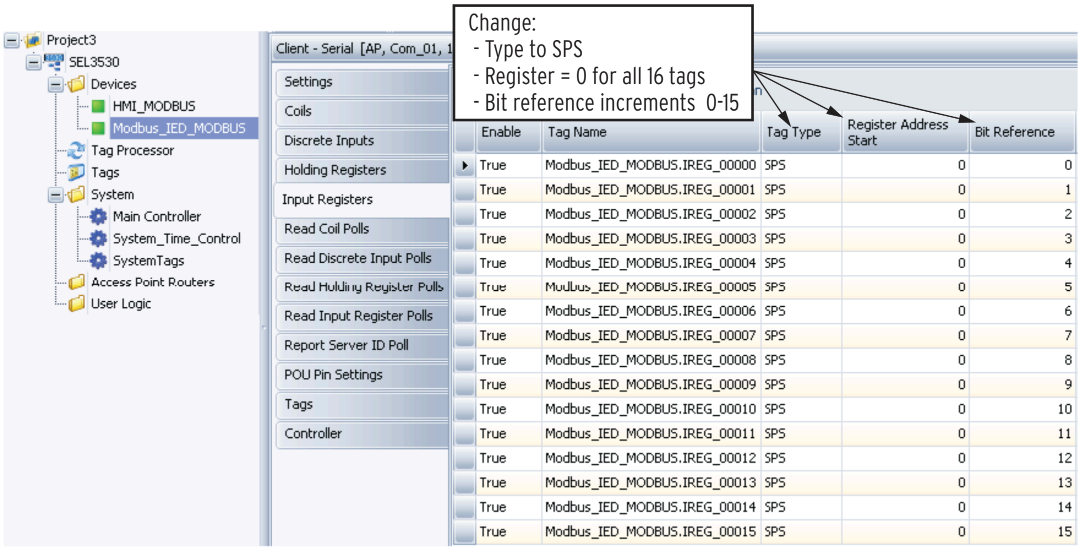
Step 2. Configure the device as Client Serial and change the Device Name to **IED1**.

Step 3. On the resulting **Modbus Client** configuration screen, click on the **Input Registers** tab.

Step 4. Add 16 Input Register tags and make the following changes:

- ➤ Change Tag Type to single point status (SPS).
- ➤ Change Register Address Start to 0 for all 16 tags.
- ➤ Press **<Ctrl+S>** to save.
- ➤ Change Bit Reference to increment from 0–15.

The RTAC will take a single bit *n*, where *n* = bit reference, from Register 0 and place it in IED_Modbus_IREG_*n*.



SEL Protocol

You can configure as many as 254 SEL protocol client and/or server devices in an SEL-3555 ACSELERATOR RTAC project type. For SEL-3350 ACSELERATOR RTAC project types, 100 client and/or server devices are supported. For all other ACSELERATOR RTAC project types, you can configure 64 SEL devices. You can split this limit between client and server devices as necessary. The ACSELERATOR RTAC supports SEL protocol on any of the RTAC serial or Ethernet ports. Use preconfigured device definition files (DDFs) that contain the necessary information to communicate with supported relays. Advanced users can also define custom DDFs to communicate with other SEL products. The RTAC SEL client can detect and notify if any polled devices have changes in their configurations.

SEL Client Configuration

The RTAC SEL Client supports the following:

- Auto-configure
- Auto-bps detection
- Compressed ASCII messages (e.g., History, Status)
- Fast Meter messages (e.g., Demand Meter, Meter, Peak Meter)
- Limited Fast Meter for SEL-100 and SEL-200 series relays
- IED configuration change monitor
- Access password management
- Unsolicited write messages

- User-configured flex parse messages
- SER logging to SOE log
- Event filtering and collection
- Remote access and file transfer

User-configured flex parse messages provide the tools to create custom messages and parse data from those messages into tags. See *Flex Parse Protocol on page 339* for more detail. There can be only one client association (a relay, for example) for each client connection. Only one lower-tier communications processor (RTAC, SEL-2032, etc.) is valid per client connection.

The ACSELERATOR RTAC software SEL client provides a DDF for each supported device. Each DDF is a template containing a superset of every point available for each SEL device. Select which points you want to use from the superset by setting Enable = TRUE in the data type tabs. Save the project and upload it into the RTAC. The RTAC will perform an autoconfiguration on each SEL device to which it is connected and compare the results with your configuration. Tags configured in the RTAC project that are not found during the autoconfiguration process will be marked with an invalid quality. Verify the RTAC configuration while offline and make the necessary changes to resolve the unmatched data tags.

You can create custom DDFs for SEL devices in ACSELERATOR RTAC. For example, you can modify an existing DDF, save it in the Device Store, and use it in other ACSELERATOR RTAC projects. Configure predefined SEL devices under the **SEL** tab.

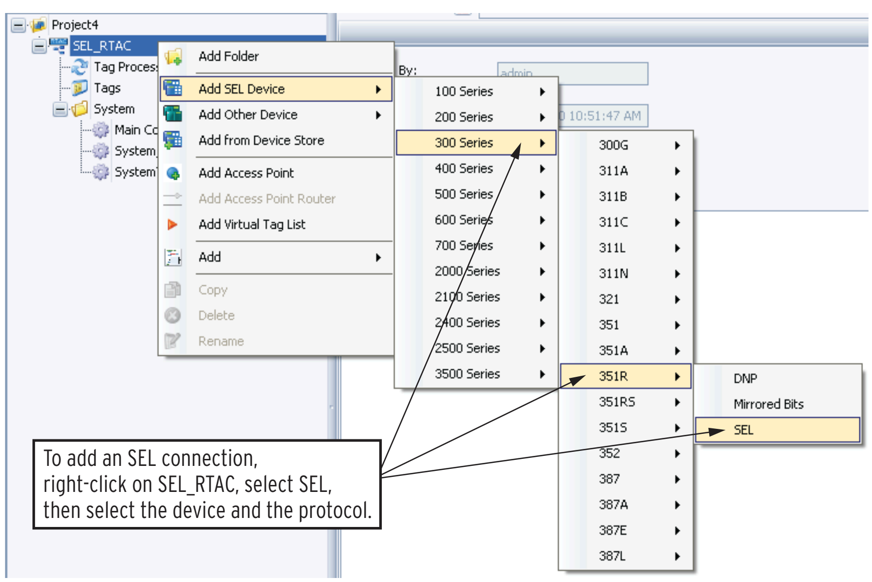


Figure 2.20 Insert SEL Client

To insert an SEL client, choose **SEL** under the Insert ribbon or right-click on **SEL_RTAC**. Select the IED series, the specific device, and then the SEL protocol.

Give the device connection a unique name, and select the connection type as shown in *Figure 2.1*. Because SEL protocol is a point-to-point protocol, multi-drop options are not available for selection.

The **Settings** tab contains communications and time configuration parameters. Check the Description column for details on each configuration item. Move the slider, or hover your cursor over a description to see the entire text of an item description. Type any necessary comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Configuration Monitor

Use the Check IED Configuration settings to monitor configuration changes in connected SEL devices. Click the **Advanced Settings** box under the **Settings** tab to expose these settings. The IED configurations are monitored as follows:

⚠**IMPORTANT**
Any modifications made to the Level 1 and Level 2 access passwords (via the password pins) while the project is running in the RTAC will be overwritten when new settings are downloaded to the RTAC. Before downloading a project to the RTAC, you should first read the currently running project to retain the current value of the Level 1 and Level 2 access passwords. Failure to do so will result in the inability to access the relays.

Step 1. The RTAC sends each user-defined Command String found under the **Check IED Configuration Commands** tab at the interval, in minutes, defined by the setting **Period between attempts to detect a change in the configuration of the connected IED**. This setting is found under the **Settings** tab.

Step 2. The RTAC calculates a 32-bit CRC on the response for all characters between STX and ETX.

Step 3. The response CRC is compared to a CRC stored in the non-volatile memory that was calculated for the same response sent previously.

Step 4. If a mismatch is detected, or if the response is not received within the **Timeout for a Check IED Configuration request**, the process repeats until one of the following occurs:

   a. **Number of retries for a Check IED Configuration request** is exhausted.

   b. The calculated CRC matches the stored CRC.

Step 5. If the CRC cannot be validated, the **Check_IED_Configuration_Error** output POU pin is pulsed for one task cycle.

Step 6. If one or more of the CRCs do not match the stored CRCs, then **Check_IED_Configuration_ Mismatch** output POU pin is pulsed and the calculated CRCs are stored in non-volatile memory and used the next time the Check_IED_Configuration POU pin is triggered.

If the number of CRCs stored in the non-volatile memory is not equal to the number of user-defined requests, all stored CRCs are deleted and the **Check_IED_Configuration_ Mismatch** output POU pin will pulse the next time the configuration monitor process completes.

Auto-configuration and transparent connections will interfere with this process. The process will resume once auto-configuration is completed and all transparent or direct transparent connections are closed.

Access Passwords

You can configure the Level 1 and Level 2 access passwords that the RTAC uses to connect to the relay under the **Settings** tab. You can also, through user logic, modify the Level 1 and Level 2 access passwords in real time by setting the POU pins Level_1_Password or Level_2_ Password and asserting the Write_Level_1_Password or Write_Level_2_Password. When Write_Level_1_Password or Write_Level_2_Password is asserted, the Level 1 or Level 2 access passwords are updated with the value of Level_1_Password or Level_2_Password, respectively. Through the use of custom logic, it is possible for the RTAC to perform password management and allow controlled engineering access to users, without the user knowing the relay passwords. Whenever passwords are modified, it is good practice to log the new POU pin password values, pass the new values to SCADA, or in some other way store the password values as part of a backup contingency plan.

As described in *SEL Server Configuration on page 85*, you can transparently connect to SEL clients through the SEL server connection (available in R134 firmware or above). Set **Enable Password Monitor** to True in each SEL client and the RTAC detects when the transparently connected user uses the **PAS** command to change the relay password, captures the password the user enters, and automatically updates the passwords in the SEL client in the RTAC project. Note that if you resend the original project to the RTAC, you will overwrite the new passwords and the SEL clients will no longer communicate properly to the connected devices. You must first read the project from the RTAC to obtain the updated passwords.

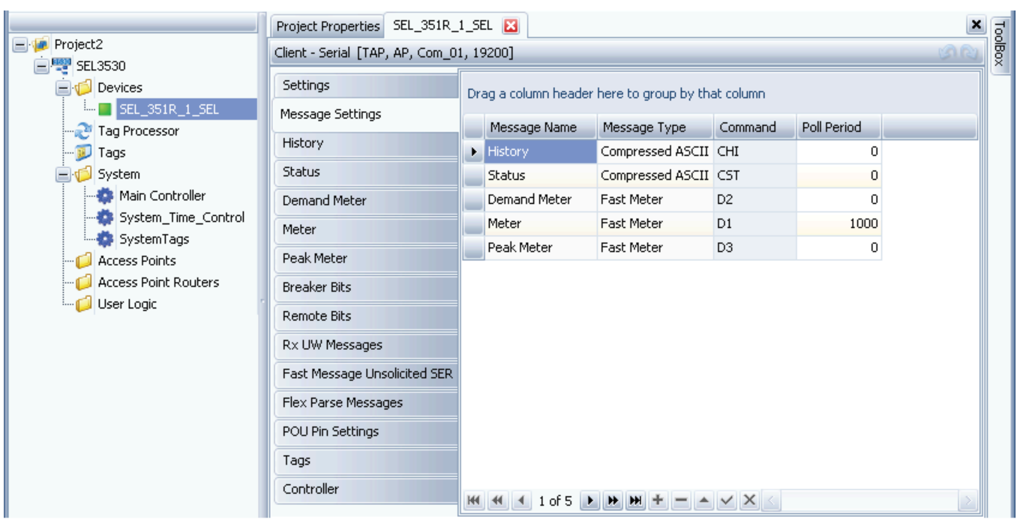


Figure 2.21 Message Settings

As shown in *Figure 2.21*, the **Message Settings** tab defines which messages the RTAC will send to the SEL IED and how often it will send them. You can adjust the default settings to alter the poll period or the commands the messages use.

Column Name	Description
Message Name	User-defined message name.
Message Type	Which SEL message type is used to get the data.

Column Name	Description
Command	Which SEL command is sent in the message to get the data.
Poll Period	User-defined time in milliseconds between messages. A value of 0 disables the message.

Look in the individual SEL IED instruction manual for the points you want to poll and for the messages and commands to poll them. The RTAC supports SEL Compressed ASCII (CASCII), SEL Fast Operate, SEL Fast Meter, Unsolicited Write, and SEL Fast SER. The SEL client uses the configuration on each message name tab to parse the data from messages. When the same data are available in CASCII and binary, always use the binary message data. CASCII is compressed, but it is neither as efficient nor as secure as binary messages. Binary messages are also uninterrupted by transparent engineering access, which does interrupt CASCII messages.

See SEL *Application Guide AG95-10: Configuration and Fast Meter Messages* for a comprehensive description of SEL binary messages.

Unsolicited SER Data

The RTAC can receive unsolicited SER data through two exclusive methods. With the first method, you can create and configure tags under the **Fast Message Unsolicited SER** tab. The RTAC identifies which tag to populate with received SER data based on either index or label. The index identifies sequentially which SER point the SEL IED identifies. For example, the first SER point configured in the IED is Index 1, the second is Index 2, etc. The label identifies the SER point by the label the SEL IED defined. You generally should refer to the SER point by label; the SER point order can change because it is configurable in the IED. The RTAC assigns all unsolicited SER data (states and times) to these tags.

To use the second method in assigning unsolicited SER data in the RTAC, do not create tags in the **Fast Message Unsolicited SER** tab. Instead, set the enable flag to True for binary tags in the **Meter** tab in the SEL client configuration. As the RTAC receives unsolicited SER data, it will search through the **Meter** tab for a binary tag label that matches the SER data table. When it finds a match, it assigns the time and state to that tag. This allows you to map an SEL binary tag (IN_201, for example) to a DNP binary tag in the Tag Processor and update the DNP tag automatically with unsolicited SER data.

NOTE
Use Copy (**<Ctrl+C>**) and Paste (**<Ctrl+V>**) to populate columns of Tag names and to duplicate Devices.

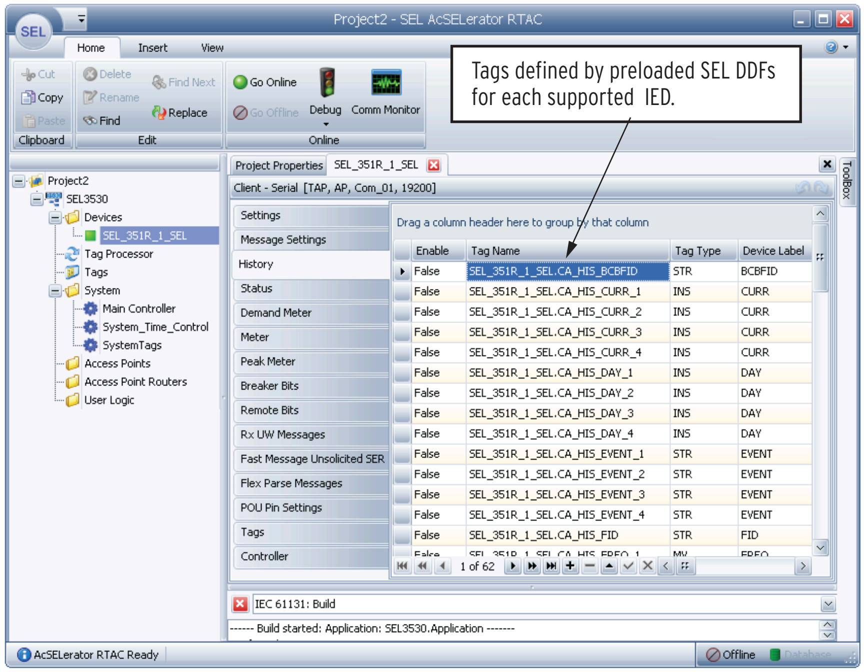


Figure 2.22 Tag Configuration

The ACSELERATOR RTAC contains predefined DDFs for each supported SEL IED, so you typically do not have to add data. If you want to enable or disable tags from the database, set the Enable field to True or False. Enabling only those tags you need helps optimize system performance. After making your modifications, you can use the  icon in the Device Store to create a custom DDF. Use the custom DDF in other projects that have the same SEL client requirements.

NOTE
Save the configuration by clicking **Save** or **<Ctrl+S>**.

Each message in the SEL protocol returns a specific group of data. The Message Names in the **Message Settings** tab define the labels for each data group setting tab. For example, **History**, **Status**, **Demand Meter**, etc., are each data groups from different messages. Configure each tag in each data group by clicking on that configuration tab. The ACSELERATOR RTAC grays out nonapplicable configuration columns.

Table 2.20 SEL Tag Configuration Parameters

Parameter	Description	Default
Enable	The processing enable flag. Set this flag to True to enable processing of this tag. Set this flag to False to disable processing.	False for most tags.
Tag Name	The default tag name contains the device name and tag description. You can change this name as necessary.	
Tag Type	The data type of this tag. See *Data Types on page 747* for more details.	
Device Label	The logical name of a group of data in an SEL protocol message. Typically, but not always, a label is unique in a message.	

Parameter	Description	Default
Tag Alias	An optional descriptive tag name. Use this tag alias anywhere in the RTAC system in place of the actual tag name.	
Device Label Instance	When a label in a message is not unique, the Device Label Instance identifies the occurrence of the label for the data element you want in the message.	0
Device Label Data Index	Specifies the data element within the data group under a particular label.	
Device Bit Label	The logical name of a group of binary data in an SEL protocol message. Typically, but not always, a Device Bit Label is unique in a message.	
Device Bit Label Instance	When a device bit label in a message is not unique, the Device Bit Label Instance identifies the occurrence of the device bit label for the data element you want in the message.	0
Status Value	The default value for the tag prior to initialization of the tag at startup.	False
Inst Magnitude	The default instantaneous value prior to initialization of the tag at startup.	
Magnitude	The default dead-banded value prior to initialization of the tag at startup.	
Dead Band	The number of units of change necessary to move the Inst Magnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag. An excursion from this dead band causes the RTAC to generate a time-stamped event. Setting the dead band to 0 causes an event for every time the RTAC receives an instantaneous magnitude.	
Zero Dead Band	The number of units at or below which the RTAC forces Magnitude to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value the RTAC allows for this point.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
Min Value	The minimum value the RTAC allows for this point.	
Inst Angle	The instantaneous angle value measured in degrees.	
Angle	A snapshot value of Inst Angle at the time of an Inst Magnitude excursion past the dead-band value.	
Actual Value	The default actual value of a BCR type tag before initialization on startup.	
String Value	The default string value the tag will have prior to initialization of the tag at startup.	
Comment	Optional user-entered comment field.	

Flex Parse Messages

Use Flex Parse Messages to create custom commands as long as 255 bytes and optionally parse the response messages. You can configure the RTAC to send a Command String to the connected SEL device at a configurable Poll Period interval. You can also configure the Start of Response Parse Expression and other parse expressions to enable the RTAC to determine when a valid response is received and to interpret the returned message and populate user-created tags. You can define parse expressions in SEL type class expressions or Perl 5 regular expressions (regex). It is important to note that the SEL client will not send user-defined flex parse messages to the configured device until after the auto-configuration is complete. For details, including SEL type class expressions, regex, and examples of using flex parse within the SEL client or using flex parse directly with other devices, please refer to *Flex Parse Protocol on page 339* in this instruction manual.

Add a flex parse command by performing the following steps:

Step 1. After adding an SEL device, click on **Flex Parse Messages**.

Step 2. Click on + to add a message, then click on **Add**.

Step 3. Configure the Command String, which is the message that will be sent to the SEL device.

Step 4. Configure the timing parameters to dictate how often the command should be sent and the time allotted for the command to complete.

Step 5. Configure the Start of Response Expression that the RTAC will compare with the incoming message to ensure it is parsing the correct message.

Step 6. Configure individual flex parse tags and parsing messages by using the information in *Flex Parse Protocol on page 339*. The configuration for flex parse messages added to an SEL client and flex parse messages added to a Flex Parse Protocol client is the same.

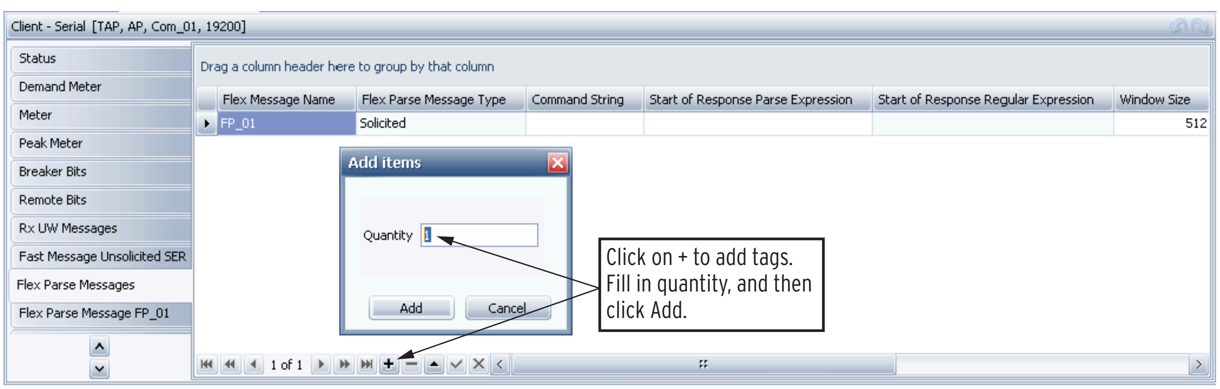


Figure 2.23 Add Flex Parse Message

Unsolicited Messages

Unsolicited messages write to specific memory locations on a remote device. An SEL client or SEL server configuration can receive unsolicited write messages from a remote device and can transmit unsolicited write messages to another device.

Configure transmit unsolicited write (Tx UW) messages in SEL client or server to create, define, and write tag values into memory locations in a target SEL IED. Configure receive unsolicited write (Rx UW) messages in SEL client or server to create and define tags and memory locations in the RTAC to which another SEL device will write. A remote SEL IED uses UW messages to write data to these memory locations.

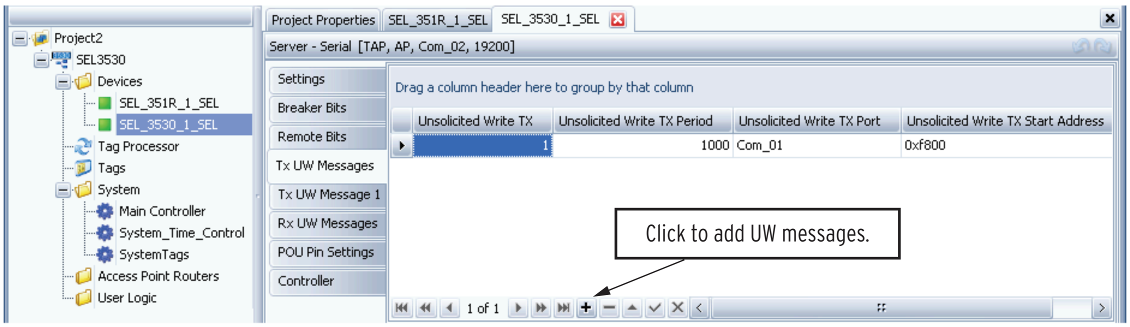


Figure 2.24 Unsolicited Write Messages

Add Tx UW messages to send data to the remote SEL IED. Click + to add a number of Tx UW messages. Configure each message.

Column	Description
Unsolicited Write TX	The order in which the RTAC sends the Tx UW messages.
Unsolicited Write TX Period	The time period in milliseconds between each UW transmission.
Unsolicited Write TX Port	The port the RTAC uses for this UW communication.
Unsolicited Write TX Start Address	The starting address in hexadecimal format on the receiving RTAC to which the transmitting device will be sending data.

Each Tx UW message you add causes ACSELERATOR RTAC to create a **Tx UW Message** tab. Use each **Tx UW Message** tab to add and configure data tags for its respective Tx UW message. The maximum number of registers an unsolicited message can contain is 117, so if you configure each tag in the UW Message tab to use one register, you can configure a maximum of 117 tags. But, you can configure as many as 2000 tags to address as many as 1872 individual bits (16 bits 117 registers) by configuring the following fields:

NOTE
Use Copy and Paste to populate columns of tag names and to duplicate devices.

Tag Type = SPS

Treat As = BOOL

Address Range Start = Use the same register address for each group of 16 tags.

Bit Index = Increment from 0–15 for each group of 16 tags of the same register address.

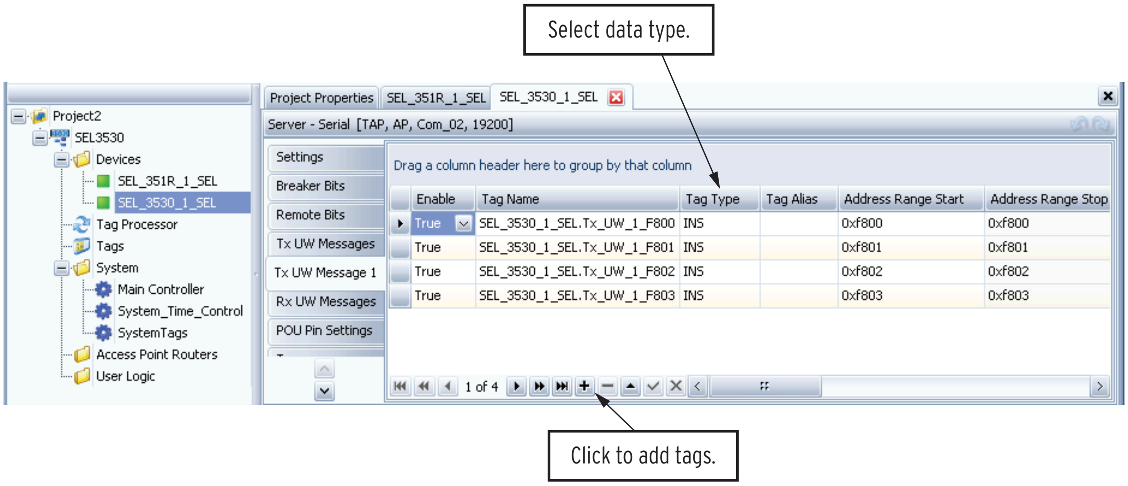


Figure 2.25 SEL Unsolicited Write Tags

To add Tx UW tags:

Step 1. Add an SEL server device.

Step 2. Click on the **Tx UW Message *n*** (where *n* is the Tx UW message number) tab to add and configure tags.

Step 3. Click + to add tags (10,000 tag limit or 5000 for SEL-3505, per device tag type).

Creating and enabling only the number of tags necessary helps optimize system performance.

Step 4. Change tag names, if necessary.

Step 5. Change other tag-related information as necessary.

Configure each Tx UW message group of data tags. The settings for each tab depend upon the Tag Type you have selected for that tag. ACSELERATOR RTAC grays out nonapplicable configuration columns. The configuration settings have the same meaning as those for SEL client configuration, except as the following table describes.

Setting Column	Description
Address Range Start	The beginning address on the target SEL IED to which the RTAC will write the Tx UW tag value.
Address Range Stop	The ending address on the target SEL IED beyond which the RTAC will not write the Tx UW tag.
Treat As	Convert the tag value to this data type when writing to the target. For example, to send two bytes of data per tag, change the Treat As type to INT. Remember to renumber the Address Range Start as needed if you change the Treat As data type.

Configure Rx UW messages and tags in the same way as Tx UW messages and tags. Remember that Rx UW messages define data that are coming from an SEL device. The remote SEL IED will populate the tags you define in the Rx UW messages by writing to the memory locations you have defined for each tag. Use the Tag Processor to map those values to other tags in the RTAC.

Remote Bits and Breaker Bits

Add remote bits (RB) and breaker bits (BB) as necessary. Link RB and BB tags in the Tag Processor to control tags in the RTAC. The RTAC will issue controls to a remote SEL IED through the configured RBs and BBs.

Fast Message Unsolicited SER

The RTAC will automatically route all SER data it receives from SEL devices to all configured SEL server device connections. No configuration is necessary on SEL client or SEL server device connections to enable this feature.

You can, however, also store SER data in the RTAC database and use this information in other protocols, user logic, etc. First, configure the SEL IED per the instruction manual for that IED to send SER data automatically to the RTAC. Enable the RTAC to store Fast SER data the IED sends by setting Enable = True in the **Fast Message Unsolicited SER** tab in ACSELERATOR RTAC. Enable the number of tags that the IED has configured for SER.

The RTAC will now store in the enabled RTAC database tags all SER data it receives. This data storage does not affect the automatic routing of SER data to all configured SER device connections.

SER Logging

You can configure the RTAC to automatically collect SER data from an SEL client and log that SER data into the RTAC SOE log. For each SEL client, set the POU pin setting **Enable_ASCII_SER_Logging** to True. Select the **Advanced Settings** check box under the **Settings** tab and locate the applicable settings under the Event section.

The ASCII SER Logging Collection Period is the interval at which the RTAC sends the **SER** command to the SEL relay to collect the events. To identify which device the SER came from, you can enter a unique device identifier by using the Device GUID setting. When the RTAC receives the events, it notifies each configured SEL server that new SERs have been collected and logs the SERs in the RTAC SOE log. The SER event is logged into the SOE with the optional GUID name and a time stamp that is constructed from the response of the ASCII SER command. If needed, modify the date format with the **ASCII SER Logging Date Format** setting. The command to collect the SER data is dependent on the SEL device with the **CSE** command taking precedence over the **SER** command.

Event Filtering and Collection

You can configure the RTAC to collect events automatically from each configured SEL client if the relay supports the compressed ASCII history (CHI) message. Depending on configuration, the RTAC can store event information (fault current, location, time of event, etc.) for the first event of the most recent fault and/or all event information in the RTAC database. The RTAC stores new event information in tags available to any RTAC protocols and applications from the event tab. Enable this feature by setting the POU pin Enable_New_Event_Filtering = True. The RTAC can also store as many as 512 records of all event information it retrieves from the compressed ASCII event (CEV) message into nonvolatile storage in the RTAC database. Enable this feature by setting the POU pin Enable_Event_Collection = True. Once the RTAC stores event information in the database, this information is available to SEL server connections configured in the RTAC through special event commands. These two event collection methods are neither exclusive nor dependent upon each other.

Configure New Event Filtering

Step 1. Insert an SEL client for the correct SEL relay type.

Step 2. On the **POU Pin Settings** tab, set **Enable_New_Event_Filtering** to True.

Step 3. Select the **Advanced Settings** check box to enable configuration of advanced settings. You can now edit the settings related to event filtering, collection, and reporting.

The event polling period dictates how often the RTAC sends a **CHI** command to determine whether there are any new events. If the relay replies to one of these solicited polls with new events, or if the relay sends an unsolicited short event report at any time, the RTAC knows at least one event is ready for filtering/collection/reporting. It is important to note that the **CHI History** command in the message settings tab populates tags in the **History** tab; it is not part of this event-collection process.

The RTAC defines the newest event as either a single event or the first in a group of events in which the time stamp is greater (by the configured New Event Lockout Period) than for adjacent events. If Report New Event On Detection = True, then the RTAC uses the new event data to populate the tags in the **History - New Event** tab. If Report New Event On Detection = FALSE or a new event is unavailable, the tags do not update unless the POU pin report_new_event is asserted.

If the POU pin Reset_New_Event is asserted, the RTAC replaces the data in the History - New Event tags with the present system time and configurations in the new event reset location value and with the new event reset event value. Additionally, if the POU pin Report_New_Event is asserted and the RTAC finds no new event after issuing a **CHI** command, it replaces the data in the History - New Events tags as if the Reset_New_Event pin had been asserted.

Configure Event Collection

Step 1. Insert an SEL client for a particular SEL relay (or use one already configured for event filtering).

Step 2. On the appropriate settings tab, set either **Enable Event Collection** or **Enable Comtrade Collection** to True.

Step 3. Select the **Advanced Settings** check box to enable configuration of advanced settings. Event polling period works the same as in new event filtering except that it is the period at which the RTAC will issue a compressed ASCII event (CEV) message to look for new events. The RTAC will also request a CEV report if it receives a short event report from the relay.

NOTE
You can use the Event Collection Parameter in the **Settings** tab as necessary to augment the CEV request. However, do not use TERSE or NSUM; this will result in a compressed event report that the RTAC does not recognize.

In event collection, all event information is gathered from the CEV reports and stored in nonvolatile RAM in the RTAC. Optionally, use the Event Collection parameter in the **Settings** tab to modify the **CEV** command that the RTAC sends to collect the events. For example, an entry in the Event Collection parameter of S16 L12 will result in the command **CEV *x* S16 L12** to the relay, where *x* is the event number and S16 L12 specifies 16 samples plus 12 cycles. Only the 12 most recent entries are collected from each relay, with a maximum of 512 records stored in the RTAC's memory. If 512 records already exist in the database, the next record stored will cause the oldest 64 records to be purged. When the RTAC collects events, it sends a message to each configured SEL server notifying it that there are new events ready for collection. You must set Enable_Event_Collection = TRUE for the RTAC to collect events. If Enable_New_Event_Filtering = TRUE but Enable_Event_Collection = FALSE, the RTAC will not store event data in the database. Neither will it notify configured SEL servers that there are event data available. In addition to collecting CEV-based events with the SEL client, starting in firmware version R147, COMTRADE collection is also supported for SEL IEDs that support the COMTRADE collection format. The RTAC supports storing as many as 1024 COMTRADE events at a single time. This 1024 COMTRADE collection count is shared with COMTRADE events collected through Modbus, MMS, and EtherCAT CT/PT modules. The COMTRADE event collection queue is separate from the 512 CEV collection count. You can download collected events in the RTAC web interface by selecting the **Event Collection** tab. Enter the start and end dates for the events you want to download and select **Download**. Starting in R147, COMTRADE collection is also available via FTP. This allows the SEL connection to use either the primary SEL protocol connection or a separate FTP connection to the SEL relay to collect COMTRADE events. The method for collecting events is configured in settings for the protocol connection. Some SEL relays have large COMTRADE files, which may prevent the files from being collected in a reasonable amount of time via the primary connection. For these files, SEL recommends using the FTP collection method. Some SEL relays have multiple types of event reports; for these relays, configure the RTAC to collect all desired event types. Relay-specific instruction manuals document what types of events each relay supports. Some examples of event types include traveling-wave (TW), high-resolution, and 10 kHz (TDR). (Note that the RTAC collects high-resolution event types by default because that is the most common event type across SEL relays.)

Event Collection File Naming

Starting in R148 and later, the format for how an event is named on the RTAC file system is a configurable option. Events can maintain the standard format, which was used prior to R148, or the event can be named in compliance with the COMNAME standard.

For CEV collected events, the formatting shall be as follows.

If Event_Name_Format is configured as default:

- Format: SEL_event_<sequence # -- 0 - (n-1) events>.cev
- Example: "SEL_event_3.cev"

If Event_Name_Format is configured as COMNAME:

- Format: <Event's Faulted Timestamp -- YYMMDD,hhmmss000000>, <Time Code+t>,<COMNAME_Event_Name_String>,<Event Number -- 0, 10000-65535>.cev
- The time stamp stored with the .cev file (displayed by the Web), shall be the time stamps extracted from the CHI response.
- The value of Time Code (i.e., UTC Offset) shall be derived from interpreting the Event's Trigger time stamp and the SEL Client Date-Time settings.
- The event reference will be derived from the "REF_NUM" value (10000–65535) from the CHI response, if it exists; otherwise, it will be 0.
- Example:
  - 170220,215117461,+2h30t,StationId,DeviceId,Company,10009.cev

For COMTRADE collected events, the formatting shall be as follows.

If Event_Name_Format is configured as default:

- The file name of the zipped COMTRADE event files and ZIP file shall be the base file name of the COMTRADE Event files from the relays file directory response.
- The entire ZIP file name shall be uppercase; the cfg/dat/hdr file names may be mixed-case and the extension shall be lowercase.
- The time stamp stored with the COMTRADE Event files (displayed by the Web), shall be the time stamps extracted from the relay's directory response. If the relay's directory response does not include time stamps for the COMTRADE Event files, the time of retrieval shall be used.
- Example:
  - "170220,215117461,2T,T400L,0L#53,KAY,TDR,10009.CFG/DAT/HDR" becomes:
    - 170220,215117461,2T,T400L,0L#53,KAY,TDR,10009.ZIP
    - 170220,215117461,2T,T400L,0L#53,KAY,TDR,10009.cfg
    - 170220,215117461,2T,T400L,0L#53,KAY,TDR,10009.dat
    - 170220,215117461,2T,T400L,0L#53,KAY,TDR,10009.hdr
  - "StationA,SEL-T401L,SEL-T401L,200313,232558.5775, -1h30t,No.10000,MHR.CFG/DAT/HDR" becomes:
    - STATIONA,SEL-T401L,SEL-T401L,200313,232558.5775, -1H30T,NO.10000,MHR.ZIP
    - StationA,SEL-T401L,SEL-T401L,200313,232558.5775, -1h30t,No.10000,MHR.cfg
    - StationA,SEL-T401L,SEL-T401L,200313,232558.5775, -1h30t,No.10000,MHR.dat
    - StationA,SEL-T401L,SEL-T401L,200313,232558.5775, -1h30t,No.10000,MHR.hdr
  - "HR_10001.CFG/DAT/HDR" becomes:
    - HR_12345.ZIP
    - HR_12345.cfg
    - HR_12345.dat
    - HR_12345.hdr

If Event_Name_Format is configured as COMNAME:

- The name of the zipped COMTRADE event files and ZIP file shall be as follows:
  - Format: <Event's Faulted Timestamp -- YYMMDD,hhmmss000000>, <Time Code+t>,<COMNAME_Event_Name_String>,<Event Type -- e.g., "HR", "TDR", etc.>,<Event Number -- 10000-65535>.CFG/ DAT/HDR/ZIP
- The entire ZIP file name shall be uppercase; the cfg/dat/hdr file names may be mixed-case and the extension shall be lowercase.
- Presence and accuracy of milliseconds and microseconds in the Event's file name shall be dependent on the presence and accuracy of the Event's Trigger time stamp.
- The time stamp stored with the COMTRADE Event files (displayed by the Web), shall be the time stamps extracted from the relay's directory response. If the relay's directory response does not include time stamps for the COMTRADE Event files, the time of retrieval shall be used.
- The value of Time Code (i.e., UTC Offset) shall be derived from interpreting the Event's Trigger time stamp and the SEL clients Date-Time settings.
- Example:
  - "170220,215117461,2T,T400L,0L#53,KAY,TDR,10009.CFG/DAT/ HDR" becomes:
    - 170220,215117461,2T,STATIONID2,DEVICEID2,COMPANY2, TDR,10009.ZIP
    - 170220,215117461,2t,StationId2,DeviceId2,Company2,TDR, 10009.cfg
    - 170220,215117461,2t,StationId2,DeviceId2,Company2,TDR, 10009.dat
    - 170220,215117461,2t,StationId2,DeviceId2,Company2,TDR, 10009.hdr
  - "StationA,SEL-T401L,SEL-T401L,200313,232558.5775,-1h30t, No.10000,MHR.CFG" becomes:
    - 200313,2325585775,-1H30T,STATIONID3,DEVICEID3, COMPANY3,MHR,10000.ZIP
    - 200313,2325585775,-1h30t,StationId3,DeviceId3,Company3, MHR,10000.cfg
    - 200313,2325585775,-1h30t,StationId3,DeviceId3,Company3, MHR,10000.dat
    - 200313,2325585775,-1h30t,StationId3,DeviceId3,Company3, MHR,10000.hdr
  - "HR\12345.CFG/DAT/HDR" becomes:
    - 200102,030405666,+2H30T,STATIONID1,DEVICEID1, COMPANY1,HR,12345.ZIP
    - 200102,030405666,+2h30t,StationId1,DeviceId1,Company1, HR,12345.cfg

- 200102,030405666,+2h30t,StationId1,DeviceId1,Company1, HR,12345.dat
- 200102,030405666,+2h30t,StationId1,DeviceId1,Company1, HR,12345.hdr

Remote Access and File Transfer

The SEL client supports remote engineering access to the connected SEL device. There are two methods for configuring engineering access. The first method is to configure a transparent connection by using an access point and access point router (APR). For further information on this method, refer to *Section 4: Engineering Access*. The second method uses the SEL server device type. For more information on this method, refer to *SEL Server Configuration on page 85*.

When initiated through an SEL server, remote engineering access attempts to open a secondary Ethernet connection to the SEL client device. For serial connections, access is only available through the primary data polling connection. If the SEL client device receives no response within the time configured in the **Secondary Transparent Connection Timeout**, and **Allow Transparent on Polling Connection** is set to True, the SEL server establishes an engineer access session on the primary data polling connection. To disable remote engineering access for an SEL client, set **Disable Remote Access** to True in the **Settings** tab.

Both engineering access methods support file transfer for reading settings from the connected SEL IED. When engineering access is successfully established through a transparent connection or through an SEL server device, the SEL client monitors the connection for a file transfer initiation. If the SEL client detects a file transfer initiation, the connection switches to a direct transparent connection. If the connection is on the primary data polling connection, all data polling is suspended until the file transfer succeeds or fails. To disable file transfer through the SEL client, set **Enable File Transfer** to False in the **Settings** tab.

SEL Server Configuration

The RTAC supports one primary SEL protocol communications session with one upper-tier SEL device (such as another RTAC) per SEL server connection. Create SEL unsolicited write tags in an SEL server as placeholders that the server will use to exchange data with a polling RTAC. Use the Tag Processor to map those tags to client IED or other source tags in the RTAC database. You can also configure RBs and BBs for control messages. The primary SEL protocol session is associated with all SEL protocol data exchanges mentioned previously as well as unsolicited SER reporting and event notifications.

Each SEL server in the RTAC supports only one primary SEL protocol client connection. If more than one SEL client will connect to this RTAC, create another SEL server for each client connection.

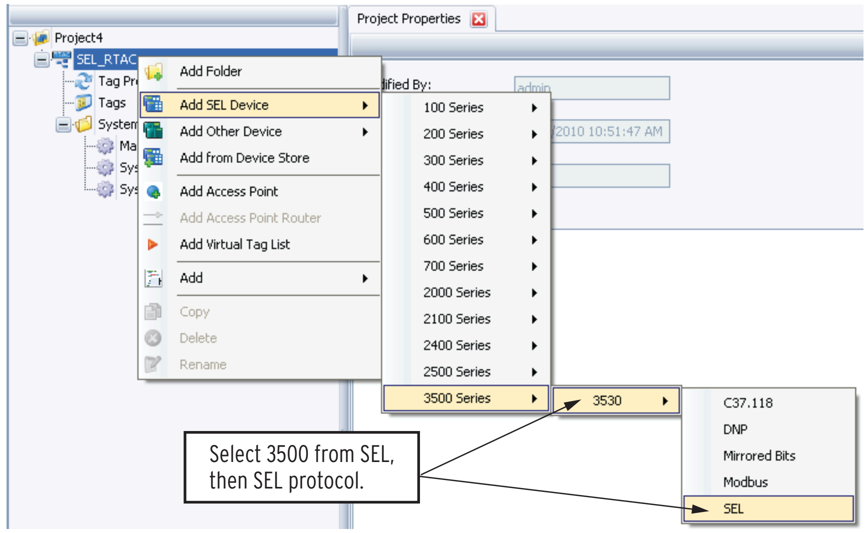


Figure 2.26 SEL Server

For example, insert an RTAC from the SEL ribbon menu under the **Insert** ribbon or from right-clicking on **SEL_RTAC** in the device pane. Select **3500 Series**, then **3530**, then **SEL** protocol.

Give the device a unique name, or accept the default as shown in *Figure 2.3*. Select either **Server – Serial** or **Server – Ethernet Tunneled Serial**.

The **Settings** tab contains all configurable items for communication and time. Check the Description column for details on each configuration item. Move the slider, or hover your cursor over the description to see the entire description text. Type any necessary comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Multiple Simultaneous Connections Support

The SEL server supports multiple connections. All connections other than the primary connection are secondary connections and will only allow ASCII command line sessions. If **Allow Anonymous SEL IP Clients** is set to True, then all connections are accepted. If **Allow Anonymous SEL IP Client** is set to False, then only the IP addresses configured in the **Client IP Address** and **Engineering Access IP Address** may connect to the SEL server.

When a single connection is initiated through the SEL server to an SEL client, the server either creates a new connection to the client or uses the primary connection, which provides for binary data transfer. Client connection type and the settings in the client device determine the behavior. Clients configured for serial connections automatically use the primary connection because serial communications channels only support one connection. Clients configured for Ethernet communication create a secondary connection if configured in the client settings. See *SEL Client Configuration on page 70* for more information.

When engineering access initiates multiple connections to the same SEL client device, the SEL server attempts to create a new connection to the client if configured in the client settings. Many SEL IEDs support multiple concurrent Telnet sessions. For every connection you initiate through the SEL server, the server attempts to create a unique session to the IED until the IED rejects the connection request. At that point, the server may attempt the primary connection if the client settings allow it.

Primary SEL Protocol Connection Configuration

The primary SEL protocol connection is associated with sending all unsolicited message notifications (including SER, event report oscillography, and SOE notifications) and such other SEL protocol functions as unsolicited writes and remote bit operation. An SEL server device accepts only one primary connection. If **Allow Anonymous SEL IP Clients** is set to False, the Client IP Address and Client IP Port settings determine the primary SEL protocol connection. If **Allow Anonymous SEL IP Clients** is set to True, the primary connection is associated with the first successful connection to the SEL server.

Unsolicited Event and SOE Notification

You can configure the SEL server to send notifications to other applications when new event report oscillography and SOE data are available for collection. To configure this capability, navigate to the **POU Pin Settings** tab in the SEL server. To send notifications when the RTAC has new SOE data available, set the default value for **Enable_Unsolicited_ASCII_SER_Notice_TX** to True. To send notifications when new event report oscillography is available, set **Enable_Unsolicited_Event_Report_TX** to True. There are two exclusive event report oscillography notifications because there are two different automated methods for collecting event report oscillography from the RTAC. The legacy method uses the commands available in the SEL server to read event reports through an ASCII data session. See *Table 2.21* for the **CAR *archive*** and other SEL server commands. The database collection method uses a direct connection to the RTAC database over a TLS tunnel, which encrypts the data transfer. COMTRADE event collection is supported during use of the database collection method. Select an event collection mechanism in accordance with the security and infrastructure requirements of your organization.

To configure which notification the SEL server sends, navigate to the **Settings** tab and configure the **New Event Notification**. For legacy event collection, select **Legacy**, and for the database collection method, select **Encrypted Database**. If you select **Legacy**, the SEL server sends a new notification when new CEV events are available for collection. If you select **Encrypted Database**, the SEL server sends a new notification when either a new CEV or COMTRADE event is available. When new SOE data are available, the SEL server sends the NEW SOE message. The respective notifications repeat every minute until the events report oscillography or the SEL client acknowledges the SOE data.

Unsolicited Messages

See *SEL Client Configuration on page 70* for a discussion of unsolicited messages.

Remote Bits and Breaker Bits

Add RBs and BBs as necessary. The RTAC will receive controls from a remote SEL client through the use of configured RBs and BBs. Link RB and BB tags in the Tag Processor to control tags in the RTAC.

SEL Server SER Data

As discussed in *SEL Client Configuration on page 70*, the RTAC automatically routes all SER data from all SEL client connections to all configured SEL server connections. The RTAC can also generate local system SER data from any configured RTAC tags. The RTAC automatically routes all local system SER data to all configured SEL server connections.

Step 1. Under **System**, click **SystemTags**, then click on the **SEL Server SER** tab.

Step 2. Click on the + icon to add the number of SER tags necessary.

Step 3. Configure the Tag Name and SEL Label, if needed.

Step 4. Map the new SER tags as Destinations in the Tag Processor.

Step 5. Map Source Expressions in the Tag Processor to populate the SER tags with data.

The RTAC will generate an SER message for each change of state it detects in the defined system SER tags. All configured SER server device connections will transmit these unsolicited SER messages.

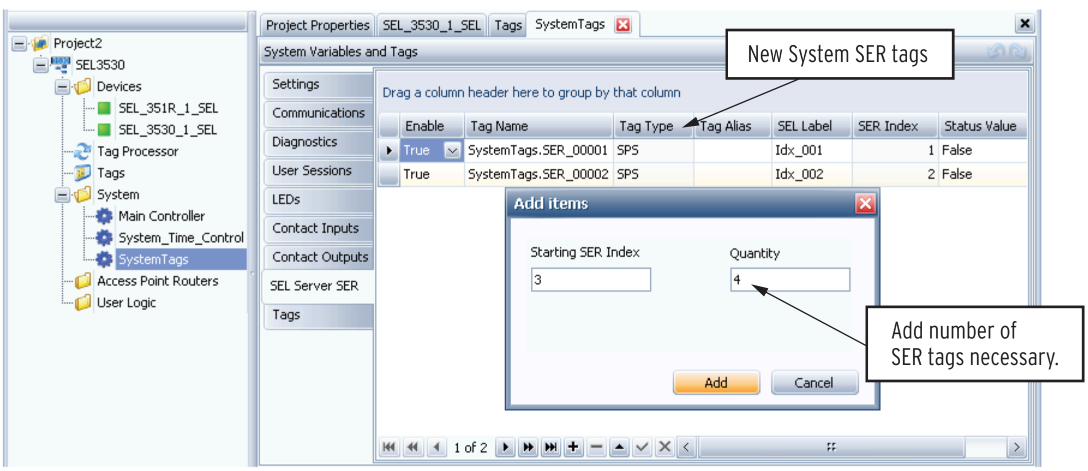


Figure 2.27 System SER Tags

SER Server Commands

The SEL server for the RTAC is similar to other SEL devices in supporting user authentication and other commands to provide an interactive user session. From the advanced options under the settings tab, set Perform Authentication = True if you want the RTAC to authenticate users before allowing access to other SEL server commands. Login usernames and passwords are those you configure through the RTAC web interface as local accounts or accounts you manage through an LDAP central authentication server. If user authentication is turned on, you can log in to the SEL server through a terminal emulation program as you would any SEL relay or IED. Enter the correct username and password, then enter **ACC** to reach Level 1 access and **2AC** to reach Level 2 access. If user authentication is not enabled, you must enter **ACC** and then use the password OTTER to enter Level 1 access. Enter **2AC** with password TAIL to enter Level 2 access. You can change the default passwords of OTTER and TAIL using the **PAS** command from Level 2. SEL server authentication has three access levels associated with RTAC user account levels:

- Level 0: Attain this level upon successful login. This is also the level you access if client authentication is disabled.
- Level 1: Access Level 1 by entering the **ACC** command. User manager and monitor local system roles can attain Level 1 access.
- Level 2: Access Level 2 by entering the **2AC** command. Administrator and engineer local system roles can attain Level 2 access. From Level 2 you can change the default SEL server ACC and 2AC passwords.

Table 2.21 lists SEL server commands.

The **POR** command provides an immediate direct transparent or transparent connection to configured SEL clients on the indicated port. For example, to transparently connect to an SEL device on Port 1 without using an Access Point Router, follow these steps:

Step 1. Configure an SEL server on the Ethernet tunneled serial or serial connection, as desired.

Step 2. Load the project into the RTAC.

Step 3. Connect to the SEL server by using the Ethernet serial tunneled IP port number or directly with a serial cable, depending on your configuration.

Step 4. If you have configured Authentication as True, log in as described in this section.

Step 5. Enter the command **POR 1** and press **<Enter>**. You are transparently connected to the SEL device configured on Port 1. Alternatively, enter the command **POR 1 D** and press **<Enter>**. You now have a direct transparent connection to the SEL device configured on Port 1.

Step 6. Press **<Ctrl+D>** to exit the transparent connection and return to the SEL server.

If you change the SEL device password while transparently connected, you will interrupt normal SEL client polling operations. To resolve this, you can either change the password in the SEL client configuration in the RTAC project and resend the project, or you can configure the system to do it automatically. Set the advanced setting **Enable Password Monitor** on each SEL client to True and resend the project. Each time you transparently connect to one of the SEL clients through the SEL server, the RTAC monitors for password changes and automatically updates the configured passwords in each affected SEL client.

⚠ **IMPORTANT**

You must read the project from the RTAC to ensure you have the latest passwords. Resending an older project to the RTAC will erase passwords that have been stored during a transparent connection password monitoring activity.

Table 2.21 SEL Server Commands (All Commands Access Level 1)

Command	Description
ACC	Gain access to Level 1. Password = OTTER if Perform Authentication = FALSE.
2AC	Gain access to Level 2. Password = TAIL if Perform Authentication = FALSE.
ID	Display RTAC ID information.
CAR SUM	Retrieve event report summary.
CAR SUM ALL	Retrieve event report summary for all ports.
CAR SUM port_#	Retrieve event report summary for specified port.
CAR READ port_# event_#	Retrieve detailed event data for specified port# and event#. Each event has a unique event#.
CAR ACK ALL	Acknowledge all events.
CAR ACK [port_#] ALL	Acknowledge all events for specified port number.
CAR ACK [port_#] [event_#]	Acknowledge all events for specified port number and/ or event number.
PAS	From the 2AC level, change the SEL server default user passwords for either Level 1 or Level 2. Usage is **PAS *n* <Enter>** (where *n* is either 1 or 2) and follow the prompts to change the default password.
POR *n*	Transparently connect to the SEL client configured on the port number indicated.
POR *n* DIRECT	Direct transparent connection to the SEL client configured on the port number indicated.
STA	Shows other clients who have active direct/transparent connections.
WHO	Lists information for configured SEL clients.
FIL SHO <file name>	Outputs content of listed file in ASCII format. SEL server setting **Perform Authentication** must be True. The full file path is required and must be preceded by FILES/ (i.e., FILES/<directory path and file>). For example, if the FTP server is enabled, a valid command would be **FIL SHO FILES/ftplog.txt**.

Example: CAR SUM

The CAR SUM response format is as follows:

```
"NEWEVENT=00000000000..,", "CSUM",
```

Where CSUM is the message checksum, and the zeros following NEWEVENT total 64 characters. Each character in the response corresponds to a virtual port on the RTAC. The first character is for **Port 1**, the second is for **Port 2**, and so on. If there are unacknowledged reports for a given port, the RTAC sets the corresponding character to a value of one in the **CAR SUM** response. If no ports have unacknowledged events, the **CAR SUM** response contains 64 zeros.

Example: CAR SUM ALL

The **CAR SUM ALL** responds with a summary of all event reports for all ports configured for event collection. The event summary contains a header row followed by zero or more rows of data. Data rows are organized in descending chronological order.

Command:

CAR SUM ALL

Response:

```
"EVENT ID","FID","PORT","MONTH","DAY","YEAR","HOUR","MIN","SEC",
"MSEC","USEC","EVENT","CURR","TARGETS","STATUS","CSUM"

227,"SEL-421-R127-V0-Z012011-D20090218",3,7,6,2010,10,25,21,123,
"OVR",0,"","UNACK","CSUM"

228,"SEL-421-R127-V0-Z012011-D20090218",3,7,6,2010,10,24,21,123,
"TRIG",0,"","UNACK","CSUM"

229,"SEL-421-R127-V0-Z012011-D20090218",3,7,6,2010,10,23,21,123,
"OVR",0,"","UNACK","CSUM"

230,"SEL-421-R127-V0-Z012011-D20090218",3,7,6,2010,10,22,21,123,
"TRIG",0,"","ACK","CSUM"

231,"SEL-351-R225-V0-Z012011-D20080315",4,7,6,2010,10,22,10,123,
"TRIG",0,"","ACK","CSUM"

232,"SEL-351-R225-V0-Z012011-D20080315",4,7,6,2010,9,30,21,123,
"OVR",0,"","UNACK","CSUM"

233,"SEL-734-R325-V0-Z012011-D20060520",2,7,5,2010,6,22,21,673,
"TRIG",0,"","UNACK","CSUM"

234,"SEL-734-R325-V0-Z012011-D20080520",16,7,1,2010,11,43,31,225,
"OVR",0,"","UNACK","CSUM"
```

Example CAR READ

Command:

CAR READ 3 232

Response:

```
"CMD","EVENT ID","FID","CSUM"

"CEV 1 S8","232","SEL-421-R127-V0-Z012011-D20090218","CSUM"

[Event Report. 232]
```

where Event Report. 232 is the response to the **CEV 1 S8** command associated with Port 3 on the RTAC. The RTAC transfers the report in CASCII format.

Password Command and Managing User Accounts

Starting in R145-V0 users are able to manage passwords for local accounts on the RTAC. In order for this functionality to be active the SEL server setting **Perform Authentication** must be true. If this setting is false then the password command will work in the same manner as specified in *Table 2.21*.

Example Password Command (R145-V0 and later):

Command: **PAS**

Response: `User Name: ?`

Command: After entering the account name to change the password for.

Response: `New Password: ?`

Command: Enter a new desired password for specified local account.

Response: `Confirm New Password: ?`

Command: Re-enter password to confirm.

Response: `Password Changed`

POU Pin Settings

See *POU Pin Settings (Advanced Usage) on page 37* for a description of the **POU Pin Settings** tab.

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

MIRRORED BITS

Overview

MIRRORED BITS communications is a direct relay-to-relay communications protocol that allows IEDs to exchange information quickly, securely, and with minimal expense. The MIRRORED BITS message consists of eight transmit bits (TMBs) and eight receive bits (RMBs). The RTAC sends TMBs to IEDs. IEDs send TMBs to the RTAC, which receives these as RMBs.

The RTAC communicates MIRRORED BITS as would a relay, but it also acts as a data gateway between MIRRORED BITS and other protocols or programs running in the RTAC. For example, you can map MIRRORED BITS RMBs to DNP digital inputs to interface to a SCADA system. You can map control tags from user-defined logic or a remote client to TMBs to issue high-speed controls to an SEL IED through MIRRORED BITS.

MIRRORED BITS Configuration

Because MIRRORED BITS is a peer-to-peer serial communications protocol, there are no client, server, or Ethernet options. You can configure MIRRORED BITS on an EIA-485/EIA-422 or EIA-232 serial line, but this configuration does not support multi-drop communication. You can configure as many MIRRORED BITS communications channels on one RTAC as there are serial ports not already used for another purpose.

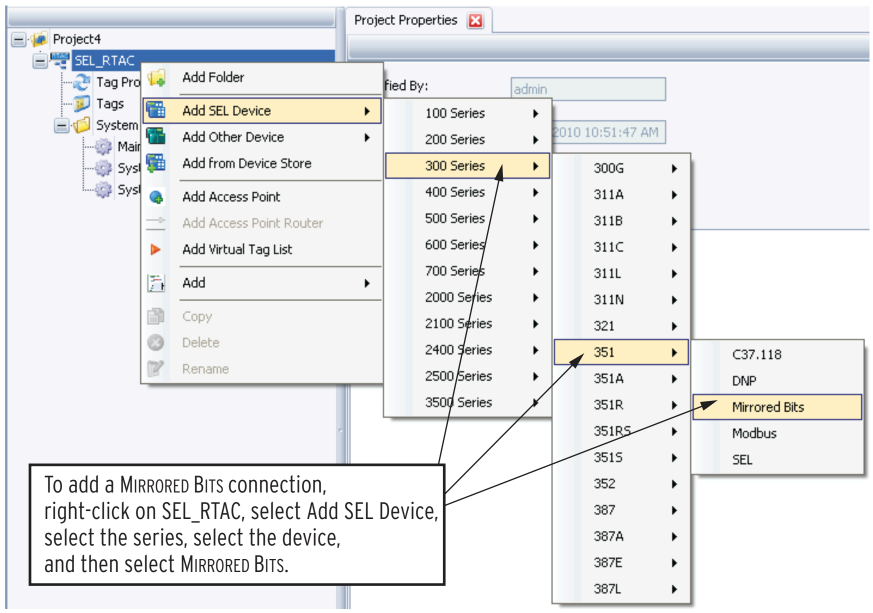


Figure 2.28 Insert MIRRORED BITS Device

Give the device connection a unique name, and select a connection type as shown in *Figure 2.1*. Note that MIRRORED BITS is a point-to-point serial protocol, so the only connection option is serial.

MIRRORED BITS is a peer-to-peer protocol designed for one-to-one high-speed serial communications; it cannot be multidropped. Insert one of these types of connections for each communications port on the RTAC that connects to a MIRRORED BITS IED connection via EIA-232 or EIA-485/EIA-422.

The **Settings** tab contains all configurable items for communications. Check the description column for details on each configuration item. Move the slider, or hover your mouse over the description to see the entire description text. Type any necessary comments in the blank column to the far right. Right-click on the title bar above the column headings or click **Options** to show advanced configuration settings. The following text explains some of the MIRRORED BITS-specific settings.

MIRRORED BITS Message Type	
MB	The standard MIRRORED BITS configuration using six data bits, one parity bit, and one stop bit for data encoding.
MB8	MIRRORED BITS using six data bits, one parity bit, and two stop bits for data encoding.
MBT9600	Provide for use over a Pulsar MBT-9600 modem. Set data rate = 9600 A delay of at least 2 ms shall be inserted after each outgoing message RTS shall be reset (to a negative voltage at the EIA-232 connector)
Accept Receive Identification	
	The incoming MIRRORED BITS message will have an encoded address (historically called RXID). Configure this receive address to match the address in the message. The received message is only valid if its RXID matches the Accept Receive Identification. The MIRRORED BITS protocol driver will reject the message if the receive ID in the message does not match the receive address you configured.
Send Transmit Identification	
	The RTAC will encode the outgoing MIRRORED BITS message with this configurable transmit address (historically called TXID). The receiving device will match this address with its receive address to ensure that it has received a valid message.
Receive MIRRORED BITS Bad Channel Pickup	
	The number of seconds a MIRRORED BITS channel must be without communications before the RTAC considers the channel to be no longer functioning. If the channel drops out for more than this configurable number of seconds, the Rx_Channel_Bad output pin asserts.
PPM MIRRORED BITS Unavailable Channel Pickup	
	Used to detect degraded communication on this MIRRORED BITS channel. Communication is still occurring, but the RTAC is dropping some number of packets. The PPM setting specifies an acceptable bad or dropped packet error rate in parts per million (PPM). If the dropped packet rate exceeds this setting, the output pin Rx_Channel_Unavailable asserts. The present value of this calculation is contained in the POU output pin Rx_Channel_Unavailable_PPM. For example, if the RTAC were to drop half the packets, the PPM error rate would be 500000.
Transmission Mode	
	Paced mode will send MIRRORED BITS at a rate of one message every 3 ms to allow for communications with relays that only communicate every quarter cycle (every 4 ms at 60 Hz). Set Transmission Mode to Normal to communicate to other RTACs and for 400-series relays.
Loopback Timeout	
	Automatically disconnect loopback mode after this time-out. Note that the time-out does not initiate loopback. Initiate loopback by forcing the Loop_Back_Without_Data POU pin. See *Example 2.4* for details of performing a MB loopback test.

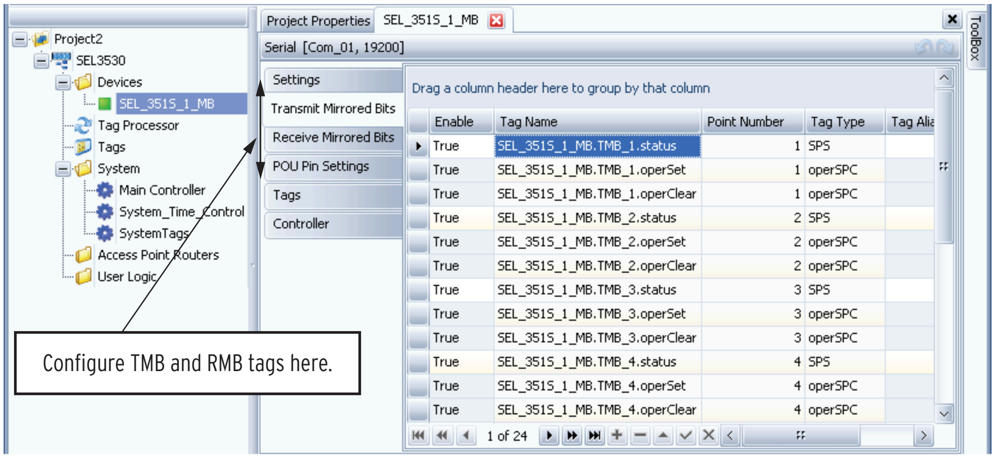


Figure 2.29 Configure MIRRORED BITS Tags

Data Configuration Parameters

Configure default values and other information for transmit and receive MIRRORED BITS. Descriptions of common fields will only be detailed in *Transmit MIRRORED BITS on page 95*.

Step 1. Click on the **Transmit MIRRORED BITS** tab.

Step 2. Change the names of the tags, as necessary.

Step 3. Add alias names, as necessary.

Step 4. Change other point-related information as needed.

Transmit MIRRORED BITS

Transmit MIRRORED BITS populate outgoing messages to IEDs. They have a default value and are set through mapping in the Tag Processor or through custom logic.

Table 2.22 TX MIRRORED BITS Parameters

Parameter	Description	Default
Enable	Set this flag True to enable processing of this point. Set this flag False to disable.	True
Tag Name	The default tag name contains the device name, and point type, and it is numbered 1–*n* points. You can change this name as necessary.	
Point Number	The point number of these MIRRORED BITS. Note that point numbers are duplicated in sets of three. This is because the attributes, operSet, operClear, and status, belong to the same point.	
Tag Type	See *Data Types on page 747* for more details.	Each tag has three attributes. Two are controls (operSPC type), and one is a status (SPS type).

Parameter	Description	Default
Tag Alias	Enter an optional descriptive tag name in this field and then reference this tag alias anywhere in the RTAC system in place of the actual tag name.	
Status Value	The initialized value at startup.	False
Comment	Optional user-entered comment field.	

Receive MIRRORED BITS are contained in messages that come from another IED.

Table 2.23 RX MIRRORED BITS Parameters

Parameter	Description	Default
Bit Number	The bit in the received MIRRORED BITS message corresponding with this point.	
Channel Fail State	The state to which the RTAC will drive the MIRRORED BITS in the event of channel loss.	True, False, or Previous (last known state)
Pickup Delay	The number of consecutive packets in which an incoming bit must be asserted before the associated receive MIRRORED BITS asserts.	
Dropout Delay	The number of consecutive packets in which an incoming bit must be deasserted before the associated receive MIRRORED BITS deassert.	

MIRRORED BITS Tag

Each MIRRORED BITS point has three main components:

1. **operSet.ctlVal**: A rising-edge change to operSet.ctlVal will set the MIRRORED BITS to a value of 1.
2. **operClear.ctlVal**: A rising-edge change to operClear.ctlVal will set the MIRRORED BITS to a value of 0.
3. **status**: The present state of the MIRRORED BITS.

A rising edge triggers the change, so the values of operSet.ctlVal and operClear.ctlVal must go low before they can trigger another change. For example:

T-0: SEL_351R_1_MB.TMB_1.operSet := TRUE

SEL_351R_1_MB.TMB_1.status becomes TRUE (1)

T+1: SEL_351R_1_MB.TMB_1.operSet := TRUE

No change because operSet is already high

T+2: SEL_351R_1_MB.TMB_1.operClear := TRUE

SEL_351R_1_MB.TMB_1.status becomes FALSE (0)

Figure 2.30 illustrates the relationship of operSet, operClear, and status in the MIRRORED BITS data structure.

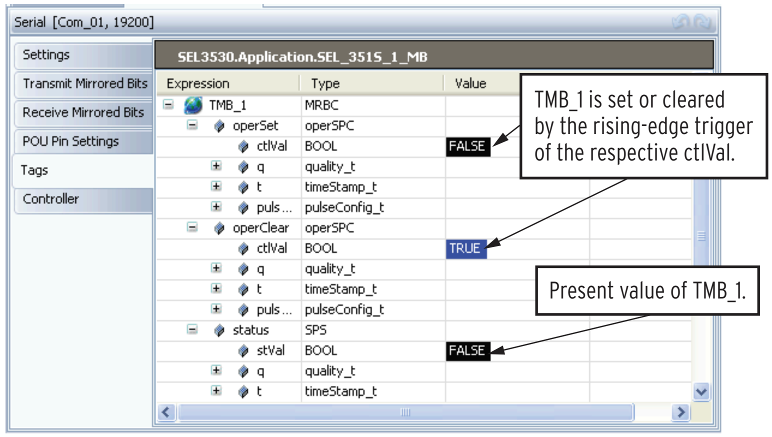


Figure 2.30 Online MIRRORED BITS

Example 2.4 Loopback Test With Controller

This is an example of forcing the POU pin Loop_Back_Without_Data while online to initiate a loopback test. Forcing the pin to TRUE initiates the loopback test. Forcing the pin back to FALSE ends the test.

Step 1. Download project to RTAC and stay online.

Step 2. Click on the **Controller** tab.

Step 3. Type **TRUE** in the prepared **Value** column for Loop_Back_Without_Data pin.

NOTE
The RTAC will only increment the Tx_Message_Count when a transmit bit changes state. It is possible then for the RTAC to transmit many messages without incrementing this counter.

Step 4. Press **<F6>** to force value.

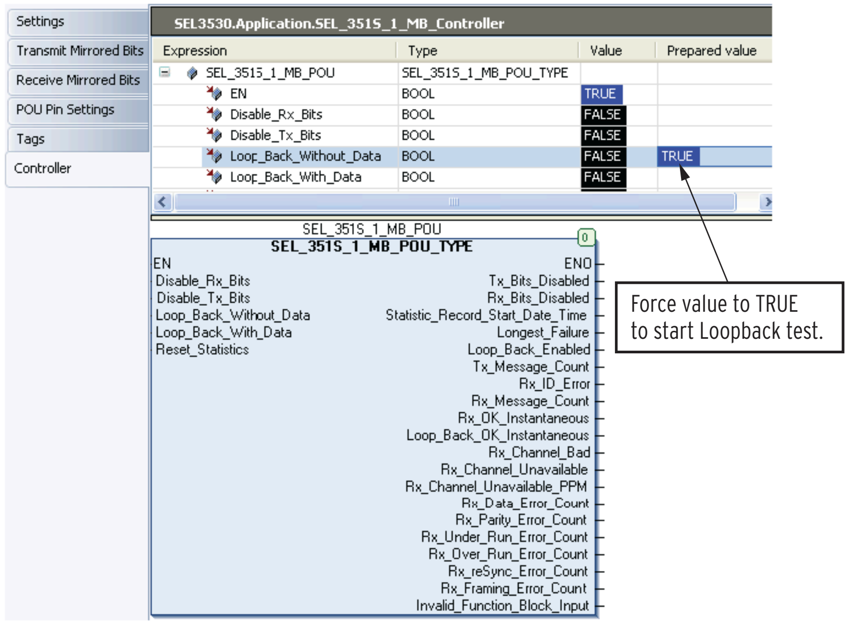
Loopback test begins.

Step 5. Click on **TRUE**, and a dialog box prompts for another forced value.

Step 6. Type **FALSE** in the dialog box, and click **Force**.

Step 7. The value of Loop_Back_Without_Data is False, and the test ends.

Step 8. Enter **<Shift+F6>** to unforce the value.



IEEE C37.118 Synchrophasors

Synchrophasors provide the ability to compare time-stamped power measurements from different devices and physical locations throughout a power system. A device that measures and reports synchrophasors is called a phasor measurement unit (PMU). IEEE C37.118 is the industry standard protocol that defines how to send and receive synchrophasor messages between PMUs and other devices such as the RTAC. A synchrophasor message includes magnitude, angle, and time stamp for each measured quantity. Synchrophasors can provide comparisons of simultaneous measurements from different locations with errors of less than a quarter of an electrical degree.

Configure the IEEE C37.118 client protocol on any of the RTAC serial or Ethernet ports to communicate with any device that supports an IEEE C37.118 server to gather synchrophasor values. You can configure as many as 100 synchrophasor client sessions on the RTAC at any given time. The RTAC also supports 100 synchrophasor server sessions.

Synchrophasor systems often employ phasor data concentrators (PDCs) to collect data from multiple PMUs, align the samples based on a common time stamp, and aggregate the samples into a concentrated data message. These time-aligned data sets can provide a coherent snapshot of wide-area measurements to an end application. A PDC can be a standalone device, supporting data receipt and transmission over a standardized protocol such as IEEE C37.118, or a function within an end-application. The RTAC, as a PDC, supports IEEE C37.118 client and server functionality and applies time alignment to all configured IEEE C37.118 clients. This ensures coherency of the phasor data presented to the SEL-RTAC logic engine as well as any configured IEEE C37.118 servers. The SEL-RTAC IEEE C37.118 client is also capable of reading concentrated data messages from other PDCs, see IEEE C37.118 Client Configuration.

This section describes the configuration and use of the IEEE C37.118 protocol client and server with ACSELERATOR RTAC. For specific information on the IEEE C37.118 protocol, see *IEEE C37.118.2-2011, IEEE Standard for Synchrophasor Data Transfer for Power Systems*.

For a detailed description of client and server concepts and other general information on protocol configuration, see *Overview on page 25*.

IEEE C37.118 System Configuration

Configure the RTAC for basic data concentration and time alignment, as a synchrophasor logic processor and controller, or as a combination of the two. The RTAC provides the option to route synchrophasor data directly between IEEE C37.118 clients or SEL Axion modules and IEEE C37.118 servers, thus bypassing the logic engine. Synchrophasor data channels from IEEE C37.118 clients and Axion modules can optionally be enabled for use as tags in the logic engine. *Figure 2.35* shows the available routing paths for synchrophasor data in the RTAC.

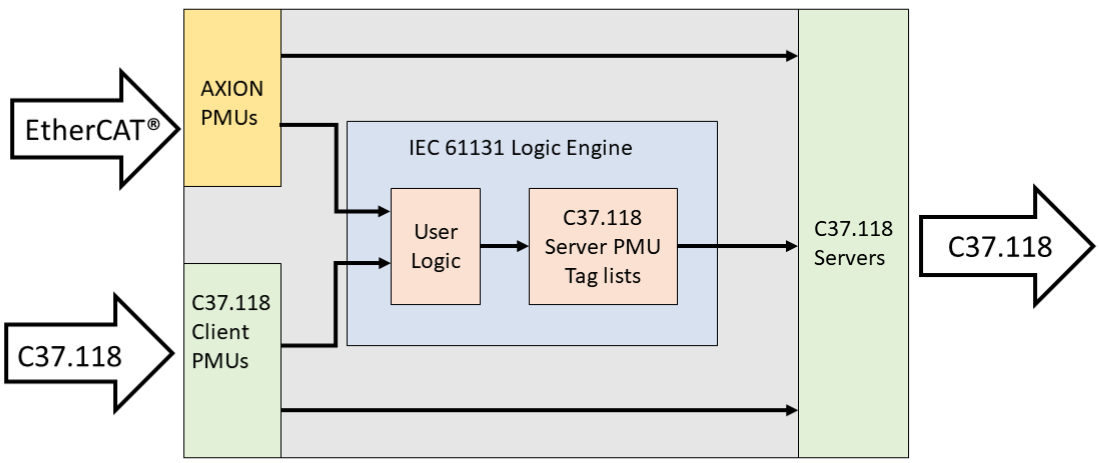


Figure 2.31 SEL RTAC Synchrophasor Data Routing Options

Note that application extensions, such as the Dynamic Disturbance Recorder (DDR) for synchrophasor data archiving, are dependent on the logic engine. Therefore, synchrophasor data channels must be enabled for use as logic engine tags before they can be applied to these extensions.

IEEE C37.118 Client Configuration

Configure an IEEE C37.118 client connection to communicate via serial or Ethernet to PMUs and relays that provide phasor data via the IEEE C37.118 protocol. The RTAC accepts phasor measurement data from these IEDs, makes the PMU data sets directly available to configured IEEE C37.118 servers, and optionally inserts the data into the logic engine.

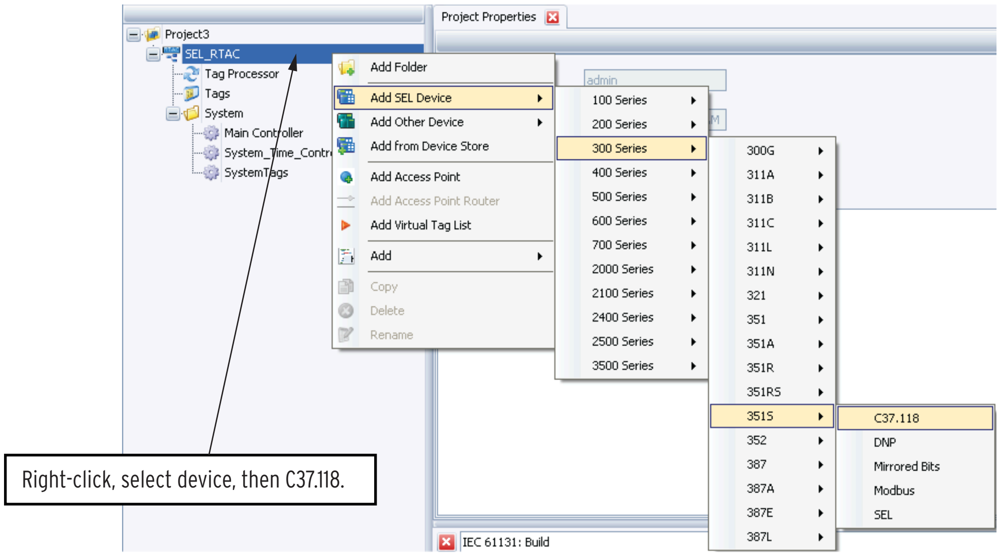


Figure 2.32 Insert IEEE C37.118 Client Device

Insert a non-SEL IEEE C37.118 device by selecting **Other** from the **Insert** ribbon. SEL devices that support IEEE C37.118 (including the SEL-421, SEL-451, SEL-751A, etc.) are preconfigured in ACSELERATOR RTAC. Insert those devices by selecting **SEL** from the **Insert** ribbon.

Give the device connection a unique name and select the connection type as shown in *Figure 2.1*. For Ethernet connections, the combination of the PMUs IP address and the PMUs port must be unique (i.e., two PMUs can have the same IP port number only if those two PMUs have different IP addresses).

By selecting the **Settings** tab, you see all configurable items for communications. Check the **Description** column for details on each configuration item. Move the slider bar or hover your mouse over a truncated description to see the entire text of a description. Type any necessary comments in the blank column to the far right.

Add IEEE C37.118 PMUs

Data messages transmitted directly from a PMU generally incorporate a single PMU data set within the data message. Data messages transmitted by a PDC may contain one or more PMU data sets for a given time stamp. Each SEL RTAC IEEE C37.118 client can accept data messages that represent between 1 and 100 individual PMU data sets. Preconfigured SEL IEEE C37.118 clients are preloaded with a single predefined PMU data set when they are added to the project. Non-SEL IEEE C37.118 devices added to the project must be assigned one or more empty PMU data sets that must be configured appropriately. To add a PMU to an IEEE C37.118 client, perform the following:

Step 1. Click on the **PMUs** tab.

Step 2. Click + to add PMUs.

Step 3. Specify the number of PMUs to add to the IEEE C37.118 client and click **Add**.

Step 4. Specify the **Station Name**. The first 16 characters must exactly match the Station Name setting programmed in the IED.

Step 5. Specify the **Pmu Id**. This must match the PMU ID setting programmed in the IED.

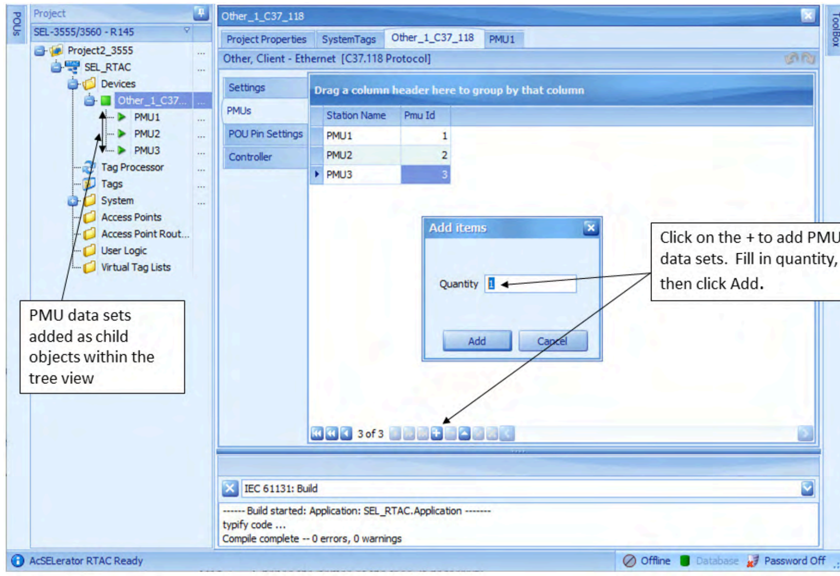


Figure 2.33 Add IEEE C37.118 PMUs

Configure PMU Datasets

Each PMU data set added to an IEEE C37.118 client contains five tabs (Status, Frequency, Phasors, Analogs, and Digitals) that can be configured to represent the full set of measurements and status from a PMU. All channels represented in these tabs use an identical set of parameters.

Table 2.24 Common Settings for PMU Data Set Tags

Parameter	Description
Enable	Enables the channel for direct access by IEEE C37.118 servers
Channel Name	Name as defined in the IED (as many as 16 characters)
Enable Tag	Enable the channel for use in the logic engine (requires Enable = TRUE)
Tag Name	Logic engine tag name for tag enabled channels
Tag Type	IEC 61131 data structures used for tag enabled channels

Channels can be independently enabled and enabled for logic engine use. All enabled channels are directly available to any configured IEEE C37.118 servers. This allows the RTAC to perform basic PDC time alignment and data aggregation without requiring the logic engine for tag mapping between client and server. See *IEEE C37.118 Client Configuration on page 99* for additional information on direct mapping of IEEE C37.118 client PMU data sets to IEEE C37.118 servers.

On each of the Phasors, Analogs, and Digitals tabs, add channels as needed to correspond with the IED data set. Again, SEL IEEE C37.118 clients come preloaded with a single PMU data set with predefined Phasor, Analog, and Digital data channels.

Using IEEE C37.118 Client POU Controller Outputs for Troubleshooting

This section details a subset of useful POU controller output pins. For advanced troubleshooting, set the **Visible** field to True for all tags on the IEEE C37.118 client POU Pin Settings page and observe the **Controller** tab while online with the project.

Unresolved_Tag_Reference_Error and Unresolved_Tag_Reference_Count

These outputs indicate the presence and number of configured data channels that contain a Channel Name setting that does not exactly match any channel name found in the IEEE C37.118 configuration frame received from the connected PMU. The count encompasses all channels amongst all configured PMUs on the IEEE C37.118 client.

Missed_Data_Count

This output is the running count of data messages that have been missed. This number is updated by comparing the most recently received synchrophasor time stamp against the previously received time stamp and comparing the difference to the data rate interval (i.e., 1 / Data_Rate), where Data_Rate is configured on the System Settings page.

Missed data may be a result of network congestion that leads to complete packet loss. It can also result from network latency causing the packet to be received at the RTAC after the synchrophasor time alignment wait period has elapsed, in which case it will be thrown away. You can adjust the duration of the wait period via the Waiting_Period setting on the System Tags page.

Considerations for Logic Engine Enabled Channels

Task Cycle

When channels are optionally enabled for use in the logic engine, consideration needs to be given to the associated Cycle Time setting on the Main Controller settings page. SEL recommends setting Cycle Time to approximately 1 / Data_rate, where Data_Rate is the C37.118 Message Rate setting on the System Tags page.

Mapping From IEEE C37.118 Protocol to Logic Engine Tags

Corresponding logic engine tags will be created for channels that are enabled for use in the logic engine. The IEC tag data structures (e.g., MV, CMV, SPS) will be populated from the IEEE C37.118 data frame.

Status Mapping

Common tag sub-structures, such as timestamp_t and quality_t, will be synchronized across all tags associated with a given PMU data set and mapped as described in tables *Table 2.25* and *Table 2.26*.

Table 2.25 IEEE C37.118 Mapping to timeStamp_t

IEEE C37.118 Field	Manipulations	To Logic Engine Tag
SOC	Convert to local time	.value.dateTime
FRACSEC value	Round via Time Variance setting*	.value.uSec
Fault – Clock Failure		quality.accuracy = IEC_Unspecified
Within 10 s of UTC		quality.accuracy = IEC_Unspecified
Within 1 s of UTC		quality.accuracy = IEC_Unspecified
Within 100 ms of UTC		quality.accuracy = IEC_Unspecified
Within 10 ms of UTC		.quality.accuracy = IEC_T0
Within 1 ms of UTC		.quality.accuracy = IEC_T1
Within 100 µs of UTC		quality.accuracy = IEC_T2
Within 10 µs of UTC		quality.accuracy = IEC_T3
Within 1 µs of UTC		quality.accuracy = IEC_T5
Within 100 ns of UTC		quality.accuracy = IEC_T5
Within 10 ns of UTC		.quality.accuracy = IEC_T5
Within 1 ns of UTC		quality.accuracy = IEC_T5
Clock Locked		quality.accuracy = IEC_T5
STAT bit 13: PMU No Sync = TRUE		.quality.clock_not_synchronized .quality.clock_failure

The Time Variance setting is an advanced setting on the **Settings** tab of each IEEE C37.118 client. It defines the allowable limit by which an incoming IEEE C37.118 time stamp can be rounded to conform to the expected IEEE C37.118 time stamp.

Table 2.26 IEEE C37.118 Mapping to quality_t

IEEE C37.118 Field	To Logic Engine Tag
STAT bit 15 (Data Invalid) = FALSE STAT bit 14 (PMU Error) = FALSE STAT bit 12 (Sort by arrival) = FALSE	.validity = IEC_GOOD
STAT bit 15 (Data Invalid) = TRUE STAT bit 14 (PMU Error) = TRUE	.validity = IEC_INVALID
STAT bit 12 (Sort by arrival) = TRUE STAT bit 15 (Data Invalid) = FALSE STAT bit 14 (PMU Error) = FALSE	.validity = IEC_QUESTIONABLE

Measurement Mapping

Phasor, analog, and Boolean quantities within the PMU data set that are enabled for use in the logic engine will be mapped to respective CMV, MV, and SPS structures as described in *Table 2.27*.

Table 2.27 IEEE C37.118 Measurement Mapping to Logic Engine Tags

IEEE C37.118 Field	Manipulations	To Logic Engine Tag
Phasor	Convert to polar coordinates with angle in degrees	CMV.instCVal.mag CMV.cVal.mag CMV.instCVal.ang CMV.cVal.ang
Analog		MV.instMag MV.mag
Boolean		SPS.stVal

Starting with RTAC firmware version R145-V0, the range and dead-band settings associated with IEEE C37.118 phasors and analogs are no longer included when those data are mapped to CMV and MV tags, respectively. This was done to promote the streaming nature of IEEE C37.118 data. If desired, the range and dead-band settings can be applied to the data in user logic. See the MVRangeAndDeadbandCheck and CMVRangeAndDeadbandCheck functions in *Appendix B: IEC 61131-3 Programming Reference* for more information.

Required System Settings and IEEE C37.118 Client Time Alignment

All PMUs communicating with IEEE C37.118 clients on the RTAC must be configured with the same Nominal_Frequency and Data_Rate as configured in the RTAC system settings.

The RTAC applies time alignment to all IEEE C37.118 clients. The system setting Waiting_ Period specifies how long the RTAC will wait to receive all IEEE C37.118 data messages for a given time stamp amongst all configured IEEE C37.118 clients. The internal counter, to which the Waiting_Period is applied, begins upon the receipt of the first data message with a new time stamp. This relative form of time alignment eliminates the need for RTAC synchronization to a GPS time source. Once the set is received, or the Waiting_Period duration has elapsed, the received samples will be made available to the rest of the system.

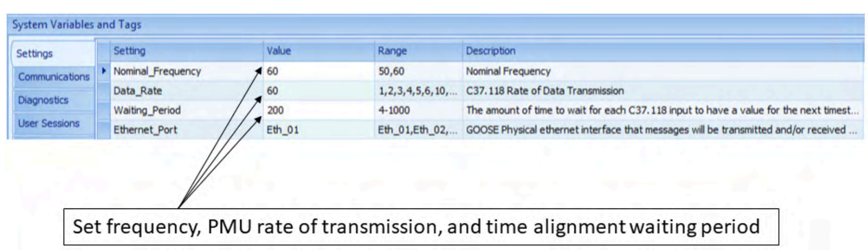


Figure 2.34 System Tags

Ensure configuration of all PMUs to the same frequency and data rate as the RTAC client. The RTAC accepts as valid only PMU messages that match the RTAC settings.

IEEE C37.118 Axion PMU

Each SEL Axion module that provides IEEE C37.118.1-2011-compliant synchrophasor measurements represents an independent PMU with frequency, voltage, and current measurements. The following Axion modules provide such synchrophasor measurements:

- SEL-2245-4 AC Metering Module
- SEL-2245-411 Standard Current and Low-Voltage (LEA) Monitoring Module
- SEL-2245-22 DC Analog Input Extended Range Module
- SEL-2245-221 Low-Voltage (LEA) Monitoring Module
- SEL-2245-42 AC Protection Module

At a minimum, each Axion PMU includes its own PMU ID, station name, frequency measurement, and rate-of-change-of-frequency measurement. Additionally, each Axion PMU provides an optional global PMU ID configuration and a large station name (greater than 16 characters) to support Configuration 3 (CFG3) requests. The Axion automatically synchronizes all modules and their respective synchrophasor measurements to its IRIG-B input time source and requires no additional time alignment from the IEEE C37.118 server.

Settings

An Axion PMU includes the settings shown in *Table 2.28*.

NOTE
Only ac analog input modules that have current input channels (such as the SEL-2245-42 AC Protection module) support the Current Phasor Data Set. Additionally, the SEL-2245-42 AC Protection Module does not support IN or VS.

Table 2.28 Axion PMU Settings

Setting	Range	Description
Station Name	1–255 characters	The station name of the PMU
Current Phasor Data Set	All, Phase, Positive Sequence, None	Enable current phasor channels for the PMU data set
Voltage Phasor Data Set	All, Phase, Positive Sequence, None	Enable voltage phasor channels for the PMU data set
PMU ID	1–65534	The identifier of the PMU
Global PMU ID	$0–2^{128-1}$	The global identifier (Advanced Setting) of the PMU

Table 2.29 shows the current and voltage channels that the Axion PMU includes for each data set configuration.

Table 2.29 Axion Voltage and Current Settings

Voltage/Current Phasor Data Set	PMU Current Channels	Wye PMU Voltage Channels	Delta PMU Voltage Channel
All	IA, IB, IC, IN, I1, I2, and I0	VA, VB, VC, VS, V1, V2, and V0	VAB, VBC, VCA, V1, and V2
Phase	IA, IB, IC, and IN	VA, VB, VC, and VS	VAB, VBC, and VCA
Positive Sequence	I1	V1	V1
None	No current channels	No voltage channels	No voltage channels

Table 2.30 shows the settings for Axion PMU message rate and message class. All Axion PMUs use the same synchrophasor message rate and message class.

Table 2.30 Axion PMU Rate

Setting	Range	Description
Synchrophasor Message Rate[a]	1, 2, 4, 5, 10, 12, 15, 20, 30, or 60 (Nominal_Frequence = 60) 1, 2, 5, 10, 25, or 50 (Nominal_Frequency = 50)	The synchrophasor messaging rate (messages/second)
Synchrophasor Performance Class[b]	P or M	The synchrophasor performance class

[a]The Synchrophasor Message Rate must match the Data Rate of the IEEE C37.118 server.

[b]IEEE C37.118.1-2011 defines two performance classes: P (protection) and M (meter). P-class measurements use less filtering, resulting in faster response times and lower message latency. M-class measurements use more filtering, resulting in slower response times and higher message latency.

Configuration

Configuring an Axion PMU does not require channel mapping or naming. After adding an ac analog input module (see *EtherCAT on page 151*), configure its PMU settings in the **Settings** tab under the PMU grouping, as shown in *Figure 2.35*.

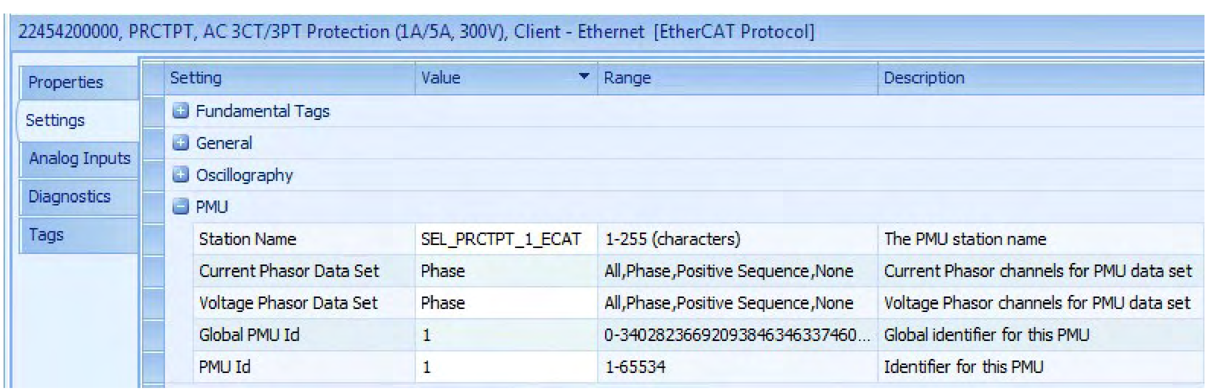


Figure 2.35 Axion PMU Configuration

Add the Axion PMU to a IEEE C37.118 server by selecting the name of the module in the PMU Reference column in the **PMUs** tab of the server, as shown in *Figure 2.36*.

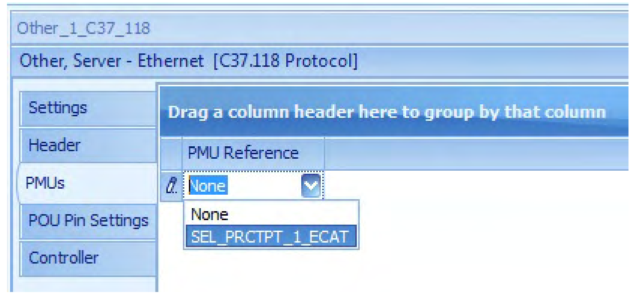


Figure 2.36 Add Axion PMU

Configure the Synchrophasor Messaging Rate and Synchrophasor Performance Class in the **Settings** tab in the EtherCAT I/O Network page. ACSELERATOR RTAC will indicate an error if the Synchrophasor Messaging Rate does not match the data rate of the IEEE C37.118 server.

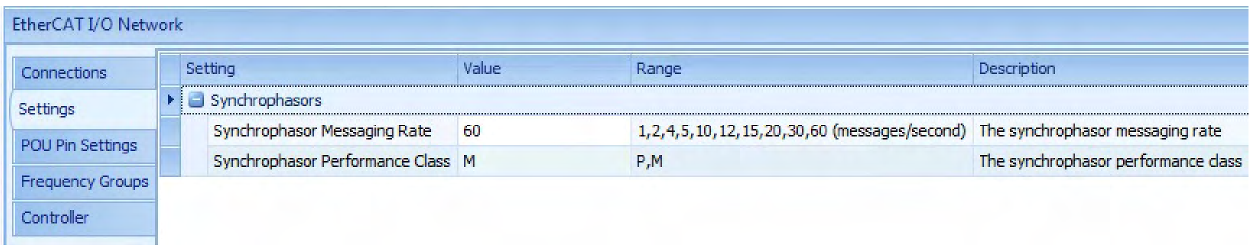


Figure 2.37 Axion PMU Message Rate Settings

IEEE C37.118 Server PMU

A Server PMU is a tag list that defines a custom PMU data set within the RTAC logic engine. Data values, time stamps, and quality status associated with the data set must be specified by the user. This allows for the creation of a virtual PMU within the RTAC logic engine. An IEEE C37.118 server can be configured to transmit any PMU data set defined in the RTAC project, including IEEE C37.118 server PMUs, Axion PMUs *IEEE C37.118 Axion PMU on page 105*, and IEEE C37.118 client PMUs.

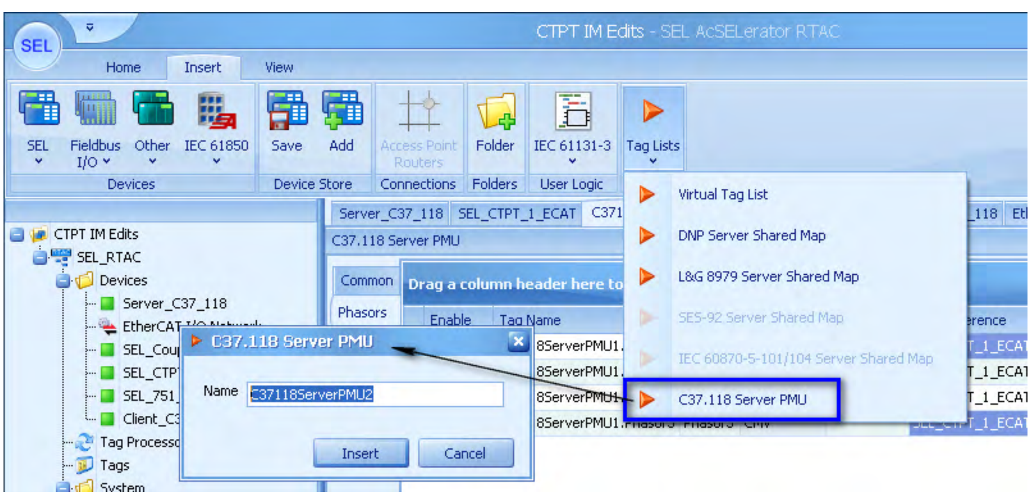


Figure 2.38 Insert IEEE C37.118 Server PMU

Add a IEEE C37.118 server PMU by selecting **Tag Lists** from the **Insert** menu. Give the PMU a unique name.

Server PMU Time Stamps

The IEEE C37.118 server PMU data set is associated with a single time stamp that is derived from either a user-specified time stamp or the system time. The original time stamp is adjusted to conform to the IEEE C37.118-mandated sequence of sub-second values which begin at zero for the top of the second and are evenly divided based on the synchrophasor reporting rate.

Tag Referencing

The IEEE C37.118 server PMU tag list allows the optional specification of a tag reference for all measurements and statuses, as well as a required tag reference for the Time Reference and Quality Reference settings on the **Settings** tab. Creation of a data assignment through use of the tag referencing differs from creation of a data assignment made through the tag processor or user logic. Instead of copying the value of the tag from the source tag to the destination tag, the software creates a direct link to the source tag in the memory of the RTAC. Because of this, working with reference tags within RTAC settings differs from use of the tag processor or user logic in the following ways:

- No data manipulation (such as scaling, offsetting, or logical math) can be performed on the data prior to assignment.
- Because both the source and destination tag point to the same memory location, a change to the destination tag changes the source tag (the tag reference), and a change to the source tag changes the destination tag.

⚠IMPORTANT

Only tags from communication protocol clients and tag lists are considered valid for tag referencing. This excludes variables declared in custom POUs or GVLs.

Server PMU Configuration

The server PMU contains a tab for general settings as well as tabs for defining measurements and additional status information.

Settings Tab Configuration

The **Settings** tab allows for the definition of global identifiers for the PMU data set as well as settings that can affect how an IEEE C37.118 server handles the data set.

Time Reference

Assign a timestamp_t tag or any tag that contains a timestamp_t structure. This tag is used as the time reference for the PMU data set. This time reference will supersede the timestamp_t substructure on each tag in the data set.

Time Variance (Advanced Setting)

This is the maximum allowable amount that the time reference can be rounded in order to conform to an IEEE C37.118 sequence of sub-second values for the given Data_Rate System Tags setting. If the time reference time stamp deviates by more than the specified time variance, the time stamp assigned to the IEEE C37.118 server PMU will not update. Any IEEE C37.118 servers configured in the RTAC project will observe this as a missing sample. *Table 2.31* shows an example sequence of Time Reference values and the corresponding rounded sub-second time stamps that are assigned to the IEEE C37.118 server PMU data set.

Table 2.31 PMU Time Stamp Rounding Example for System Tags

Time Reference Fraction of Second (in microseconds)	IEEE C37.118 Server PMU Fraction of Second (in microseconds)
1000	0
15000	16667
34000	33333
48500	50000
69000	50000 (No change. Expected 66667 ± 2000)
84000	83333
110000	100000

Streaming (Advanced Setting)

When streaming mode is enabled, the Time Reference and Time Variance settings are disabled. Synchrophasor time stamps for the PMU data set will be derived from the system time. Note that when streaming mode is enabled, the outgoing IEEE C37.118 messages will indicate fault time quality for the entire message and will indicate "PMU Not Synchronized and Unlocked for over 1000 s" for the PMU data set within the message.

Quality Reference

Assign a quality_t tag or any tag that contains a quality_t structure. This tag is used as the quality reference for the PMU data set. This quality reference will supersede the quality_t substructure on each tag in the data set.

Station Name

Assign a station name (1–255 characters) to the PMU data set. An IEEE C37.118 server that includes this data set will apply the station name to the associated STN field of any transmitted configuration frames.

PMU ID

Assign a numerical ID between 1 and 65534 to the PMU data set. An IEEE C37.118 server that includes this data set will apply the ID to the associated IDCODE field of any transmitted configuration frames.

Global PMU ID (Advanced Setting)

Optionally assign a global ID code for the PMU data set. This can be any whole number representable by a 128-bit unsigned integer. An IEEE C37.118 server that includes this data set will apply the global ID to the associated G_PMU_ID field of the CFG3 message, as defined by IEEE C37.118.2-2011. This setting provides support for unique identification in synchrophasor systems with more than 65534 PMUs.

PMU Latitude, Longitude, and Elevation

Optionally assign global coordinates and elevation to the IEEE C37.118 server PMU data set. An IEEE C37.118 server that includes this data set will apply the coordinates and elevation to the corresponding fields in the CFG3 message.

Measurement and Status Configuration

Configure the **Frequency**, **Status**, **Phasors**, **Analogs**, and **Digitals** tabs to fully define the PMU data set that will be made available to one or more IEEE C37.118 servers. Each tab contains a **Tag Reference** column that can be used to map directly from the memory space of an existing tag. If this field is left blank, the **Tag Name** column is enabled. This column specifies the name of the newly created tag which can be populated through user logic in the tag processor or a dedicated POU. The **Channel Name** field specifies the data channel name, the first 16 characters of which will appear in the CHNAM field of any outgoing IEEE C37.118 configuration frames transmitted by an IEEE C37.118 server that includes this data set.

Frequency Tab Configuration

The **Frequency** tab allows for the user specification of the frequency and rate-of-change-of-frequency (ROCOF, denoted as DF_DT in settings) quantities to be associated with the IEEE C37.118 server PMU. The inclusion of these quantities is required by the IEEE C37.118 standard. If either tag is disabled, an IEEE C37.118 server that includes this data set will set the invalid data flag (IEEE C37.118 Data Frame STAT bit 15 = TRUE) for the given PMU data set in the outgoing data message.

Status Tab Configuration

The **Status** tab allows for user definition of additional information that is implicit to the IEEE C37.118 protocol.

- **PMTRIG** (PMU trigger) is a Boolean quantity, per PMU data set, in each IEEE C37.118 data message, which allows the PMU to inform an upstream device that a predefined condition has been met. This bit is typically allowed to be driven by user logic in most PMUs and is often used to initiate triggered archiving in upstream PDCs.
- **TREA** (trigger reason) is an integer value that is meant to be set in conjunction with an assertion of the PMTRIG Boolean quantity and is intended to provide additional information about the PMU trigger. The IEEE C37.118 standard specifies predefined interpretations of the trigger reason, as shown in *Table 2.32*.

Table 2.32 Trigger Reason

Bit	Reason
0	Manual
1	Magnitude low
2	Magnitude high
3	Phase angle diff
4	Frequency high or low

Bit	Reason
5	Df/dt high
6	Reserved
7	Digital
8–15	Available for user definition

Add IEEE C37.118 Data

Step 1. Click on a tag type to add and configure tags.

Step 2. Click + to add tags.

Step 3. Change Channel Names of tags if necessary.

Step 4. Assign a tag reference or populate the Tag Name tag in user logic.

Add as many as 3200 phasor tags, 3200 analog tags, and 4000 digital tags to the IEEE C37.118 server PMU data set by configuring the respective **Phasors**, **Analogs**, and **Digitals** tabs. The first 16 characters of each Channel Name setting, for each channel, will be used as the associated channel name (CHNAM field) within an outgoing IEEE C37.118 configuration frame that includes this data set. For phasor tags, the Phasor Type will be applied to the phasor type byte within the associated PHUNIT field of an outgoing IEEE C37.118 configuration frame.

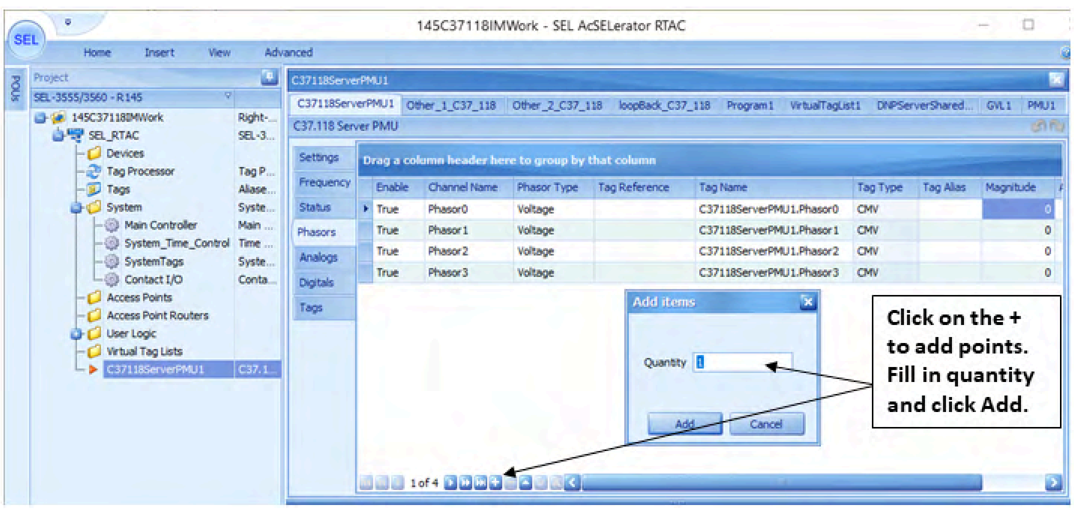


Figure 2.39 Add IEEE C37.118 Tags

Mapping From Logic Engine Tags to IEEE C37.118 Protocol

An IEEE C37.118 server PMU tag list not only allows the specification of a custom PMU data set, but also converts the data, time stamp, and status indicators into an IEEE C37.118-formatted message for consumption by the IEEE C37.118 server. *Table 2.33*, *Table 2.34*, and *Table 2.35* detail the mapping from IEEE C37.118 server PMU tags to the IEEE C37.118 message format.

Table 2.33 timestamp_t Mapping to IEEE C37.118

From Logic Engine Tag	Manipulations	To C37.118 Field
.value.dateTime	Convert to UTC	SOC
.value.uSec	Round subsecond value based on Time Variance setting and System Tags Data_Rate setting.	FRACSEC value
		Unlocked Time (Bits 4–5 of STAT field)
.quality.clock_not_synchronized OR .quality.clock_failure OR .quality.accuracy = IEC_Unspecified		If TRUE: Unlocked for 1000 s or more If FALSE: Locked or unlocked for less than 10 s
		FRACSEC Time Quality
.quality.clock_not_synchronized OR .quality.clock_failure OR .quality.accuracy = IEC_Unspecified		If TRUE: Fault—Clock Failure If FALSE: defer to .quality.accuracy
.quality.accuracy = IEC_T0		Within 10 ms of UTC
.quality.accuracy = IEC_T1		Within 1 ms of UTC
.quality.accuracy = IEC_T2		Within 100 µs of UTC
.quality.accuracy = IEC_T3		Within 100 µs of UTC
.quality.accuracy = IEC_T4		Within 10 µs of UTC
.quality.accuracy = IEC_T5		Clock Locked
		PMU Time Quality (Bits 6–8 of STAT field)
.quality.clock_not_synchronized OR .quality.clock_failure OR .quality.accuracy = IEC_Unspecified		If TRUE: Estimated time error >10 ms or time error unknown If FALSE: defer to .quality.accuracy
.quality.accuracy = IEC_T0		Estimated time error <10 ms
.quality.accuracy = IEC_T1		Estimated time error <1 ms
quality.accuracy = IEC_T2		Estimated time error <100 µs
.quality.accuracy = IEC_T3		Estimated time error <100 µs
.quality.accuracy = IEC_T4		Estimated time error <10 µs
.quality.accuracy = IEC_T5		Estimated time error <1 µs

Table 2.34 quality_t Mapping to IEEE C37.118

From Logic Engine Tag	To C37.118 Field
.test OR .validity = IEC_Invalid OR .validity = IEC_Questionable	STAT bit 15: Data Invalid
.validity = IEC_Invalid	STAT bit 14: PMU Error

Table 2.35 Data Tags Mapping to IEEE C37.118

From Logic Engine Tag	Manipulations	To C37.118 Field
CMV.instCVal.mag CMV.cVal.mag	Convert to rectangular coordinates	Phasor
MV.mag		Analog
SPS.stVal		Digital

IEEE C37.118 Server

Add an IEEE C37.118 server to transmit PMU data sets from IEEE C37.118 client PMUs, IEEE C37.118 server PMUs, or Axion PMUs. The IEEE C37.118 server applies time alignment to all PMU data sets added to the **PMUs** tab. See *Required System Settings and IEEE C37.118 Client Time Alignment on page 104* for additional information on synchrophasor time alignment in the RTAC.

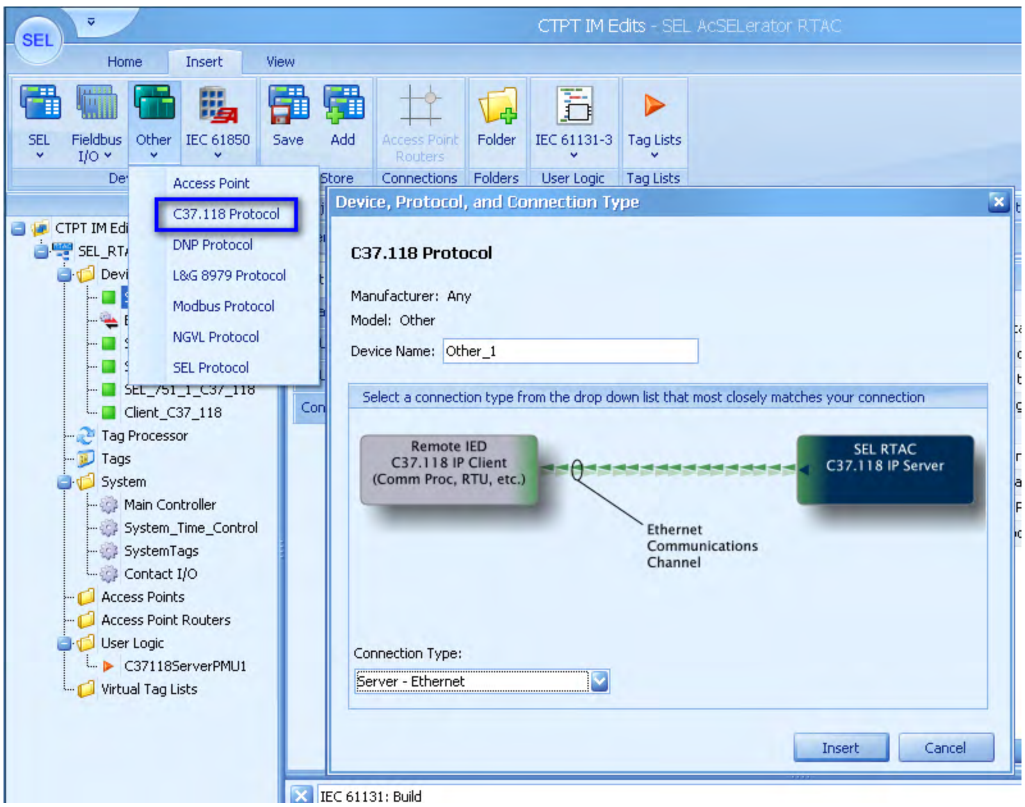
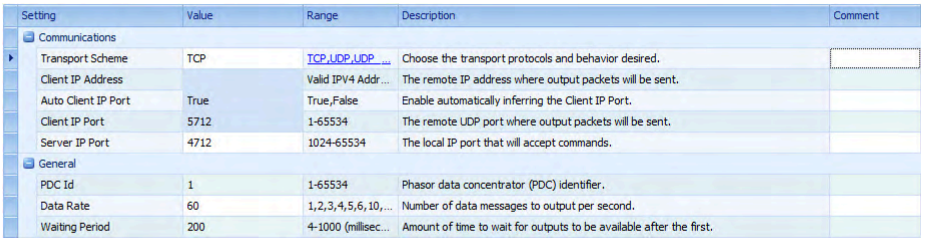


Figure 2.40 Insert IEEE C37.118 Server Device

Insert a non-SEL C37.118 device by selecting **Other** from the **Insert** ribbon. Give the device a unique name and select **Server – Ethernet** under **Connection Type**.

By selecting the **Settings** tab, you see all configurable items for communications. Check the **Description** column for details on each configuration item. Move the scroll bar or hover your mouse over a truncated description to see the entire text of a description. Type any necessary comments in the blank column to the far right.



Setting	Value	Range	Description	Comment
Communications				
Transport Scheme	TCP	TCP,UDP,UDP ...	Choose the transport protocols and behavior desired.	
Client IP Address		Valid IPV4 Addr...	The remote IP address where output packets will be sent.	
Auto Client IP Port	True	True,False	Enable automatically inferring the Client IP Port.	
Client IP Port	5712	1-65534	The remote UDP port where output packets will be sent.	
Server IP Port	4712	1024-65534	The local IP port that will accept commands.	
General				
PDC Id	1	1-65534	Phasor data concentrator (PDC) identifier.	
Data Rate	60	1,2,3,4,5,6,10,...	Number of data messages to output per second.	
Waiting Period	200	4-1000 (millisec...	Amount of time to wait for outputs to be available after the first.	

Figure 2.41 IEEE C37.118 Server Device Settings

PDC ID

Assign the global ID for transmitted messages. This will be applied to the IDCODE field in outgoing IEEE C37.118 data messages.

Data Rate

Data rate is the number of data messages transmitted per second. The IEEE C37.118 server does not require that the data rates associated with the incoming PMUs match the Data Rate setting (except for Axion PMUs); however, the incoming data rates and the IEEE C37.118 server Data Rate value must be divisible into each other. For example, an IEEE C37.118 server can use a Data Rate value of 30 messages per second while including PMUs that use data rates of 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60, 120, or 240 messages per second. An IEEE C37.118 server cannot, for example, use a Data Rate value of 20 messages per second while including PMUs that use data rates of 3, 6, 12, 15 or 30.

Data rate conversions from higher to lower are accomplished with simple decimation (no filtering applied).

Data rate conversions from low to high are accomplished by the insertion of zero-filled data samples and invalid quality indicators in between valid samples.

The data rate associated with Axion PMUs is required to match the IEEE C37.118 server Data Rate.

Waiting Period

The Waiting Period setting defines the length of time after receiving the first value that the server will wait before determining that data are missing and declaring the message invalid. When this occurs, the value is set to zero and the status of the message is set to invalid in the corresponding IEEE C37.118 server PMU. Additionally, the IEEE C37.118 Controller POU **Missing_Data_Error** pin will assert and the **Missing_Data_Count** will increment.

Communications

Transport Scheme

Transport Scheme configures transport protocols and behavior.

TCP

All messages are sent over a TCP connection. TCP provides a more robust communication path and can recover from some network problems, but adds additional latency.

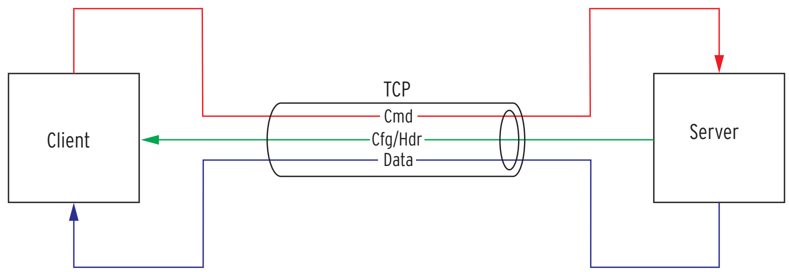


Figure 2.42 TCP Connection

UDP

UDP communicates exclusively over UDP including commands, data, headers, and configuration messages. In this scheme, the client must know the server address and port number. The disadvantage of UDP is that server-to-client communication receives no confirmation, making troubleshooting errors in communication problematic. Because there is no retransmission of data in the event of an error, packets can be permanently lost. In this transport scheme, set **Auto Client IP Port** to **True** to automatically infer the Client IP Port.

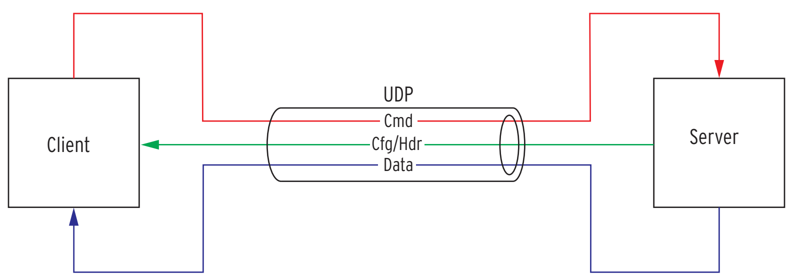


Figure 2.43 UDP Connection

UDP_S

Configuration and data messages are sent from the server to a UDP port on the client machine. No commands are required or accepted by the server. Configuration messages are sent when the IEEE C37.118 server is enabled and once per minute after that.

UDP_S provides more secure connections because no incoming connections and no additional firewall exceptions are required.

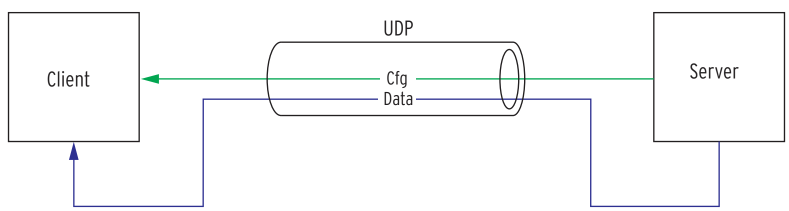


Figure 2.44 UDP_S Connection

UDP_T

Command, configuration, and header messages are transmitted and received over a TCP connection, while the data are transmitted to a separate UDP port on the client machine. This allows the command and configuration messages to have the robust transport of TCP while reducing latency on the data messages through UDP. The TCP connection must be maintained for continued transmission of the data messages.

NOTE
A large Time Variance increases the probability of successfully aggregating the tag, but introduces data to time stamp coherency errors.

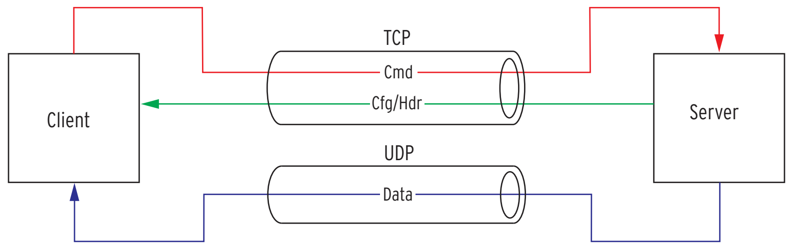


Figure 2.45 UDP_T Connection

UDP_U

Only command messages are transmitted over a TCP connection. The configuration, header, and data messages are transmitted over UDP to the client machine. The TCP connection must be maintained for continued transmission of the data messages.

This is useful in certain firewall situations. Each port is largely unidirectional; the TCP connection is always command messages from the client to the server, and the configuration and data messages are always going to the UDP port on the client.

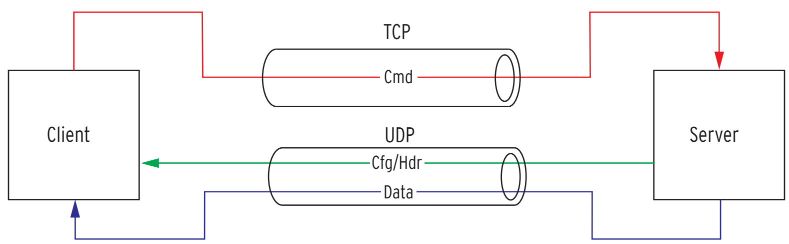


Figure 2.46 UDP_U Connection

Client IP Address

The remote IP address where output packets will be sent.

Auto Client IP Port

Automatically infer the Client IP Port. Only available in Transport Scheme UDP.

Client IP Port

The remote UDP port where output packets will be sent.

Server IP Port

Local TCP port that will accept connections.

Enable Multicast

Only available in Transport Scheme UDP_S.

Multicast Group IP Address

The multicast group where output packets are sent.

Ethernet Port

Specify which physical port is used to multicast IEEE C37.118 messages.

Header Tab

The **Header** tab is where the human-readable header is written.

PMU Tab

The **PMU** tab allows PMU data sets to be added to the given IEEE C37.118 server.

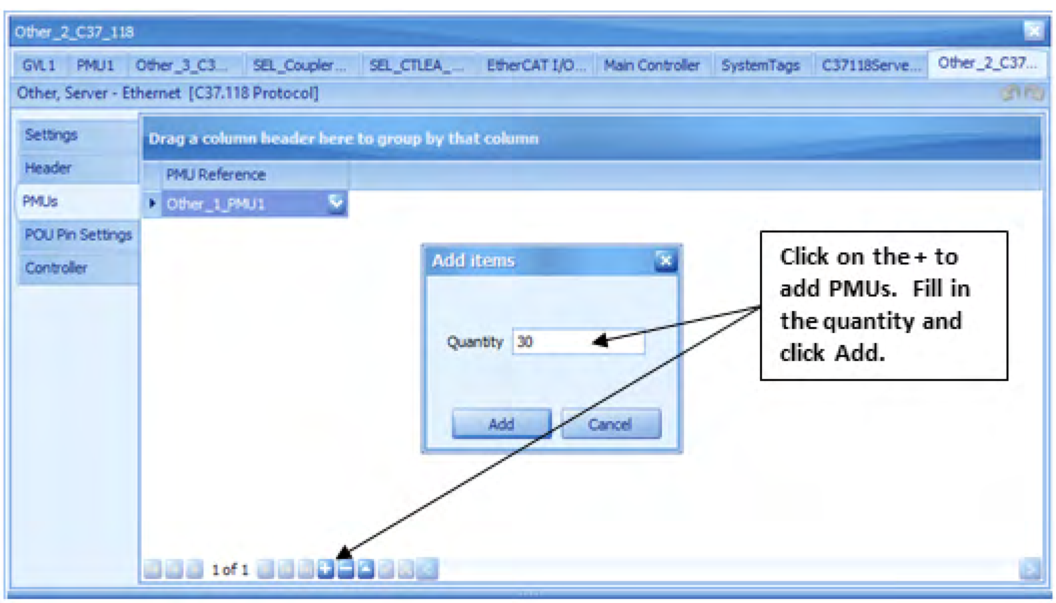


Figure 2.47 Server PMU Reference

Specify a PMU data set from the **PMU Reference** drop-down list. Note that PMU data sets from all configured IEEE C37.118 clients, Axion modules, and IEEE C37.118 server PMU tag lists will be available for inclusion in the IEEE C37.118 server configuration.

Note that the IEEE C37.118 server requires all referenced PMU datasets to use unique station names. Review *IEEE C37.118 Client Configuration on page 99*, *IEEE C37.118 Axion PMU on page 105*, and *IEEE C37.118 Server PMU on page 107* in this manual to identify the location of the associated Station Name settings.

Using IEEE C37.118 Server POU Controller Outputs for Troubleshooting

This section details a subset of useful POU controller output pins. For advanced troubleshooting, set the **Visible** field to TRUE for all tags on the IEEE C37.118 server POU Pin Settings page and observe the **Controller** tab while online with the project.

Missing_Data_Error and Missing_Data_Count

These outputs indicate the presence and quantity of missing data amongst the PMU data sets assigned to the IEEE C37.118 server. The most common cause of this condition is nonsynchronized time references amongst the PMU data sets. The IEEE C37.118 server applies time alignment to the included PMU data sets and will block data when detecting time references that differ more than the **Waiting Period** setting.

IEC 61850

Features

The RTAC uses Ethernet and IEC 61850 to support the following features.

MMS Client

Update RTAC tags with data from MMS devices by either polling for DataSets or by receiving reports. You can configure the RTAC for as many as 32 DataSets per IED, with as many as 16 used for polling and as many as 16 used for reporting. Presently, there is no defined limit on the maximum number of MMS devices with which the RTAC can communicate. Reporting can be a mixture of buffered and unbuffered reports.

In firmware R144 and later, the entryID received with buffered reports is saved to nonvolatile memory every 3 seconds. This maintains a record of which reports in a buffered report have already been processed in case the RTAC loses power, preventing the chance of processing duplicate events from IEDs when power is restored. When a project is downloaded to the RTAC, all entryID values are deleted and the RTAC will process all entryID values from the IED.

NOTE
The SEL-3505 currently does not have an IEC 61850 MMS client option.

MMS Server

Configure the MMS server to send data from the RTAC to as many as 10 remote MMS client sessions. Configure logical nodes, tags, and buffered and unbuffered reports by using ACSELERATOR Architect software. In firmware versions prior to R147, 25 buffered reports and 25 unbuffered reports are supported. Starting in version R147, 100 buffered reports and 100 unbuffered reports are supported. Map tags in the RTAC to the MMS server shared map tags by using the Tag Processor or IEC 61131 logic. Configure the MMS server to respond to anonymous polls or configure as many as 10 IP addresses for the connected clients.

Peer-to-Peer Real-Time Status and Control

Use GOOSE with as many as 150 incoming (receive) and 150 outgoing (transmit) messages. Map received GOOSE tags to RTAC tags and map any RTAC tags to GOOSE transmit tags.

Configuration

Use ACSELERATOR Architect SEL-5032 Software to configure the Substation Configuration Description file (.SCD) or SEL Architect Project (selaprj). Import the configuration file into ACSELERATOR RTAC to complete configuration and load the file into the RTAC.

Commissioning and Troubleshooting

Use software such as MMS Object Explorer and AX-S4 MMS from Sisco, Inc., to browse the relay logical nodes and verify functionality.

This section presents the information you need to use the IEC 61850 features of the RTAC:

➤ *Introduction to IEC 61850 on page 119*
➤ *IEC 61850 Operation on page 120*
➤ *IEC 61850 Configuration on page 126*
➤ *ACSI Conformance Statements on page 144*

Introduction to IEC 61850

In the early 1990s, the Electric Power Research Institute (EPRI) and the Institute of Electrical and Electronics Engineers, Inc. (IEEE) began to define a Utility Communications Architecture (UCA). They initially focused on intercontrol center and substation-to-control center communications and produced the Inter-Control Center Communications Protocol (ICCP) specification. This specification, later adopted by the IEC as 60870-6 TASE.2, became the standard protocol for real-time exchange of data between databases. In 1994, EPRI and IEEE began work on UCA 2.0 for Field Devices (simply referred to as UCA2). In 1997, they combined efforts with Technical Committee 57 of the IEC to create a common international standard. Their joint efforts created the present IEC 61850 standard. The IEC 61850 standard, a superset of UCA2, contains most of the UCA2 specification, plus additional functionality. The standard describes client/server and peer-to-peer communications, substation design and configuration, testing, and project standards. The IEC 61850 standard consists of the parts listed in *Table 2.36*.

Table 2.36 IEC 61850 Document Set

IEC 61850 Sections	Definitions
IEC 61850-1	Introduction and overview
IEC 61850-2	Glossary
IEC 61850-3	General requirements
IEC 61850-4	System and project management
IEC 61850-5	Communication requirements
IEC 61850-6	Configuration description language for substation IEDs
IEC 61850-7-1	Basic communication structure for substations and feeder equipment- principles and models
IEC 61850-7-2	Basic communication structure for substations and feeder equipment- Abstract communication service interface (ACSI)
IEC 61850-7-3	Basic communication structure for substations and feeder equipment- Common data classes
IEC 61850-7-4	Basic communication structure for substations and feeder equipment- Compatible logical node (LN) classes and data classes
IEC 61850-8-1	SCSM-Mapping to Manufacturing Messaging Specification (MMS) (ISO/IEC 9506-1 and ISO/IEC 9506-2 over ISO/IEC 8802-3)
IEC 61850-9-1	SCSM-Sampled values over serial multidrop point-to-point link
IEC 61850-9-2	SCSM-Sampled values over ISO/IEC 8802-3
IEC 61850-10	Conformance testing

The IEC 61850 document set, available directly from the IEC at *www.iec.ch*, contains information necessary for successful implementation of this protocol. SEL strongly recommends that anyone involved with the design, installation, configuration, or maintenance of IEC 61850 systems be familiar with the appropriate sections of this standard.

IEC 61850 Operation

Object Models

The IEC 61850 standard relies heavily on the Abstract Communication Service Interface (ACSI) models to define a set of services and responses to those services. In terms of network behavior, abstract modeling enables all intelligent electronic devices (IEDs) to act identically. These abstract models are used to create objects (data items) and services that exist independently of any underlying protocols.

These objects are in conformance with the common data class (CDC) specification IEC 61850-7-3, which describes the type and structure of each element within a logical node. CDCs for status, measurements, controllable analogs and statuses, and settings all have unique CDC attributes. Each CDC attribute belongs to a set of functional constraints that groups the attributes into specific categories such as status (ST), description (DC), and substituted value (SV). Functional constraints, CDCs, and CDC attributes serve as building blocks for defining logical nodes.

UCA2 used GOMSFE (Generic Object Models for Substation and Feeder Equipment) to present data from station IEDs as a series of objects called models or bricks. The IEC working group has incorporated GOMSFE concepts into the standard, with some modifications to terminology; one change was the renaming of bricks to logical nodes. Each logical node represents a group of data (controls, status, measurements, etc.) associated with a particular function. For example, the MMXU logical node (polyphase measurement unit) contains measurement data and other points associated with three-phase metering including voltages and currents. Each IED can contain many functions such as protection, metering, and control. Multiple logical nodes represent the functions in multifunction devices.

Logical nodes can be organized into logical devices that are similar to directories on a computer disk. As represented in the IEC 61850 network, each physical device can contain many logical devices and each logical device can contain many logical nodes. Many relays, meters, and other IEC 61850 devices contain one primary logical device where all models are organized.

IEC 61850 devices are capable of self-description. You do not need to refer to the specifications for the logical nodes, measurements, and other components to request data from another IEC 61850 device. IEC 61850 clients can request and display a list and description of the data available in an IEC 61850 server device. This process is similar to the autoconfiguration process SEL communications processors (SEL-2032 and SEL-2030) use. Simply run an MMS browser to query devices on an IEC 61850 network and discover what data are available. Self-description also permits extensions to both standard and custom data models. Instead of having to look up data in a profile stored in its database, an IEC 61850 client can simply query an IEC 61850 device and receive a description of all logical devices, logical nodes, and available data.

Unlike other supervisory control and data acquisition (SCADA) protocols that present data as a list of addresses or indices, IEC 61850 presents data with descriptors in a composite notation made up of components. *Table 2.37* shows how the A-phase current expressed as MMXUAphsA$cVal breaks down into its component parts.

Table 2.37 Example IEC 61850 Descriptor Components

Component Descriptor	Component Type	Description
METMMXU1	Logical Node	Polyphase measurement unit
A	Data Object	Phase-to-ground amperes
phsA	Sub-Data Object	A-phase
cVal	Data Attribute	Complex value

Data Mapping

Device data are mapped to IEC 61850 logical nodes (LN) according to rules defined by SEL. Refer to IEC 61850-5:2003(E) and IEC 61850-7 4:2003(E) for the mandatory content and usage of these LNs. The RTAC logical nodes are grouped under logical devices for organization based on function. See *Table 2.38* for descriptions of the logical devices in an RTAC.

Table 2.38 RTAC Logical Devices

Logical Device	Description
ANN	Annunciator elements-alarms, status values
CFG	Configuration elements-DataSets and report control blocks

MMS

Manufacturing messaging specification (MMS) provides services for the application-layer transfer of real-time data within a substation LAN. MMS was developed as a network-independent data exchange protocol for industrial networks in the 1980s and standardized as ISO 9506. MMS supports complex named objects and flexible services that enable mapping to IEC 61850 in a straightforward manner. This was why the UCA users group used MMS for UCA from the start, and why the IEC chose to keep it for IEC 61850. Because the RTAC uses similar complex named objects, mapping to/from IEC 61850 and other tags in the RTAC is straightforward and easy to do. As an MMS client, the RTAC supports polling for DataSets and reports as defined in ACSELERATOR Architect. As an MMS server, the RTAC can respond to as many as 10 MMS client sessions as well as send reports such as defined in an ACSELERATOR Architect project.

GOOSE

The Generic Object Oriented Substation Event (GOOSE) object within IEC 61850 is for high-speed control messaging. IEC 61850 GOOSE automatically broadcasts messages containing status, controls, and measured values onto the network for use by other devices. IEC 61850 GOOSE sends the message several times, increasing the likelihood that other devices receive the messages. GOOSE message publication is a persistent function. Once GOOSE is enabled, the IED will continuously publish GOOSE messages, regardless of their contents, until they are disabled. The publication process description indicates when and why the publication rate changes.

IEC 61850 GOOSE objects can quickly and conveniently transfer status, controls, and measured values between peers on an IEC 61850 network. Configure the RTAC to send or receive DataSets with ACSELERATOR Architect. See the ACSELERATOR Architect instruction manual or online help for more information.

Each IEC 61850 GOOSE sender includes a text identification string (GOOSE Control Block Reference), APP ID field, and an Ethernet multicast group address in each outgoing message. Some devices that receive GOOSE messages use the text identification and multicast group to identify and filter incoming

GOOSE messages. The RTAC uses only the APP ID and multicast group to identify and filter incoming GOOSE messages. See *Table 2.36* for details on which logical nodes and attributes the RTAC uses. This information can be useful when searching through device data with MMS browsers.

SCL Files

Substation Configuration Language (SCL) is an XML-based configuration language used to support the exchange of database configuration data between different tools, which may come from different manufacturers. There are four types of SCL files:

- Intelligent Electronic Device (IED) Capability Description file (.ICD)
- System Specification Description (.SSD) file
- Substation Configuration Description file (.SCD)
- Configured IED Description file (.CID)

The ICD file describes the capabilities of an IED, including information on LN and GOOSE support. You can import ICD files from non-SEL IEDs into ACSELERATOR Architect. The SSD file describes the single-line diagram of the substation and the required LNs. The SCD file contains information on all IEDs, communications configuration data, and a substation description. The CID file, of which there may be several, describes a single instantiated IED within the project and includes address information.

DataSets

The RTAC supports publishing and subscribing to DataSets in GOOSE messaging. As an MMS client, the RTAC can also poll IEC 61850 IEDs for MMS DataSets and reports. You can create custom RTAC DataSets in ACSELERATOR Architect. The data tags are placeholders in the DataSet to which you will later map RTAC project data tags in ACSELERATOR RTAC. You can import CID files from other IEDs into ACSELERATOR Architect to map the IEDs DataSets to GOOSE or MMS messaging.

Within ACSELERATOR Architect, IEC 61850 DataSets have three main purposes:

- GOOSE: You can use predefined or edited DataSets, or create new DataSets for outgoing GOOSE transmission.
- MMS Client: You can define which data the RTAC will poll via MMS by selecting DataSets and/or reports from other IEDs in the Client Inputs section.
- MMS Server: You can define server sessions with which the RTAC will connect. You can also define the server model, DataSets, and/or reports that the RTAC will send to a polling MMS client.

Supplemental Software

Examine the data structure and values of the supported IEC 61850 LNs with an MMS browser such as MMS Object Explorer and AX-S4 MMS from Sisco, Inc.

Time Stamps and Quality

In addition to the various data values, the two attributes quality and t (time stamp) are available at any time. Creation of a time stamp results upon RTAC detection of a data or quality change.

The time stamp applies to all data and quality attributes (Boolean, Bstrings, Analogs, etc.) in the same fashion when the RTAC detects a data or quality change. Time stamps within GOOSE are in Universal Time (UTC) format but the RTAC database converts these to and from RTAC system time.

The RTAC uses GOOSE quality attributes to indicate the quality of the data in its transmitted GOOSE messages. Under normal conditions, all attributes are zero, indicating good quality data. Internal status indicators provide the information necessary for the device to set these attributes.

GOOSE Processing and Performance

SEL devices support GOOSE processing as defined by IEC 61850-7-1:2003(E), IEC 61850-7-2:2003(E), and IEC 61850-8-1:2004(E).

GOOSE Construction Tips

- Quality bit strings SEL devices publish are not generally useful in determining the quality of associated data because the SEL IEDs suspend publication of GOOSE messages if any quality attribute fails. Therefore, receipt of the message indicates that all quality attributes are normal. Do not include quality bit strings in published GOOSE messages unless some other type of IED requires this.
- Make GOOSE publications as small as possible. Include in the GOOSE publication only the information subscribing relays require.
- Give higher VLAN priority tags to more important GOOSE. This allows the network to preferentially forward those GOOSE to the subscribers, and it also gives a subscribing RTAC an indication that the more important GOOSE should be decoded before lower priority GOOSE.

GOOSE Subscription (Receive) Processing

The RTAC supports as many as 150 GOOSE subscriptions. GOOSE messages that arrive at the relay are subjected to the following processing steps.

Filter

The RTAC inspects each message for the proper multicast MAC address and GOOSE App ID. If those parameters match values the RTAC expects for one of the GOOSE subscriptions, then the RTAC passes the message on to the next level of processing. Otherwise, the RTAC discards the message. Each message on the LAN must have a unique combination of multicast MAC address and GOOSE App ID.

Buffer

The RTAC retains the most recent arrival for each of as many as 150 subscriptions. If a subsequent GOOSE arrives for a subscription that already has a buffered message, then the RTAC discards the earlier arrival.

Header Decoding

Each message contains a header that indicates the status of the message. The RTAC ignores the remainder of the message if any of four indicators in the message header is true:

Configuration Mismatch

The configuration number of the incoming GOOSE changes.

Needs Commissioning

This Boolean parameter of the incoming GOOSE message is true.

Test Mode

This Boolean parameter of the incoming GOOSE message is true.

State Number

This parameter is the same as the last time the message was decoded. State number increments when the contents of the message change, so if the state number is unchanged, there is no reason to decode the rest of the message.

The RTAC decodes and processes a message for which header decoding indicates message validity.

GOOSE Publication (Transmit) Processing

The RTAC supports as many as 150 GOOSE publications. Each publication can contain data from any logical node in the RTAC. The RTAC processes GOOSE at the rate of the task cycle time. GOOSE can run as part of the Automation Task or as part of the Main Task. An RTAC project that contains a mixture of protocols or large configurations may require more processing time for GOOSE processing than a project dedicated to GOOSE messaging.

Transmit Interval

The retransmit interval of GOOSE messages in the RTAC is based on the two settings min time and max time, which you configure in the CID file in the ACSELERATOR Architect project. If these settings are not a multiple of the task cycle time in which GOOSE is assigned, then the RTAC will use a multiple of the task cycle time that is closest to the actual min time and max time values from the CID file.

When the RTAC detects a data change of one or more GOOSE tags, it will transmit the change in a GOOSE message. Following an interval of min time, the RTAC retransmits the previous GOOSE message. The sequence number of each retransmitted message is incremented to indicate it is not an original message resulting from a data change. The RTAC then waits double the length of min time (2 • min time), increments the sequence number, and retransmits the same message. The RTAC continues to double the last retransmit interval, increment the sequence number, and retransmit the message. When the transmit interval exceeds max time, the RTAC uses max time as the retransmit interval for subsequent retransmits until the RTAC detects another data change.

IEC 61850 Configuration

Settings

The RTAC enables GOOSE messaging, MMS client, or MMS server support by verifying that the MOT string, or part number, stored in RTAC memory contains the IEC 61850 GOOSE or MMS options. The MOT is set at the factory if you order IEC 61850 GOOSE or MMS on a new RTAC. If you want to add IEC 61850 GOOSE or MMS to an existing RTAC, contact your SEL sales representative. You will receive a firmware update as well as an MOT upgrade file that you can load into the RTAC through use of the firmware upgrade procedure. The MOT upgrade file will update the RTAC MOT to enable GOOSE messaging, MMS client, or MMS server support. Configure all other IEC 61850 settings, including subscriptions to incoming GOOSE messages, with ACSELERATOR Architect.

ACSELERATOR Architect

The ACSELERATOR Architect software enables protection and integration engineers to design and commission IEC 61850 substations containing SEL IEDs.

Engineers can use ACSELERATOR Architect to do the following:

- Organize and configure all SEL IEDs in a substation project.
- Configure incoming and outgoing GOOSE messages.
- Edit and create DataSets for GOOSE or MMS.
- Edit and create MMS client DataSet and report polling configurations.
- Configure MMS server sessions as anonymous (reply to any MMS client poll) or specific by identifying as many as 10 MMS client IP addresses.
- Configure the MMS server model to include logical devices and logical nodes, which are placeholders for tags that you will later map to actual tags in the RTAC project.
- Read non-SEL IED Capability Description (ICD) and Configured IED Description (CID) files and determine the available IEC 61850 messaging options.
- Use or edit preconfigured DataSets for reports.
- Load IEC 61850 CID files into SEL IEDs.

- Generate SCD or SELPRJ configuration files for import into ACSELERATOR RTAC.
- Generate ICD files that will provide SEL IED descriptions to other manufacturers tools so they can use SEL GOOSE messages and reporting features.

ACSELERATOR Architect provides a graphical user interface (GUI) for engineers to select, edit, and create IEC 61850 GOOSE messages important for substation protection, coordination, and control schemes. The engineer can also select, edit and create IEC 61850 MMS client polling definitions as well as DataSets and reports for IEC 61850 MMS server for the RTAC. Typically, you first drag icons representing IEDs from the IED Palette to a substation container (in the Project Editor window), then edit the outgoing GOOSE messages or create new ones for each IED. You can also select incoming GOOSE messages for the RTAC to receive from any other IEDs in the domain. ACSELERATOR Architect has the capability to read other manufacturers ICD and CID files, enabling you to map the data seamlessly into SEL IED logic. See the ACSELERATOR Architect online help for more information.

Use ACSELERATOR Architect to configure all GOOSE transmit and receive messages, as well as to define MMS client inputs for connected devices.

Step 1. Open a new project in ACSELERATOR Architect by clicking **New Project** in the Project Editor pane.

Step 2. Drag an SEL_RTAC IED icon from the IED Palette onto the Project Editor. This is the RTAC you are configuring for IEC 61850 messaging.

Step 3. Drag the icons for the IEDs that will communicate to the RTAC from the IED Palette onto the Project Editor pane. You can import non-SEL ICD files into the **IED Palette** by right-clicking in the IED Palette pane and selecting **Import ICD**.

Step 4. Create new and edit existing DataSets. Remember that tags in the RTAC DataSets are placeholders for tag types. You can map project tags later in ACSELERATOR RTAC.

Step 5. Configure GOOSE Receive and GOOSE Transmit messages, if applicable.

Step 6. For MMS client configuration, click the **Client Inputs** tab. The Project Editor pane will display all DataSets and reports available from the IEDs. From the **Source** window, select the DataSets and reports you want the RTAC to gather from the IEDs and move these to the **Inputs** window tab. See *Figure 2.48*.

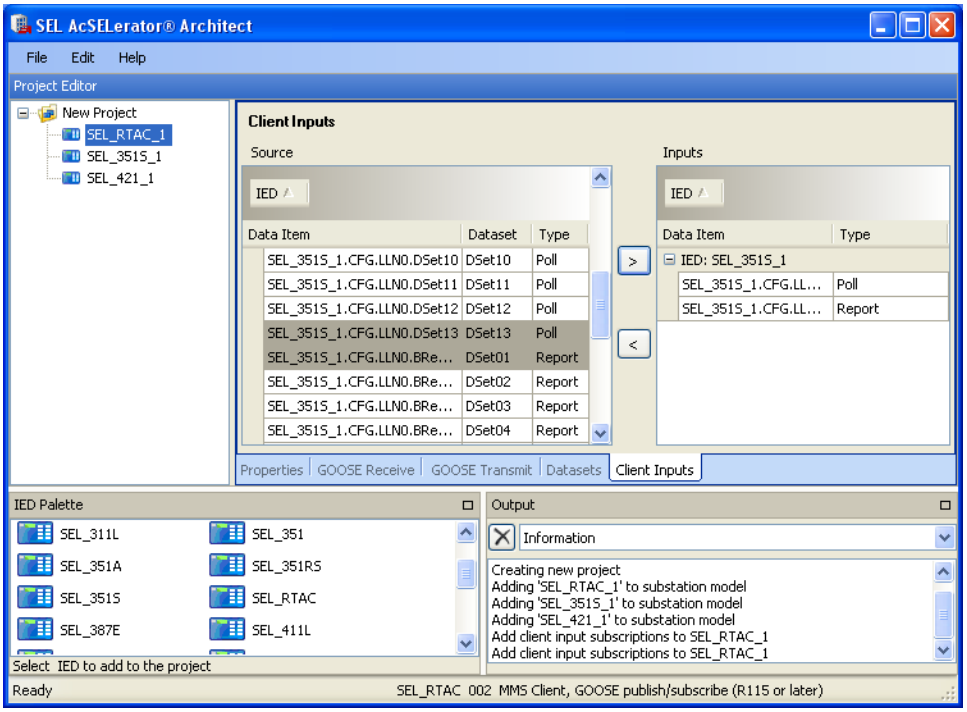


Figure 2.48 Configure IEC 61850 Messages

Step 7. Save the project.

Step 8. In ACSELERATOR RTAC, select **SET IEC 61850 Configuration** from the **Insert** ribbon and browse to the selaprj project file (see *Figure 2.49*).

Step 9. Select the file and click **Open**.

Data tags from the ACSELERATOR Architect project are now available in the RTAC project for mapping, manipulation, etc.

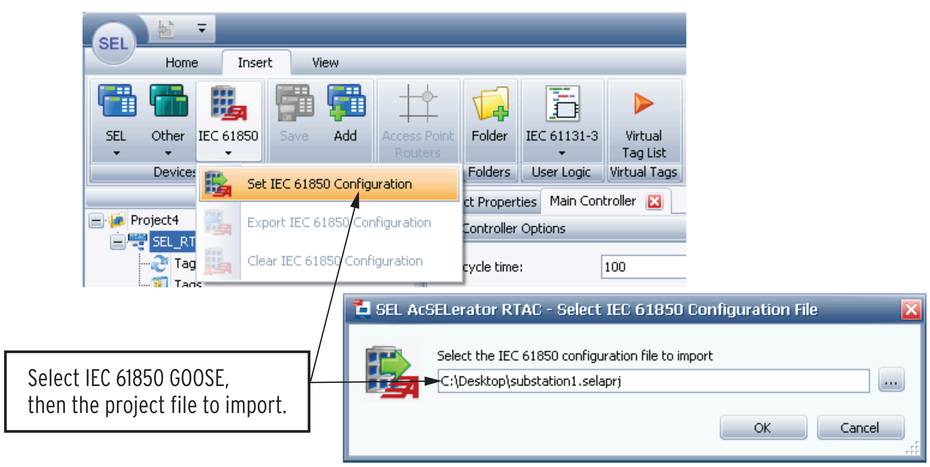


Figure 2.49 Insert GOOSE Configuration

Step 10. Configure which Ethernet interface the RTAC uses for GOOSE messaging by clicking on **SystemTags** under the device tree. Under settings, configure Ethernet_Port to use Eth_01 or Eth_02.

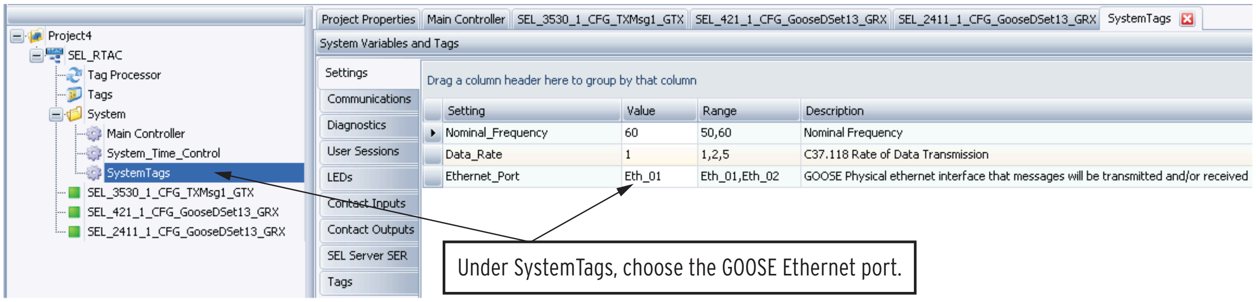


Figure 2.50 Configured GOOSE Ethernet

Because all IEC 61850 message configuration is done in ACSELERATOR Architect, you cannot make modifications to the message configuration in ACSELERATOR RTAC. *Table 2.39* describes the only configurable IEC 61850-related parameters in ACSELERATOR RTAC. Also, because the IEC 61850 configuration file contains all information necessary for the IEC 61850 communications network, ACSELERATOR RTAC allows only one ACSELERATOR Architect IEC 61850 configuration file per RTAC project. *Figure 2.51* shows how one IEC 61850 project file contains the configuration for all IEC 61850 devices on this RTAC.

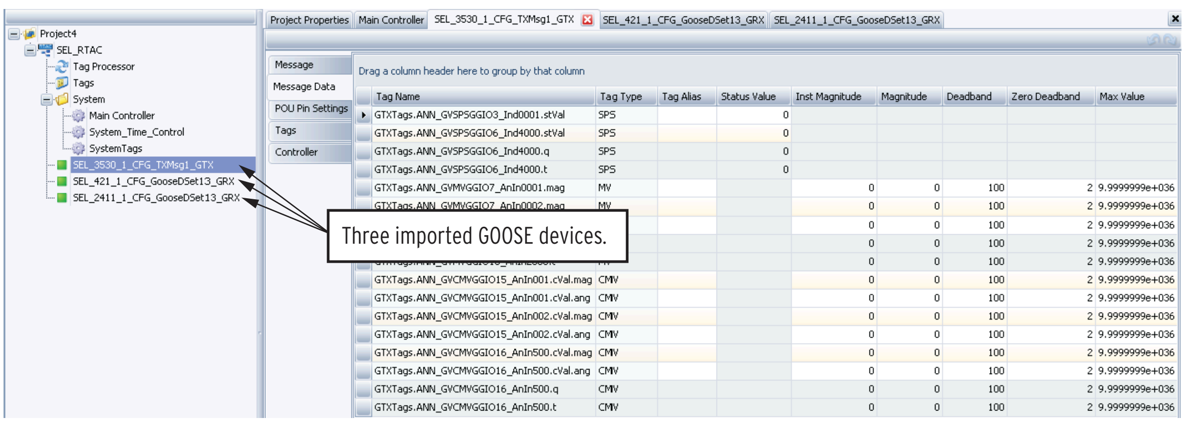


Figure 2.51 Configured GOOSE Devices

Table 2.39 Configurable GOOSE Parameters

Configurable Parameter	Description
Initial Status Value	Initial value for SPS or INS data types
Initial Measure Value	Initial Value for MV or CMV data types
Alias	Alias tag name
Ethernet_Port Under System Tags	Which Ethernet interface RTAC uses for GOOSE messages

Table 2.40 and *Table 2.41* show the maximum number of points possible in a single RX or TX GOOSE message. See *Table 2.42*.

Table 2.40 Simple Typed Global Tags Per RX or TX Message

IEC 61850 Data Type	IEC 61131 Data Type	Maximum Point Count Per Message
BOOLEAN	BOOL	468
FLOAT32	REAL	234
INT32	DINT	200
Timestamp	timeStamp_t	140
Quality	quality_t	351

Table 2.41 Aggregate Typed Global Tags Per RX or TX Message

IEC 61850 CDC Data Type	Data Included	Maximum Point Count Per Message
SPS	stVal, q, t	82
SPC	stVal, q, t	82
MV	mag, q, t	70
INS	stVal, q, t	66
BCR	actVal->INT128, q, t	42
Timestamp	timeStamp_t	140
Quality	quality_t	351

MMS Client File Services

Configuration of MMS File collection occurs only in the ACSELERATOR RTAC. The MMS file collection supports collection of COMTRADE files from any IEC 61850 MMS server, edition 1 or edition 2.

COMTRADE Collection

When the setting MMS_Comtrade_Enable is set to True, the client on the MMS_File_Services_ Period queries the MMS servers file system to find any folders called COMTRADE. The RTAC MMS client then collects all COMTRADE event files in that directory. This includes any potential COMTRADE events in directories inside the COMTRADE folder. The RTAC collects the COMTRADE files regardless of whether an event is zipped or separated into individual files on the server. Once the RTAC collects the files from the server, it stores the files on the RTAC file system as a ZIP file regardless of the original storage format of the COMTRADE file(s). All collected events are stored in the following location: /COMTRADE/*[IED name]*/*[directory structure in servers COMTRADE folder]*/*[event name on server]*.zip.

For an example of file naming and location, see the following:

IED Name
SEL_421_1

Event Location on Server
/COMTRADE/BKR_A/EVENT_00021.zip

Event Location on RTAC
/COMTRADE/SEL_421_1/BKR_A/EVENT_0021.zip

The time necessary to collect event files from a server is dependent upon the capability of the server. Edition 2 servers typically support the directory listing "/*", which returns a list of all files on the server in a single response. Using this response, the MMS client efficiently finds the new COMTRADE files to collect. If the server does not support this capability, the RTAC searches the servers file system one directory at a time. This may require many file directory polls to find all possible COMTRADE files and result in additional time for the MMS client to find and collect all COMTRADE files.

Criteria for Collecting Files

The MMS client collects all configured files if the following criteria are met:

- The MMS protocol standard has a maximum file path of 255 characters; the file path cannot exceed 255 characters.
- The file path contains only printable ASCII characters.
- The file path does not contain"." Or ".." as a "directory" or file name.

When a file is collected and placed on the RTAC file system, the files time stamp reflects the system time when the RTAC finished collecting the file. If a file name on the server changes case, the MMS client will neither rename nor attempt to again collect the file on the RTAC file system. If the RTAC collects a file from a server and that file is then deleted from the file system through either the web interface or another method, the RTAC will not collect that file again from the server.

The RTAC platform has several hardware variants with different memory capacity. The MMS client continues to collect files from servers until only 512 MB of free space on the RTAC file system remains. At this point, the MMS client ceases collection of additional files until additional free space on the file system is available.

MMS Server

Use ACSELERATOR Architect to also configure MMS server connections, DataSets, and reports for connected clients.

Step 1. Open a new project in ACSELERATOR Architect by clicking **New Project** in the **Project Editor** pane.

Step 2. Drag an SEL_RTAC IED icon from the **IED Palette** onto the **Project Editor**. In the resulting **IED Properties** window, pick the entry that includes MMS Server.

Note: You can configure MMS client, GOOSE, and MMS server for this one RTAC IED in the **Project Editor**. Follow the instructions for MMS client and GOOSE messaging, if needed, before proceeding to *Step 3*.

Step 3. Click the **Server Sessions** tab, then **Edit** or **New** to edit the existing default session or create as many as 10 sessions. Each session is a connection to a unique MMS client.

Step 4. In the **Session (Edit)** window, select the **Anonymous connections allowed** box to permit anonymous sessions, or clear the box and enter as many as 10 IP addresses of the client(s) with which the RTAC will communicate. The **Name** field (default being **Anonymous_Server**) is the name that will appear in the ACSELERATOR RTAC project in the device tree. Click **OK** when finished.

Step 5. Click on the **Server Model** tab, then click **Edit**. In this edit window you can create and modify MMS logical devices and logical nodes. The default logical device has two logical nodes, specifically **LLN0** and **LPHD0**. These cannot be modified.

Step 6. Create a new logical device by clicking **Logical Devices** +. Notice that all logical devices have two non-editable logical nodes. Click **Logical Nodes** + to create a new editable logical node. You can also hover your cursor in the window for selectable tooltips. See *Figure 2.52*.

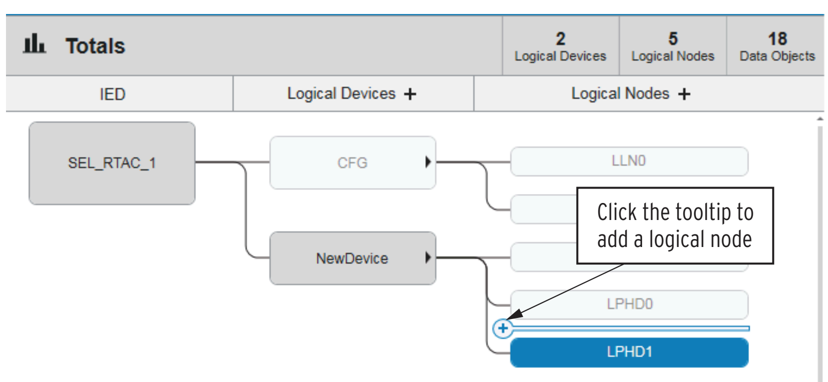


Figure 2.52 Add a New Logical Node

Step 7. Click the **Class** drop-down box to change the class of this logical node. For example, change it to a GGIO type. See *Figure 2.53*.

Hint: You can quickly find the class type by typing what you are looking for in the **Class** drop-down box.

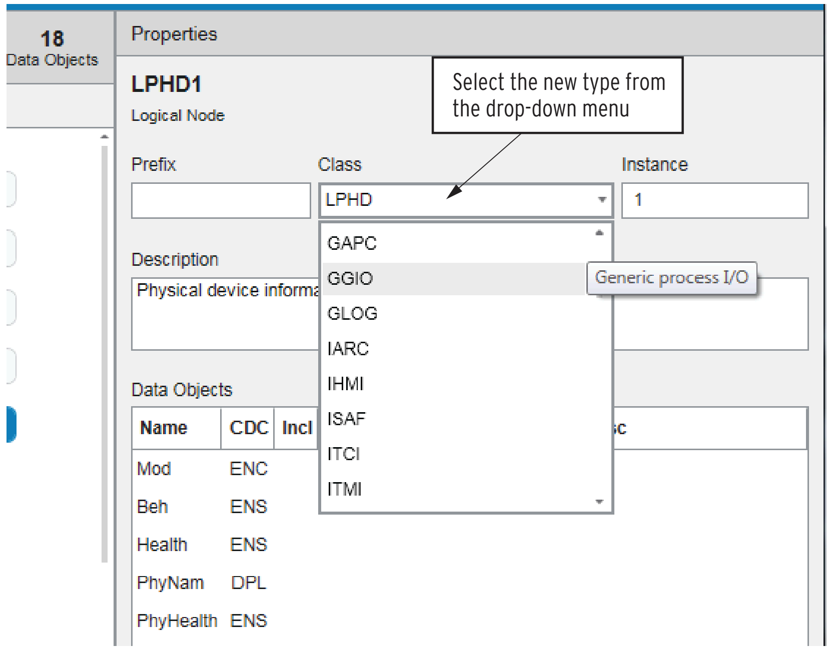


Figure 2.53 Changing the Logical Node Type

Step 8. Create data objects by selecting boxes or increasing the quantity (Qty) of the objects listed. See *Figure 2.54*. When finished, click **OK**.

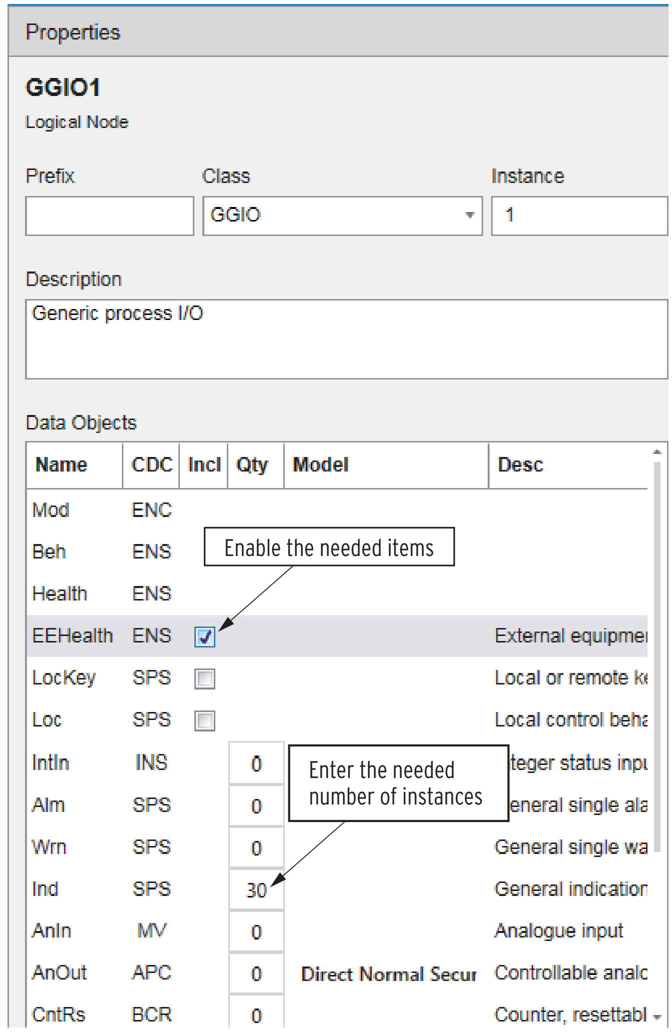


Figure 2.54 Creating Data Objects

Step 9. Click **Datasets** to create a new dataset or edit an existing dataset. Note the new logical device (named **NewDevice** by default) created in *Step 6* is available for mapping into a DataSet. See *Figure 2.55*.

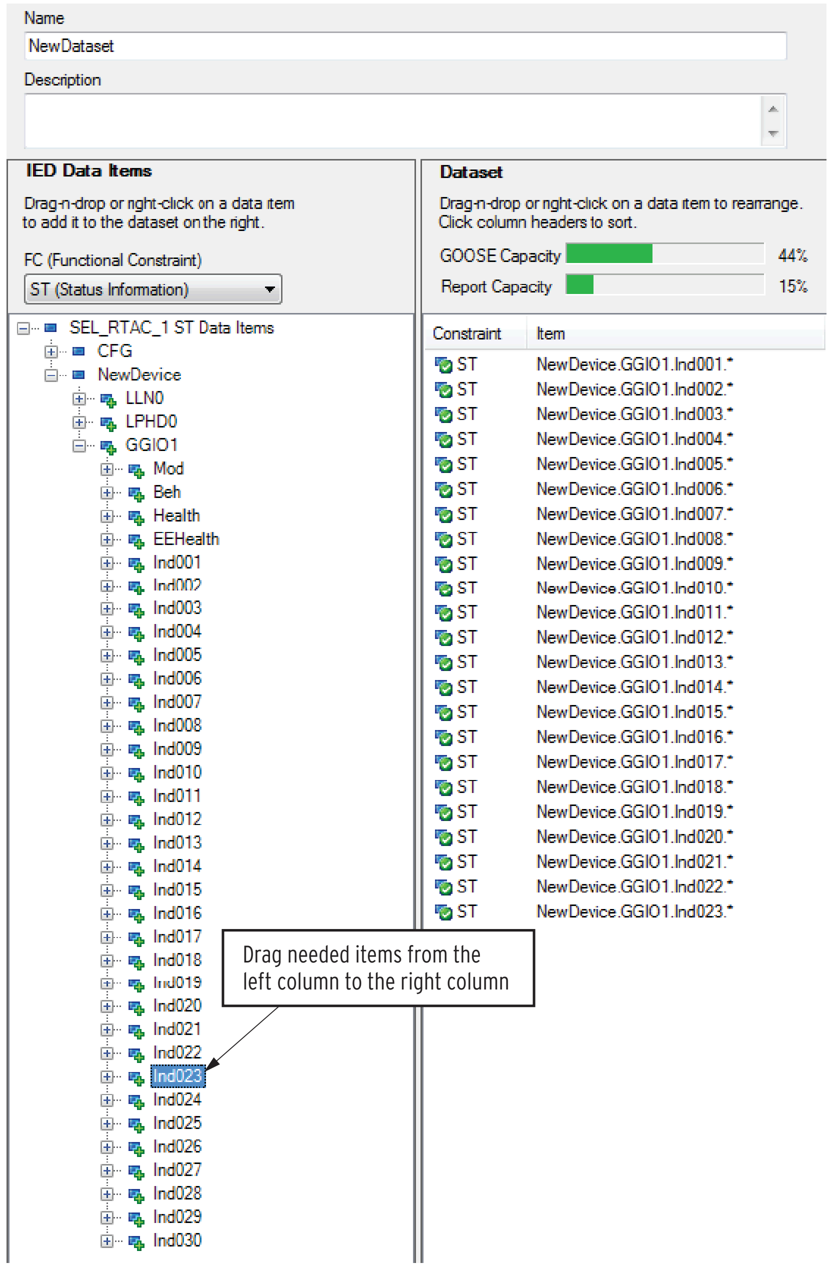


Figure 2.55 Create a Dataset

Step 10. Click on **Reports** and configure as needed. Configuration includes specifying if the report is buffered or unbuffered, which Dataset to use, and other configurations related to triggers and buffer times. See *Figure 2.56* for options.

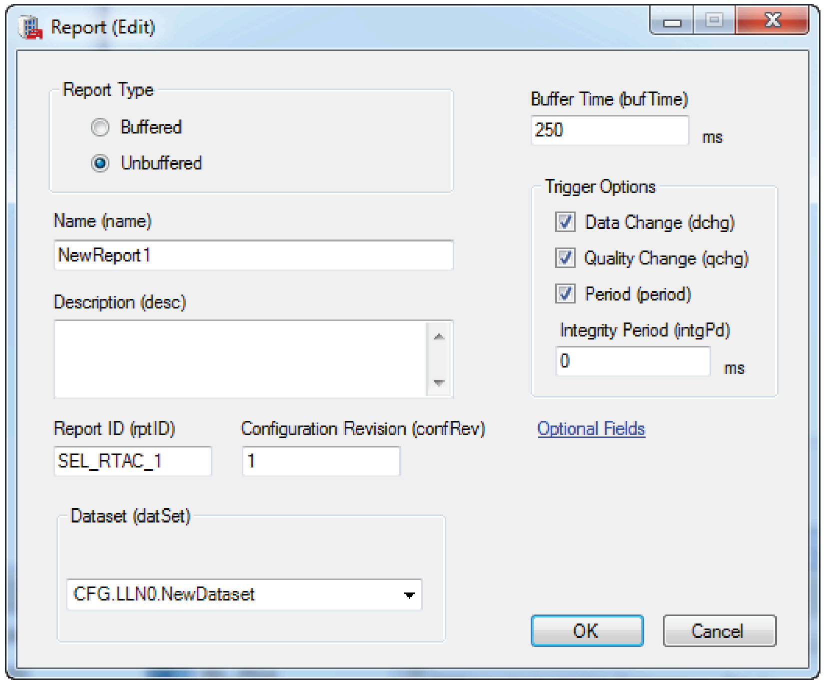


Figure 2.56 Configure Reports

Step 11. Save the project.

Step 12. In ACSELERATOR RTAC, select **SET IEC 61850 Configuration** from the **Insert** ribbon and browse to the **selaprj** project file (see *Figure 2.49*).

Step 13. Select the file and click **Open**.

Data tags from the ACSELERATOR Architect project are now available in the RTAC project for mapping, manipulation, etc.

Once the IEC 61850 configuration is set, there will be at least three new entries in the Device Tree.

1. **Anonymous_Server_850**: The default name for the server instance created in *Step 3*. There will be one of these for each of the instances created in *Step 3*. The only editable setting is allowance of anonymous connections; the IP addresses of the clients are editable if anonymous is not allowed. The POU pin settings indicate communications statistics for this one MMS server instance.
2. **SEL_RTAC_1**: The MMS server shared map, bearing the name of the IEC 61850 IED created in *Step 2*. SEL_RTAC_1 is the default name but may be changed in the ACSELERATOR Architect project. This shared map contains all the MMS server tags created in the Architect project. As with any server tags in an RTAC project, you can populate values in these tags by mapping them in the Tag Processor or in IEC 61131 logic.
3. **MMS_Server**: The POU of the actual server. Unlike other server protocols in the RTAC, there is only one actual server for MMS, with sessions for each polling client. There will be, therefore, only one of these entries. The only setting is to enable authentication and define the

authentication string, which actions will apply to all MMS server sessions defined in the RTAC project. The POU pins provide the ability to disable the server, disable the server tag updates, and disable the server controls. All of these actions will apply to all MMS server sessions.

Using MMS Server Tags

Some of the tags in the MMS server shared map are dedicated to use by the MMS server, but you can use the tags you created in Architect anywhere in the RTAC project as you would other IED tags. A unique feature to MMS server tags is that ACSELERATOR RTAC software can automatically move tags to different tasks if needed. To illustrate this feature, follow these steps:

Step 1. Create an MMS server in Architect.

Step 2. Insert that IEC 61850 configuration into an RTAC project and save the project.

Step 3. Create a program in the RTAC project.

Step 4. In that program, assign a value to one of the MMS server tags. For example: **SEL_RTAC_1.NewDevice.GGIO1.Loc.stVal := true;**

Step 5. In the **Main Controller** tab, click the program name, then select **Automation** from the **Move To** drop-down menu. The task will now execute at the speed of the Automation task. Generally you cannot use tags from an Automation task in the Main task, but in this case, the software will move only that one tag from the MMS server shared map into the Automation task. Though it is now native to the Automation task, the MMS server (in the Main task) still can access it.

Note: Once this move has taken place, you can no longer access this tag outside of the Automation task.

To view which tasks have been automatically moved, click the **View** ribbon, then the **Task Tags** icon for a textual report.

IEC 61850 MMS Server File Services

Use the setting Enable_File_Transfer to enable file service in the ACSELERATOR RTAC on the MMS server. When this value is True, the MMS server responds to file requests. When this settings is set to False, the MMS server rejects file service requests.

The MMS server has the following files available.

- The CID file from the presently configured RTAC (named "SET_61850.CID")
- All event files collected from SEL clients
- All files collected from MMS Clients
- All files collected from Modbus Clients
- All files generated by Axion I/O modules
- All files created or collected using the IEC 61131 file IO library

Note the following:

- The MMS protocol standard has a maximum file path of 255 characters. Any file on the RTAC file system whose file path name exceeds 255 characters will be unavailable through MMS file transfer.
- The MMS file server is case sensitive. An error message is returned for clients that issue file requests with incorrect casing.
- Any file request the MMS server receives that contains path constructs (e.g., /./, /../, etc.) will generate an error.
- The MMS server supports the following wildcard characters:
  - *—matches any character at its position
  - ?—matches zero or one character at its position
- A wildcard character cannot be the last character in a directory name. The server returns a file name-syntax error for such a situation.

Protocol Implementation Conformance Statement: RTAC

Table 2.42 and *Table 2.43* are as shown in the IEC 61850 standard, Part 8-1, Section 24. Note that because the standard explicitly dictates which services and functions must be implemented to achieve conformance, only the optional services and functions are listed.

Table 2.42 PICS for A-Profile Support

	Profile	Client	Server	Value/Comment
A1	Client/Server	Y	N	
A2	GOOSE/GSE management	Y	Y	Only GOOSE, not GSSE management
A3	GSSE	N	N	
A4	Time Sync	Y	Y	

Table 2.43 PICS for T-Profile Support

	Profile	Client	Server	Value/Comment
T1	TCP/IP	Y	N	
T2	OSI	N	N	
T3	GOOSE/GSE	Y	Y	Only GOOSE, not GSSE
T4	GSSE	N	N	
T5	Time Sync	Y	Y	

Refer to the ACSI Conformance Statements starting with *Table 2.55* for information on the supported services.

MMS Conformance

The Manufacturing Message Specification (MMS) stack provides the basis for many IEC 61850 protocol services. *Table 2.44* defines the service support requirement and restrictions of the MMS services in the RTAC.

Generally, only those services whose implementation is not mandatory are shown. Refer to the IEC 61850 standard Part 8-1 for more information.

Table 2.44 MMS Service Supported Conformance

MMS Service Supported CBB	Client-CR Supported	Server-CR Supported
status	Y	Y
getNameList		Y
identify	Y	Y
rename		
read	Y	Y
write	Y	Y
getVariableAccessAttributes		Y
defineNamedVariable		
defineScatteredAccess		
getScatteredAccessAttributes		
deleteVariableAccess		
defineNamedVariableList		
getNamedVariableListAttributes		Y
deleteNamedVariableList		
defineNamedType		
getNamedTypeAttributes		
deleteNamedType		
input		
output		
takeControl		
relinquishControl		
defineSemaphore		
deleteSemaphore		
reportPoolSemaphoreStatus		
reportSemaphoreStatus		
initiateDownloadSequence		
downloadSegment		
terminateDownloadSequence		
initiateUploadSequence		
uploadSegment		
terminateUploadSequence		
requestDomainDownload		
requestDomainUpload		

MMS Service Supported CBB	Client-CR Supported	Server-CR Supported
loadDomainContent		
storeDomainContent		
deleteDomain		
getDomainAttributes		Y
createProgramInvocation		
deleteProgramInvocation		
start		
stop		
resume		
reset		
kill		
getProgramInvocationAttributes		
obtainFile		
defineEventCondition		
deleteEventCondition		
getEventConditionAttributes		
reportEventConditionStatus		
alterEventConditionMonitoring		
triggerEvent		
defineEventAction		
deleteEventAction		
alterEventEnrollment		
reportEventEnrollmentStatus		
getEventEnrollmentAttributes		
acknowledgeEventNotification		
getAlarmSummary		
getAlarmEnrollmentSummary		
readJournal		
writeJournal		
initializeJournal		
reportJournalStatus		
createJournal		
deleteJournal		
fileOpen		
fileRead		
fileClose		

MMS Service Supported CBB	Client-CR Supported	Server-CR Supported
fileRename		
fileDelete		
fileDirectory		
unsolicitedStatus		
informationReport	Y	Y
eventNotification		
attachToEventCondition		
attachToSemaphore		
conclude	Y	Y
cancel	Y	Y
getDataExchangeAttributes		
exchangeData		
defineAccessControlList		
getAccessControlListAttributes		
reportAccessControlledObjects		
deleteAccessControlList		
alterAccessControl		
reconfigureProgramInvocation		

Table 2.45 lists specific settings for the MMS parameter Conformance Building Block (CBB).

Table 2.45 MMS Parameter CBB

MMS Parameter CBB	Client-CR Supported	Server-CR Supported
STR1	Y	Y
STR2	Y	Y
VNAM	Y	Y
VADR	Y	Y
VALT	Y	Y
TPY		Y
VLIS	Y	Y
CEI		

The following variable access conformance statements are listed in the order specified in the IEC 61850 standard, Part 8-1. Generally, only those services whose implementation is not mandatory are shown. Refer to the IEC 61850 standard Part 8-1 for more information.

Table 2.46 AlternateAccessSelection Conformance Statement

AlternateAccessSelection	Client-CR Supported	Server-CR Supported
accessSelection	Y	Y
component	Y	Y
index		
indexRange		
allElements		
alternateAccess	Y	Y
selectAccess	Y	Y
component	Y	Y
index		
indexRange		
allElements		

Table 2.47 VariableAccessSpecification Conformance Statement

VariableAccessSpecification	Client-CR Supported	Server-CR Supported
listOfVariable	Y	Y
variableSpecification	Y	Y
alternateAccess		

Table 2.48 VariableSpecification Conformance Statement

VariableSpecification	Client-CR Supported	Server-CR Supported
name	Y	Y
address		
variableDescription		
scatteredAccessDescription		
invalidated		

Table 2.49 Read Conformance Statement

VariableAccessSpecification	Client-CR Supported	Server-CR Supported
Request		
specificationWithResult		
variableAccessSpecification		
Response		
variableAccessSpecification	Y	Y
listOfAccessResult	Y	Y

Table 2.50 GetVariableAccessAttributes Conformance Statement

GetVariableAccessAttributes	Client-CR Supported	Server-CR Supported
Request		
name		

GetVariableAccessAttributes	Client-CR Supported	Server-CR Supported
address		
Response		
mmsDeletable		
address		
typeSpecification		

Table 2.51 DefineNamedVariableList Conformance Statement

DefineVariableAccessAttributes	Client-CR Supported	Server-CR Supported
Request		
variableListName		
listOfVariable		
variableSpecification		
alternateAccess		
Response		

Table 2.52 GetNamedVariableListAttributes Conformance Statement

GetNamedVariableListAttributes	Client-CR Supported	Server-CR Supported
Request		
ObjectName		
Response		
mmsDeletable		Y
listOfVariable		Y
variableSpecification		Y
alternateAccess		Y

Table 2.53 DeleteNamedVariableList Conformance Statement

DeleteNamedVariableList	Client-CR Supported	Server-CR Supported
Request		
Scope		
listOfVariableListName		
domainName		
Response		
numberMatched		
numberDeleted		
DeleteNamedVariableList-Error		

GOOSE Services Conformance Statement

Table 2.54 GOOSE Conformance

	Subscriber	Publisher	Value/Comment
GOOSE Services	Y	Y	
SendGOOSEMessage	Y	Y	
GetGoReference			
GetGOOSEElementNumber			
GetGoCBValues	Y	Y	
SetGoCBValues			
GSENotSupported			
GOOSE Control Block (GoCB)	Y	Y	

ACSI Conformance Statements

The following ACSI conformance statements are listed in the order specified in the IEC 61850 standard, Part 7-2 with firmware R121. Generally, only those services whose implementation is not mandatory are shown.

Table 2.55 Basic Conformance Statement

		Client/Subscriber	Server/Publisher	Value/Comments
	Client-Server Roles			
B11	**Server** side of (TWO-PARTY-APPLICATION-ASSOCIATION)	—	Y	
B12	**Client** side of (TWO-PARTY-APPLICATION-ASSOCIATION)	Y	—	
	SCSMs Supported			
B21	**SCSM:** IEC 6185-8-1 used	Y	Y	
B22	**SCSM:** IEC 6185-9-1 used			
B23	**SCSM:** IEC 6185-9-2 used			
B24	**SCSM:** other			
	Generic Substation Event Model (GSE)			
B31	**Publisher** side	—	Y	
B32	**Subscriber** side	Y	—	
	Transmission of Sampled Value Model (SVC)			
B41	**Publisher** side			
B42	**Subscriber** side			

—
Y = supported
N or empty = not supported

Table 2.56 ACSI Models Conformance Statement

		Client/Subscriber	Server/Publisher
If Server or Client side (B11/B12) supported			
M1	**Logical device**	Yes	Yes
M2	**Logical node**	Yes	Yes
M3	**Data**	Yes	Yes
M4	**Data set**	Yes	Yes
M5	**Substitution**		
M6	**Setting group control**		
	Reporting		
M7	**Buffered report control**	Yes	Yes
M7-1	sequence-number	Yes	Yes
M7-2	report-time-stamp	Yes	Yes
M7-3	reason-for-inclusion	Yes	Yes
M7-4	data-set-name	Yes	Yes
M7-5	data-reference	Yes	Yes
M7-6	buffer-overflow	Yes	Yes
M7-7	entryID	Yes	Yes
M7-8	BufTm	Yes	Yes
M7-9	IntgPd	Yes	Yes
M7-10	GI	Yes	Yes
M7-11	confRev	Yes	Yes
M8	**Unbuffered report control**	Yes	Yes
M8-1	sequence-number	Yes	Yes
M8-2	report-time-stamp	Yes	Yes
M8-3	reason-for-inclusion	Yes	Yes
M8-4	data-set-name	Yes	Yes
M8-5	data-reference	Yes	Yes
M8-6	BufTm	Yes	Yes
M8-7	IntgPd	Yes	Yes
M8-8	GI	Yes	Yes
M8-9	confRev	Yes	Yes
	Logging		
M9	**Log control**		
M9-1	IntgPd		
M10	**Log**		
M11	**Control**	Yes	Yes

		Client/Subscriber	Server/Publisher
If GSE B31/32 is supported			
M12	**GOOSE**	Yes	Yes
M13	**GSSE**		
If SVC (41/42) is supported			
M14	Multicast SVC		
M15	Unicast SVC		
If Server or Client side (B11/12) is supported			
M16	**Time**	Yes	Yes
M17	**File Transfer**		
Yes = service is supported No or empty = service is not supported			

Table 2.57 ACSI Service Conformance Statement

	Server	AA[a]: TP/MC	Client (C)	Server (S)	Comments
Server					
S1	GetServerDirectory	TP		Yes	
Application Association					
S2	Associate		Yes	Yes	
S3	Abort		Yes	Yes	
S4	Release		Yes	Yes	
Logical Device					
S5	GetLogicalDeviceCategory	TP		Yes	
Logical Node					
S6	GetLogicalNodeDirectory	TP		Yes	
S7	GetAllDataValues	TP		Yes	
Data					
S8	GetDataValues	TP	Yes	Yes	
S9	SetDataValues	TP			
S10	GetDataDirectory	TP		Yes	
S11	GetDataDefinition	TP		Yes	
Data Set					
S12	GetDataSetValues	TP	Yes	Yes	
S13	SetDataSetValues	TP			
S14	CreateDataSet	TP			
S15	DeleteDataSet	TP			
S16	GetDataSetDirectory	TP		Yes	

	Server	AA[a]: TP/MC	Client (C)	Server (S)	Comments
		Substitution			
S17	SetDataValues	TP			
		Setting Group Control			
S18	SelectActiveSG	TP			
S19	SelectEditSG	TP			
S20	SetSGValues	TP			
S21	ConfirmEditSGValues	TP			
S22	GetSGValues	TP			
S23	GetSGCBValues	TP			
		Reporting			
		Buffered Report Control Block (BRCB)			
S24	Report	TP	Yes	Yes	
S24-1	data-change (dchg)		Yes	Yes	
S24-2	quality-change (qchg)		Yes	Yes	
S24-3	data-update (dupd)		Yes		
S25	GetBRCBValues	TP	Yes	Yes	
S26	SetBRCBValues	TP	Yes	Yes	
		Unbuffered Report Control Block (URCB)			
S27	Report	TP	Yes	Yes	
S27-1	data-change (dchg)		Yes	Yes	
S27-2	quality-change (qchg)		Yes	Yes	
S27-3	data-update (dupd)		Yes		
S28	GetURCBValues	TP	Yes	Yes	
S29	SetURCBValues	TP	Yes	Yes	
		Logging			
		Log Control Block			
S30	GetLCBValues	TP			
S31	SetLCBValues	TP			
		Log			
S32	QueryLogByTime	TP			
S33	QueryLogAfter	TP			
S34	GetLogStatusValues	TP			
		Generic Substation Event Model (GSE)			
		GOOSE Control Block			
S35	SendGOOSEMessage	MC	Yes	Yes	
S36	GetGoReference	TP			

	Server	AA[a]: TP/MC	Client (C)	Server (S)	Comments
S37	GetGOOSEElementNumber	TP			
S38	GetGoCBValues	TP			
S39	SetGoCBValues	TP			
GSSE Control Block					
S40	SendGSSEMessage	MC			
S41	GetGsReference	TP			
S42	GetGSSEDataOffset	TP			
S43	GetGsCBValues	TP			
S44	SetGsCBValues	TP			
Transmission of Sampled Value Model (SVC)					
Multicast SVC					
S45	SendMSVMessage	MC			
S46	GetMSVCBValues	TP			
S47	SetMSVCBValues	TP			
Unicast SVC					
S48	SendUSVMessage	TP			
S49	GetUSVCBValues	TP			
S50	SetUSVCBValues	TP			
Control					
S51	Select	TP	Yes	Yes	
S52	SelectWithValue	TP	Yes		
S53	Cancel	TP		Yes	
S54	Operate	TP	Yes	Yes	
S55	CommandTermination	TP	Yes	Yes	
S56	TimeActivatedOperate	TP			
File Transfer					
S57	GetFile	TP	Yes	Yes	
S58	SetFile	TP			
S59	DeleteFile	TP		Yes	
S60	GetFileAttributeValues	TP	Yes	Yes	

	Server	AA[a]: TP/MC	Client (C)	Server (S)	Comments
			Time		
T1	Time resolution of internal clock		As many as 20	As many as 20	Nearest negative power of 2 (in seconds)
T2	Time accuracy of internal clock		T0	T0	T0 (10 ms) T1 (1 ms) T2 (100 µs) T3 (25 µs) T4 (4 µs) T5 (1 µs)
T3	Supported TimeStamp resolution		20 (1 µs)	10 (1 ms)	Nearest negative power of 2 (in seconds). The value of the TimeAccuracy attribute depends on the time-source accuracy of and variation of the internal clock relative to the time source.
				T0 or T1	For SNTP and NTP, T0 or T1 is typical. If accuracy is less than 10 ms, it will be set to "Unspecified".

[a]Type of application association (AA): Two Party (TP) or Multicast (MC).

Model Implementation

This model implementation conformance statement is applicable for SEL RTAC, with firmware R136:

The MICS document specifies the modeling extensions compared to IEC 61850 Edition 1. For the exact details on the standardized model please compare the ICD substation configuration file: "03530 006.ICD".

Common data class extensions

2.1 Supported common data classes

The IEC 61850 client and server in the SEL RTAC are capable of mapping the mandatory attributes of the following data classes with a functional constraint of ST or MX:

Common data class specifications for description information

Common data class specifications for status information

- Single point status (SPS)
- Double point status (DPS)
- Integer status (INS)
- Enumerated status (ENS)
- Protection activation information (ACT)
- Directional protection activation information (ACD)
- Binary counter reading (BCR)

Common data class specifications for measurand information

- Measured value (MV)
- Complex measured value (CMV)
- Phase to ground related measured values of a three phase system (WYE)
- Phase to phase related measured values of a three phase system (DEL)
- Sequence (SEQ)

Common data class specifications for controllable status information

- Controllable single point (SPC)
- Controllable double point (DPC)
- Controllable integer status (INC)
- Controllable enumerated status (ENC)
- Binary controlled step position information (BSC)
- Integer controlled step position information (ISC)

Common data class specifications for controllable analog information

- Controllable analog set point information (APC)
- Binary controlled analog process value (BAC)

In addition to directly supporting the mapping of data from the above data classes, the IEC 61850 client and server support the ability to access most attributes within any data class (supported or not), mandatory, optional, or specialized.

2.2 Unsupported common data classes

Common data class specifications for description information

- Device name plate (DPL)
- Logical node name plate (LPL)
- Curve shape description (CSD)

Common data class specifications for status information

- Security violation counting (SEC)

Common data class specifications for measurand information

- Sampled value (SAV)
- Harmonic Value (HMV)
- Harmonic value for WYE (HWYE)
- Harmonic value for DEL (HDEL)

Common data class specifications for status settings

- Single point setting (SPG)
- Integer status setting (ING)
- Enumerated status setting (ENG)
- Object reference settings (ORG)

- Time setting group (TSG)
- Currency setting group (CUG)
- Visible string setting (VSG)

Common data class specifications for analog settings

- Analog setting (ASG)
- Setting curve (CURVE)
- Curve shape settings (CSG)

EtherCAT

This section describes the configuration and use of EtherCAT® protocol with ACSELERATOR RTAC. EtherCAT is unique from other protocols available for the RTAC because connected I/O modules (EtherCAT servers) are automatically configured during network initialization. This means that you will provide configuration information for both EtherCAT servers and the EtherCAT client (SEL RTAC) in the ACSELERATOR RTAC project. You can continue to use all other client, server, and peer-to-peer protocols, such as SEL or DNP3, in an ACSELERATOR RTAC project that includes EtherCAT I/O.

NOTE
The following devices currently do not support the EtherCAT communications protocol: SEL-3505, SEL-3505-3, and SEL-3532.

The primary steps needed to configure an EtherCAT I/O network in ACSELERATOR RTAC are as follows:

- Designate an Ethernet port for EtherCAT (only needed when using the SEL-3530, SEL-3530-4, or SEL-3555). On the SEL-3555, if the synchrophasor protocol is going to be used and you need a high-accuracy time source, this port must be a PCI expansion Ethernet port with IRIG-B connected to the card.
- Create device connections for each power coupler module, digital input module, analog input module, and digital output module. You can configure as many as 60 I/O modules on a single EtherCAT network.
- Select the physical placement of modules and node network connections.
- Map I/O tags into needed Tag Processor locations or custom logic programs.

NOTE
Always ensure that power couplers are not connected to any Ethernet switches or other Ethernet devices, including RJ45-to-fiber-optic converters. The power coupler EtherCAT ports are only for direct connections between power couplers of the same type or to an RTAC EtherCAT port. Connecting a power coupler to any other Ethernet device can cause the EtherCAT network to stop communicating completely or behave unpredictably.

Designating an Ethernet Port for EtherCAT Use

The designated port will be dedicated for EtherCAT; no other protocol will be available on that port. If you are using the SEL-2241 RTAC module in an SEL-2240 Axion® node, EtherCAT is provided via the backplane connector. You cannot alter this selection or enable EtherCAT via the two front-panel Ethernet ports. Both of the Ethernet ports can be configured for any other available Ethernet protocol.

NOTE
The USB port and **ETH F** are unavailable for use with EtherCAT.

To enable an EtherCAT network by using either the SEL-3530, SEL-3530-4, or SEL-3555, you must first dedicate an Ethernet port for use by the I/O network. To configure the Ethernet port, use an Internet browser to connect with the RTAC web application (see *Section 5: Web HMI and Reports*) and navigate to the Ethernet Network page, as shown in *Figure 2.57*. Select the **List Ethernet Settings** tab. Click the **Edit** button corresponding to the port you want to use. If you have already selected a port (such as Eth_01 in *Figure 2.57*) to be the default gateway, you cannot use the gateway port for EtherCAT.

NOTE
EtherCAT connections on the SEL-3555 and SEL-3560E are only available on ports of a PCI expansion Ethernet card. IRIG should be connected to the PCI card internally to ensure high-accuracy time is available.

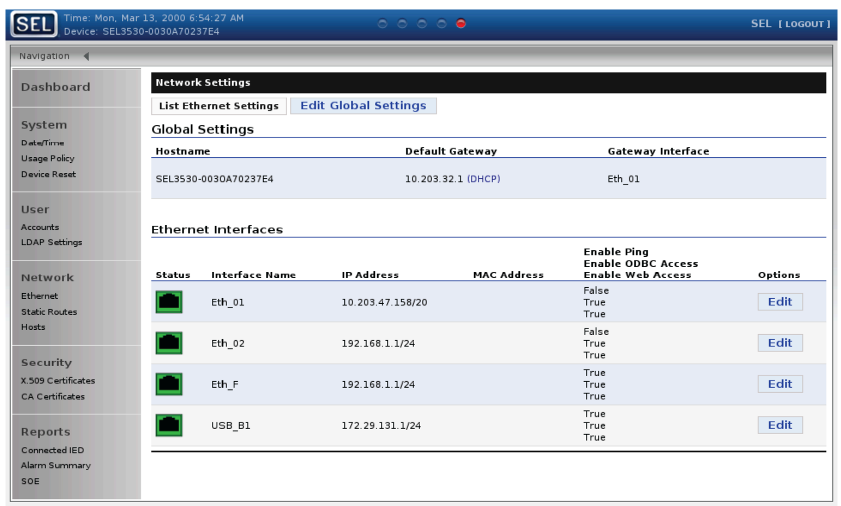


Figure 2.57 RTAC Ethernet Interfaces

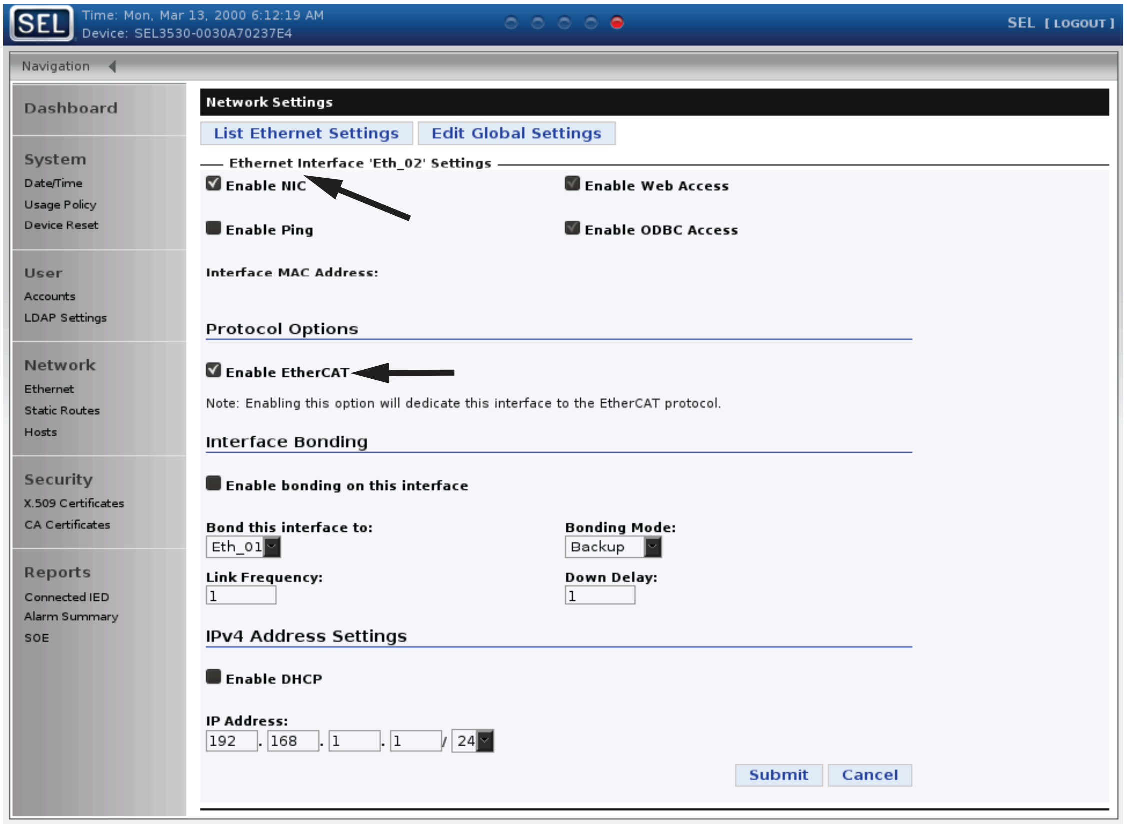


Figure 2.58 Configure a Dedicated EtherCAT Port

Verify that the port you selected is enabled. As shown in *Figure 2.58*, select the check box labeled **Enable NIC** if the port is disabled. Select the **Enable EtherCAT** check box and press **Submit** to complete the selection. The RTAC will automatically turn off other Ethernet interface services, such as web access, for your EtherCAT port. There is no requirement to manually deselect such items. Bonding, DHCP, and PRP are all unavailable on an EtherCAT port.

NOTE

Ethernet port settings will revert to previous values if you disable EtherCAT on a port.

Once you have saved settings, the Ethernet Interfaces page will display again in your browser. Notice that the port you selected for EtherCAT shows "EtherCAT Dedicated," as shown in the upper image of *Figure 2.59*. Only one port at a time can support EtherCAT. Notice the lower image in *Figure 2.59*, which shows the result of attempting to enable a second concurrent EtherCAT port.

NOTE

Do not assign 128.0.0.0 as the IP address on any Ethernet port if you are using an EtherCAT I/O network.

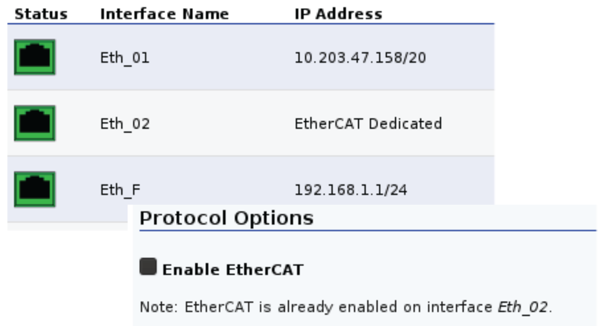


Figure 2.59 Results of EtherCAT Configuration

Once you have set the port, navigate to the **Ethernet Global Settings** tab to verify that the correct port is set to EtherCAT. The IP Address of the selected port will display "EtherCAT Dedicated," as shown in *Figure 2.60*. If you need to change the dedicated EtherCAT port at a later time, you must manually stop the EtherCAT protocol prior to making the setting change. Manually restart EtherCAT once the setting change is complete. Refer to *Replacing a Failed Module With Disabled Network* in the *SEL-2240 Instruction Manual* for the steps necessary to force the EN pin for the EtherCAT POU.

⚠**CAUTION**
Changing SFP ports while online can cause momentary communication interruptions. In the case of EtherCAT, this can cause the EtherCAT network to disable.

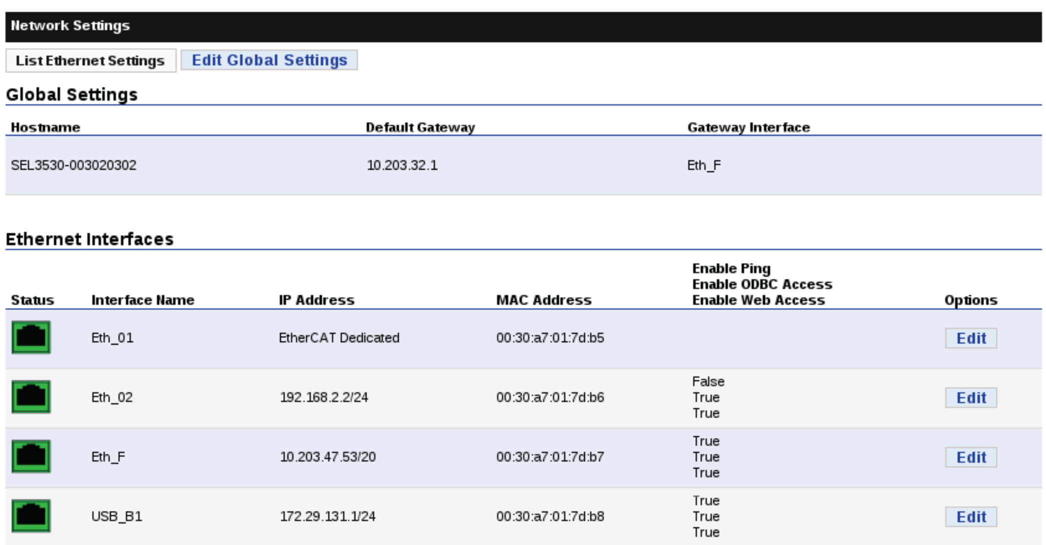


Figure 2.60 EtherCAT Port Unavailable for Other Applications

This configuration is unnecessary for the SEL-2241 RTAC module because the backplane port is permanently configured at the factory for EtherCAT.

Creating Device Connections for New Modules in the Project

The process for adding power couplers and I/O modules to an ACSELERATOR RTAC project is similar to the process for adding IED connections. Place the power couplers and I/O modules into your project by using the following steps. Also refer to *Figure 2.61*.

Step 1. Select **Fieldbus I/O** from the **Insert** ribbon.

Step 2. Hover or click the mouse on the **SEL** entry. Then hover your cursor or click the entry for the type of module you are adding to the project. Finally, click on the specific module type you need.

Step 3. Provide a module name in the dialog box that displays (or use the default name provided). IEC naming conventions are enforced; use an underscore rather than a space in the module name. The system appends the text "_ECAT" to each module name, so you can easily identify which modules in the project use EtherCAT versus other available protocols. The only **Connection Type** available will be **Client - Ethernet**. The I/O module itself is an EtherCAT server; the Connection Type selection refers to the method the RTAC uses to connect with the module.

Step 4. Press the **Insert** button.

Note in the inset photo of *Figure 2.61* that the RTAC places the new module by default at the end of the project view listing. On the other hand, if you select an existing folder prior to inserting a module, then the RTAC places the new module automatically in the folder you select.

There is no requirement that you add the I/O modules to the project in a specific sequence. The system will use the EtherCAT I/O *Node Connections Editor* to organize the modules based on their physical arrangement in a node. The arrangements of modules in the project view and I/O Node Connections Editor view do not need to match.

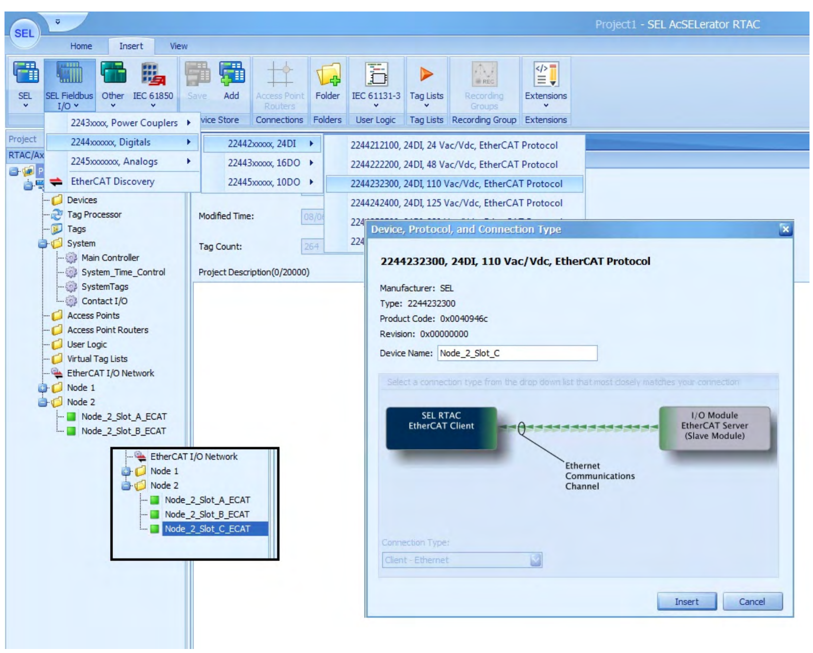


Figure 2.61 Adding SEL-2240 I/O Modules to a Project

Naming Conventions

As you plan a new project with the SEL-2240, carefully consider the node and I/O module naming conventions you will use and how these can help expedite your work during commissioning or troubleshooting.

Especially for the case of a large system with many nodes, consider a naming convention where each module name includes information about the node within which it resides. *Figure 2.61* shows a simple scheme to accomplish this. Each node is named sequentially (Node_1, Node_2). When we added a module to a particular node, we named the module by joining the node name with the physical slot in the node where we intended to place that module (i.e., Node_2_Slot_C_ECAT). If you do not know the intended physical location of a module, you can rename it at any time during the project development process. The module name automatically becomes a prefix for the tag names associated with that module, so this type of naming convention provides an efficient way to identify the physical location of a tag.

If nodes are mounted near the apparatus being monitored and controlled, then consider including a designation for the apparatus in the module naming convention. Many different naming methods will work well.

Organizing an EtherCAT I/O Network in an ACSELERATOR RTAC Project

An Axion system can include one or more I/O nodes that use EtherCAT protocol as a real-time network. When you load a project that includes I/O network information into an SEL-3530 RTAC, SEL-3530-4 RTAC, or SEL-2241 RTAC module, the system will automatically start and scan the EtherCAT network; during such scanning, the system verifies and reports proper operation. A single project can include no more than 60 I/O modules and power couplers. Because an RTAC project can include a large quantity of I/O modules, we highly recommend that you use the available tools in the project view of ACSELERATOR RTAC to organize the list of module connections. The project view has two tools available to help you keep the project organized. You can add custom folders, similar to a Microsoft Windows operating system folder organization, to collect devices into logical groups. A second option is to use drag-and-drop selections to quickly change the sequence of devices in the project view.

To add a custom folder, press the **Folder** button in the **Insert** menu, as shown in *Figure 2.62*. The software will ask for a name for each new folder you create. You can right-click on a folder to use the context menu and rename the folder at any time. Folder names follow no IEC requirements, so you can include spaces in names. ACSELERATOR RTAC also supports nesting of folders. Select the parent folder prior to pressing the **Folder** button in the **Insert** ribbon to create a child folder. Refer to *Figure 2.62* for an example.

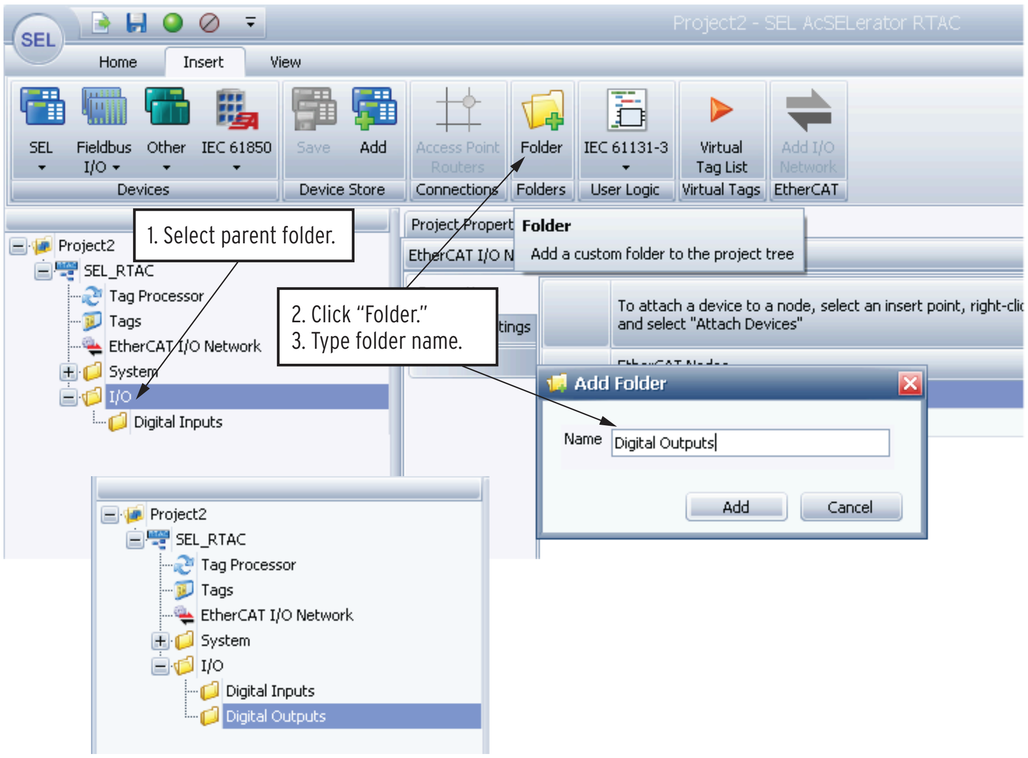


Figure 2.62 Creating Custom Project Folders

Once you add folders or devices to the project tree, organize them to meet your needs by using simple drag-and-drop operations. As a first example, consider a case where a module resides in one folder and we want to move it to a second folder.

To move an I/O module, refer to *Figure 2.63* and perform the following:

Step 1. Click and hold the mouse on the module name in the project view.

Step 2. While holding the mouse button, drag the module onto the desired folder name. Note the yellow arrow that appears to the left of the project view. This arrow points at the folder you have presently selected, and it indicates that the module will become a member of that folder.

Step 3. Release the mouse button.

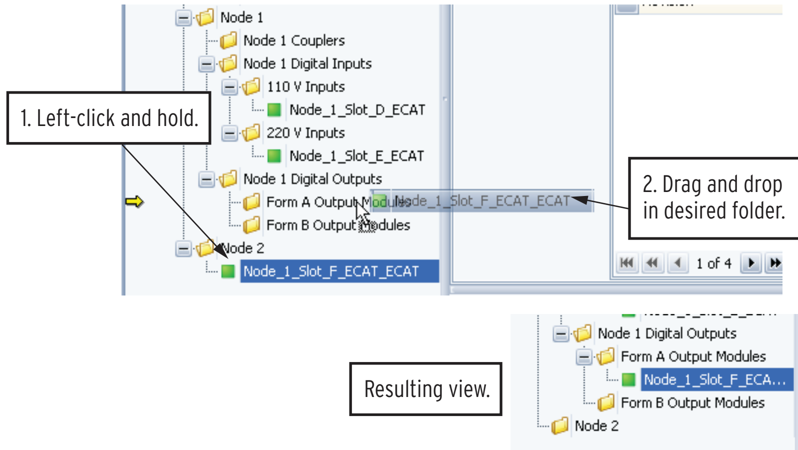


Figure 2.63 Moving an Object Between Folders in Project View

The same process works for folders or modules. If you move a folder, all of the contents (including subfolders) move with it. You cannot select and move multiple independent objects simultaneously.

As a second example, we want to change the sequence of folders within the project view. This operation is similar to the previous example, but we use a hot key in conjunction with the mouse. Please refer to *Figure 2.64*, and perform the following steps.

Step 1. Select the folder you want to move, by clicking and releasing the mouse.

Step 2. Press and hold the **<Shift>** key.

Step 3. Click, hold, and drag the folder to the position you want. Notice the curved blue arrow that appears to the left of the project view. This arrow shows the present mouse position and indicates that you are reordering the objects rather than dropping the objects into a parent folder.

Step 4. Release the mouse button and the **<Shift>** key.

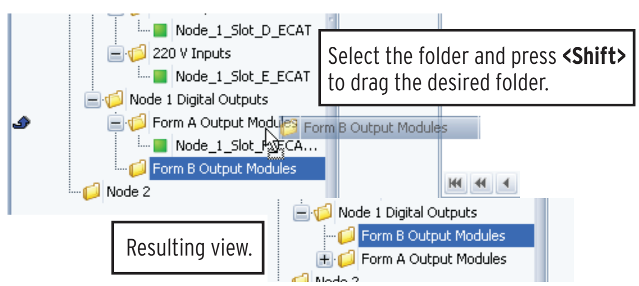


Figure 2.64 Changing Object Sequence in Project View

This procedure works for folders or modules. Perform the same steps if you want to move a folder or module from within a parent folder out to the same hierarchy level as the parent folder.

EtherCAT Discovery

EtherCAT Discovery allows a user to plug in their Axion modules as desired and automatically populate the EtherCAT I/O Network without any manual addition of modules in the software. To discover an Axion network, Select the **EtherCAT Discovery** option as shown in *Figure 2.65* to begin the discovery process for connected Axion modules.

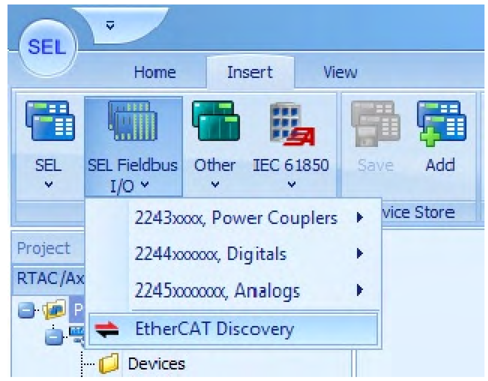


Figure 2.65 EtherCAT Discovery

EtherCAT Discovery requires you to go online with your RTAC to discover the Axion modules you have connected. The modules are subject to the same constraints governing physical placement of modules as the software requires. Log in as shown in *Figure 2.66*.

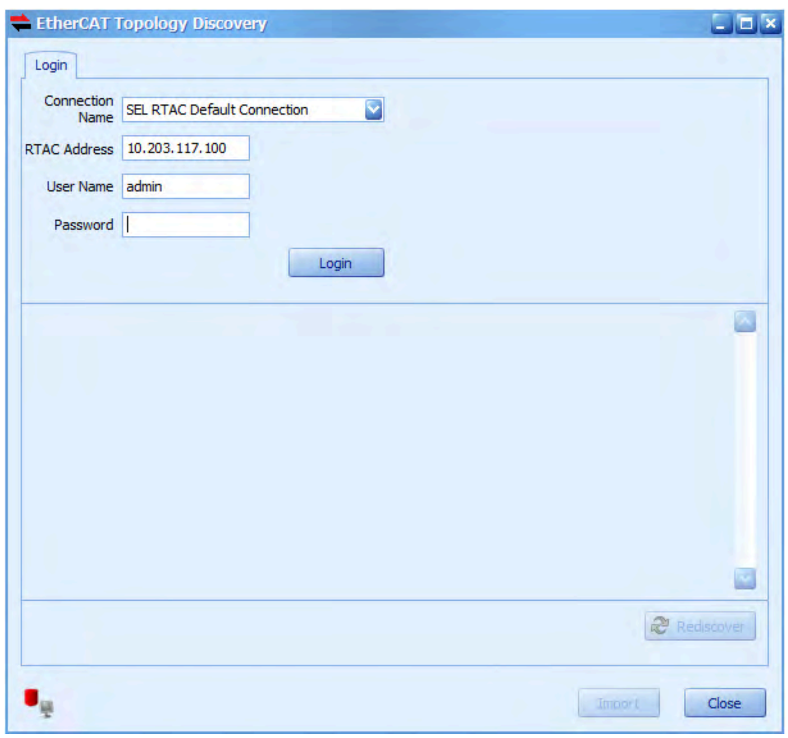


Figure 2.66 EtherCAT Topology Discovery Login

Once the EtherCAT Discovery process completes, click **Import** to accept the discovered network, as shown in *Figure 2.67*.

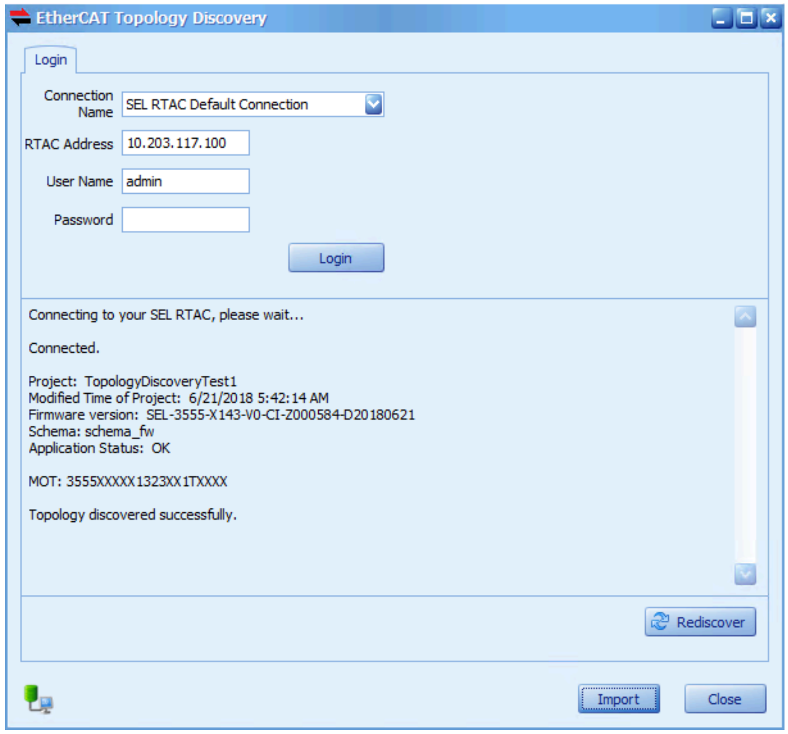


Figure 2.67 EtherCAT Topology Discovered

If an unsupported topology is discovered, the message `Invalid EtherCAT topology` will be displayed. You can correct the module or cable connections and then click **Rediscover** to run the discovery process again.

If your project contains a preconfigured or previously discovered EtherCAT network that you have already set up in this project (as shown in *Figure 2.68*), you may choose to overwrite it. This will overwrite any preconfigured EtherCAT or module settings with default values.

NOTE
Choosing to import settings from EtherCAT Discovery will overwrite any preconfigured EtherCAT network and module settings with default values. If you have existing settings that you have created, SEL recommends saving a copy of your project before overwriting the EtherCAT network.

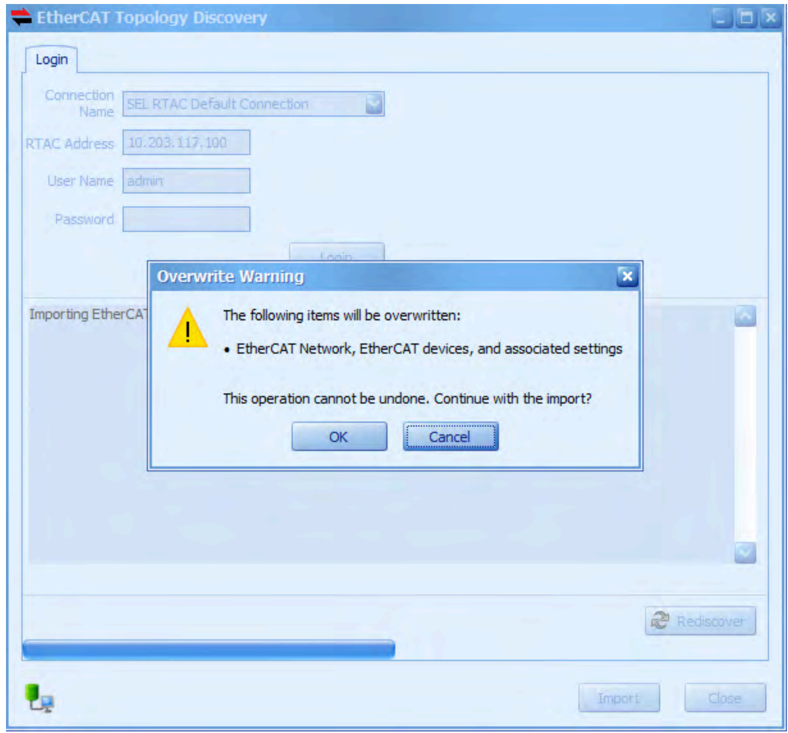


Figure 2.68 Overwrite Existing Network

Reading and Writing I/O Module Tags

Each power coupler or I/O module has a fixed set of inputs or outputs and diagnostic information available. The EtherCAT network updates these data automatically. The RTAC logic engine evaluates the data at the processing rate you set (100 ms default). Use the Tag Processor, as described in *Section 3: Tag Processor*, to apply the information in your custom IEC 61131-3 logic or map it into another protocol session.

Common Diagnostic Tags

Power couplers and I/O modules have a number of EtherCAT diagnostic tags that are common in each device. As shown in *Figure 2.69*, the diagnostic tags reside in the **Diagnostics** tab for each module. *Table 2.58* describes these tags. Diagnostic tags are always enabled and may not be renamed or aliased.

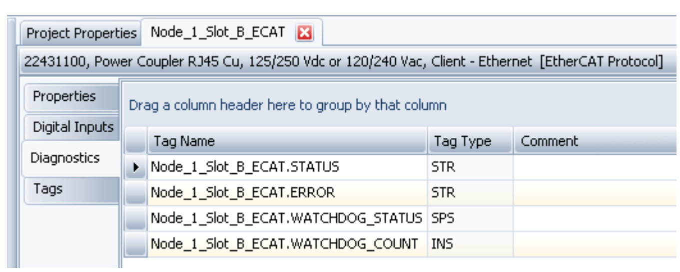


Figure 2.69 Power Coupler and I/O Module Diagnostic Tags

Table 2.58 Module Diagnostic Tag Descriptions

Parameter	Tag Type	Description	Possible Values
STATUS	STR	String representing the present operational status of the module	**Operational**—Normal operation **Non-Operational**—Module is not communicating on the EtherCAT network
ERROR	STR	String representing the network state for a module	**No Error**—Normal operation **Cable Error**—The network configuration has changed and does not match settings. This error could arise either from the movement of a server module or an Ethernet cable change.

Common Module Properties

Each power coupler and I/O module in your project will contain a set of properties that you can view in the **Properties** tab for a module, as shown in the top portion of *Figure 2.70*.

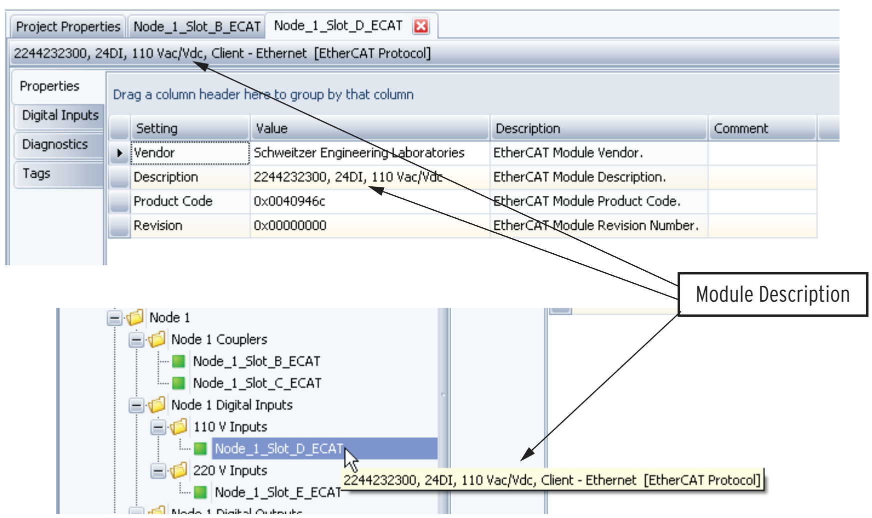


Figure 2.70 Power Coupler and I/O Module Properties

The module description includes detailed information about the function of the module. This information includes a partial model option string, module type (power coupler, digital input, digital output), and other operational information. There are three ways to see the module description. First, hover your cursor over the module name in the project view (see lower portion of *Figure 2.70*); second, hover your cursor over the title bar of an open module, and; third, hover your cursor over the description field of the **Properties** tab. If your module naming convention does not include a reference to the function of a module, use these viewing methods to quickly identify module types in the project.

SEL-2243 Power Coupler

The SEL-2243 Power Coupler provides two functions for an Axion node. First, the power coupler is a power supply with sufficient capacity to accommodate a complete node. Second, the power coupler provides internode EtherCAT network connections. When you add a power coupler to a project, the editor pane will display a tab for it, as shown in *Figure 2.69*. The previous section described module properties and diagnostics.

Power couplers have copper and fiber Ethernet ports available. The fiber ports enable greater isolation and increased physical separation of nodes. When you add a power coupler to an RTAC project, select the type of Ethernet port you are using. *Figure 2.71* illustrates this menu option.

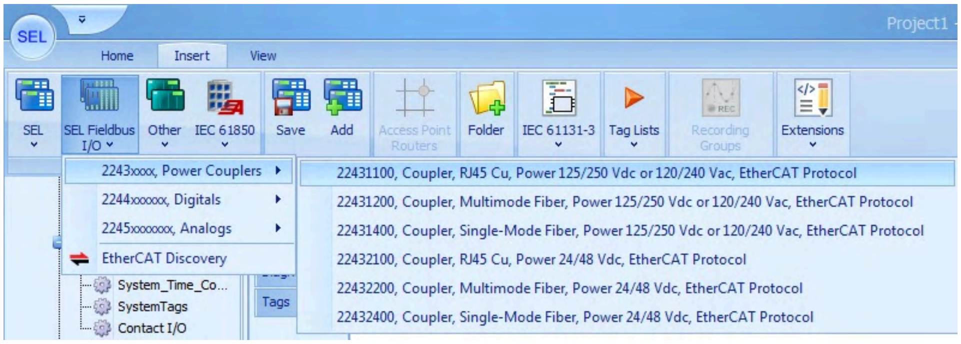


Figure 2.71 Choose Power Coupler Options

Each power coupler has one digital input that represents the state of the on-board power supply. The input will be of type Single Point Status (SPS) with a default name of <Module_Name>.HW_STATUS_001, which resides in the **Diagnostics** tab for the subject power coupler. All diagnostic and status tags for EtherCAT modules are enabled by default.

SEL-2244 24 Optoisolated Digital Input Module

The SEL-2244 Digital Input Module includes 18 common-return and six independent inputs. All of the inputs are dry contact type inputs. Use the **Insert** ribbon selection, as shown in *Figure 2.72*, to include an instance of this module in your project. Be careful to select the control voltage level you will be using for each module; the voltage is fixed for each module. You can configure an Axion node as necessary to have inputs that use a variety of control voltage types and levels.

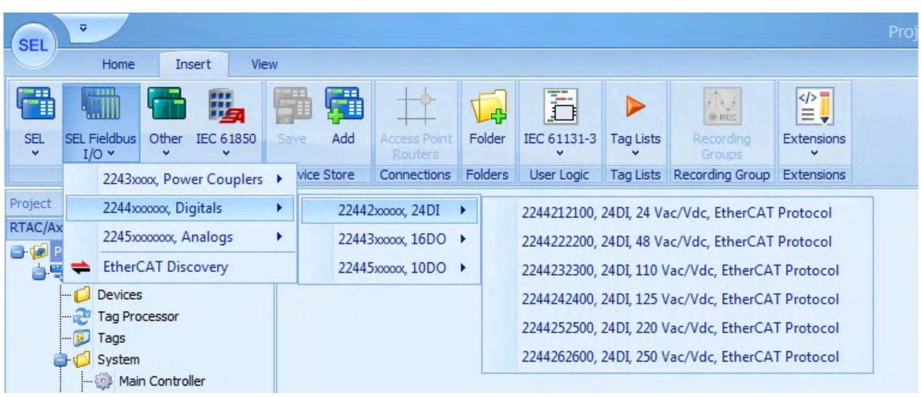
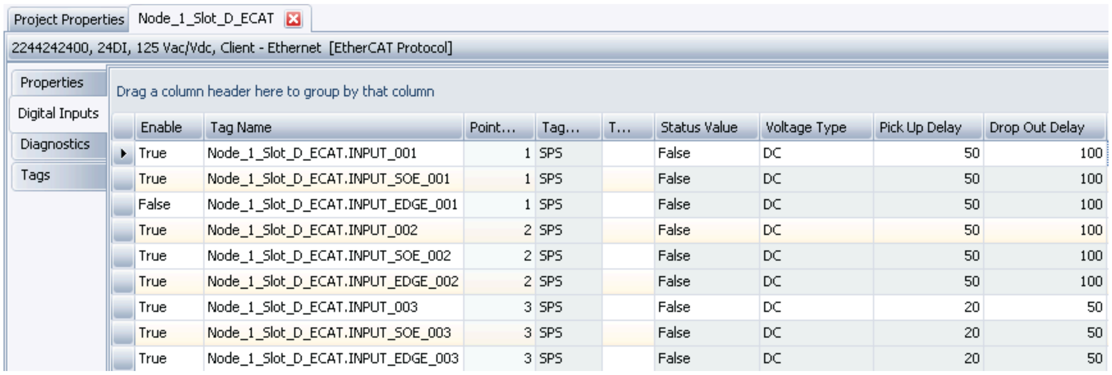


Figure 2.72 Insert an SEL-2244 Digital Input Module

Once you add a digital input module to a project, click the module name in the project view to open the connection editor. The Properties and Diagnostics tabs (see *Figure 2.73*) include the common EtherCAT module data discussed previously. The **Tags** tab lists all available and enabled tags from the module. The **Digital Inputs** tab lists the tags associated with the 24 inputs in the module. Each physical input is represented by three tags of data type SPS named

INPUT_*xxx*, INPUT_SOE_*xxx*, and INPUT_EDGE_*xxx*, where *xxx* refers to the physical input designation within a module. The following discussion describes the signal processing and function of each tag. Each input tag is enabled by default. Disable any unneeded tags to optimize processing resources.

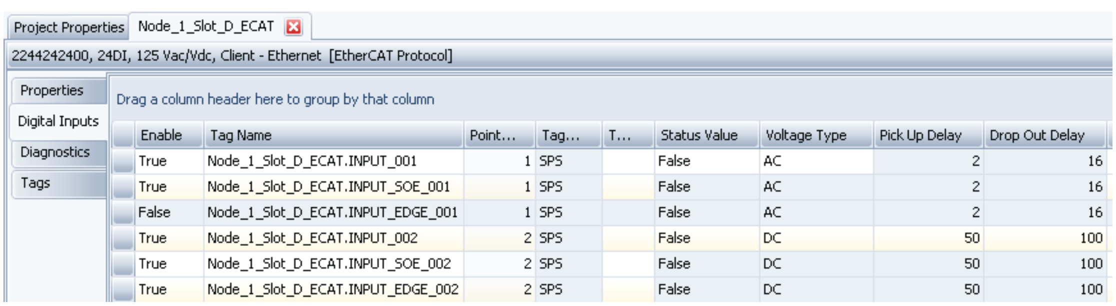


Enable	Tag Name	Point...	Tag...	T...	Status Value	Voltage Type	Pick Up Delay	Drop Out Delay
True	Node_1_Slot_D_ECAT.INPUT_001	1	SPS		False	DC	50	100
True	Node_1_Slot_D_ECAT.INPUT_SOE_001	1	SPS		False	DC	50	100
False	Node_1_Slot_D_ECAT.INPUT_EDGE_001	1	SPS		False	DC	50	100
True	Node_1_Slot_D_ECAT.INPUT_002	2	SPS		False	DC	50	100
True	Node_1_Slot_D_ECAT.INPUT_SOE_002	2	SPS		False	DC	50	100
True	Node_1_Slot_D_ECAT.INPUT_EDGE_002	2	SPS		False	DC	50	100
True	Node_1_Slot_D_ECAT.INPUT_003	3	SPS		False	DC	20	50
True	Node_1_Slot_D_ECAT.INPUT_SOE_003	3	SPS		False	DC	20	50
True	Node_1_Slot_D_ECAT.INPUT_EDGE_003	3	SPS		False	DC	20	50

Figure 2.73 Digital Inputs Tab for the SEL-2244 Digital Input Module

Digital Input Debounce

To comply with different control voltages, the SEL-2244 Digital Input Module offers dc debounce and ac debounce modes. You can use the voltage type setting in the **Digital Inputs** tab to select these modes on a per-input basis. If you change the setting for a given input from dc to ac and save the project, ACSELERATOR RTAC will configure all three tags associated with that physical input to the new setting. *Figure 2.74* illustrates this functionality after the voltage type setting for Node_1_Slot_D_ECAT.INPUT_001 changed from dc to ac. Changing the value from ac to dc and saving the project will have the reverse effect.



Enable	Tag Name	Point...	Tag...	T...	Status Value	Voltage Type	Pick Up Delay	Drop Out Delay
True	Node_1_Slot_D_ECAT.INPUT_001	1	SPS		False	AC	2	16
True	Node_1_Slot_D_ECAT.INPUT_SOE_001	1	SPS		False	AC	2	16
False	Node_1_Slot_D_ECAT.INPUT_EDGE_001	1	SPS		False	AC	2	16
True	Node_1_Slot_D_ECAT.INPUT_002	2	SPS		False	DC	50	100
True	Node_1_Slot_D_ECAT.INPUT_SOE_002	2	SPS		False	DC	50	100
True	Node_1_Slot_D_ECAT.INPUT_EDGE_002	2	SPS		False	DC	50	100

Figure 2.74 Changing Voltage Type Setting for Digital Input Module

If the control input voltage is dc, select **DC** for the voltage type. If the control input voltage is ac, select **AC** for the voltage type. In general, debounce refers to a qualifying time delay before processing the change of state of a digital input. Normally, this delay applies to the processing of the debounced input when used in IEC 61131-3 logic, and to the time-stamping of the tag for logging and SOE. We provide INPUT_*xxx* and INPUT_SOE_*xxx* as separate tags to account for two different processing intervals operating in the RTAC. First, the IEC 61131 logic engine performs logic on a user-settable interval between 4 ms and 1 second. Second, the EtherCAT network updates every 1 ms and buffers the input state and time stamp every millisecond in the INPUT_SOE_*xxx* tag, regardless of the logic engine processing interval. If you need a high-speed record for an input, then use the Tag Processor to log the INPUT_SOE_*xxx* tag.

Section 5: Web HMI and Reports provides detailed instructions on this feature. The INPUT_*xxx* tag does not buffer changes of state between logic processing intervals. It is a debounced value and always represents the last recorded state of an input. Use this tag in IEC 61131-3 programmable logic. The time stamp for INPUT_*xxx* will represent the last recorded change of state.

NOTE
AC debounce mode is processor intensive. Using a large number of AC debounced inputs is not recommended.

In some cases, it is also important to record the time of first assertion of an input. To this end, the SEL-2244 Digital Input Module provides the initial assertion time via the INPUT_EDGE_*xxx* tag. The module buffers this tag with a time stamp for each event. Include it in logging along with the INPUT_SOE_*xxx* tag as necessary. The INPUT_EDGE_*xxx* tags are automatically disabled and unavailable when you set the voltage type to ac mode.

DC Mode Processing (DC Control Voltage)

NOTE
In R118 and later, the AC/DC mode selection applies to an entire module rather than individual inputs. Converting from an older project with mixed settings will result in the first selection applying to the whole module.

Figure 2.75 shows the logic for the dc voltage type mode of operation, which we select through use of the voltage type setting shown in *Figure 2.74*. The Input_*xxx*, Input_SOE_*xxx*, and INPUT_EDGE_*xxx* tags share a debounce timer. Set the timer Pick Up Delay and Drop Out Delay for the INPUT_*xxx* tag and save the project; ACSELERATOR RTAC will automatically apply the same timer settings to all three tags. On assertion, INPUT_RAW_*xxx* (internal variable) starts Debounce Timer 1, producing INPUT_*xxx* and INPUT_SOE_*xxx* after the debounce time pickup delay.

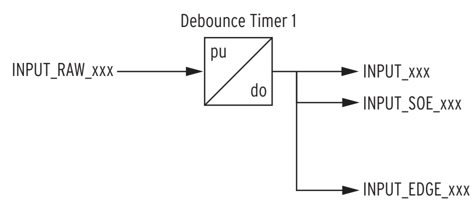


Figure 2.75 DC Mode Processing

Figure 2.76 shows a timing diagram when INPUT_RAW_*xxx* changes from the deasserted state to the asserted state. At the first assertion of INPUT_RAW_*xxx*, the following takes place in regards to the edge detection function:

- INPUT_EDGE_*xxx* asserts
- Debounce Timer 1 starts
- All edge changes are ignored

If you want to record the time of the first assertion of an input, be sure to enable logging for INPUT_EDGE_*xxx* in the Tag Processor. During the time when Debounce Timer 1 runs, the device ignores all edge changes with regards to the INPUT_EDGE_*xxx* tag. At the end of this timing period, the system evaluates the status of INPUT_RAW_*xxx* (either logical 0 or logical 1), and sets INPUT_EDGE_*xxx* to this value. In *Figure 2.76*, INPUT_RAW_*xxx* has a status of logical 1 at the expiration of the timer, and INPUT_EDGE_*xxx* remains at logical 1.

NOTE
INPUT_EDGE and INPUT_SOE tags are disabled by default in R118 and later, to optimize processing capability.

INPUT_*xxx* and INPUT_SOE_*xxx* assert only if INPUT_RAW_*xxx* remains asserted for the complete Pick Up Delay time of Debounce Timer 1. If INPUT_RAW_*xxx* deasserts at any point while Debounce Timer 1 is running, Debounce Timer 1 resets and starts timing from the beginning of the next rising edge.

The inverse operation applies when the input changes from the asserted state to the deasserted state, as shown in *Figure 2.77*.

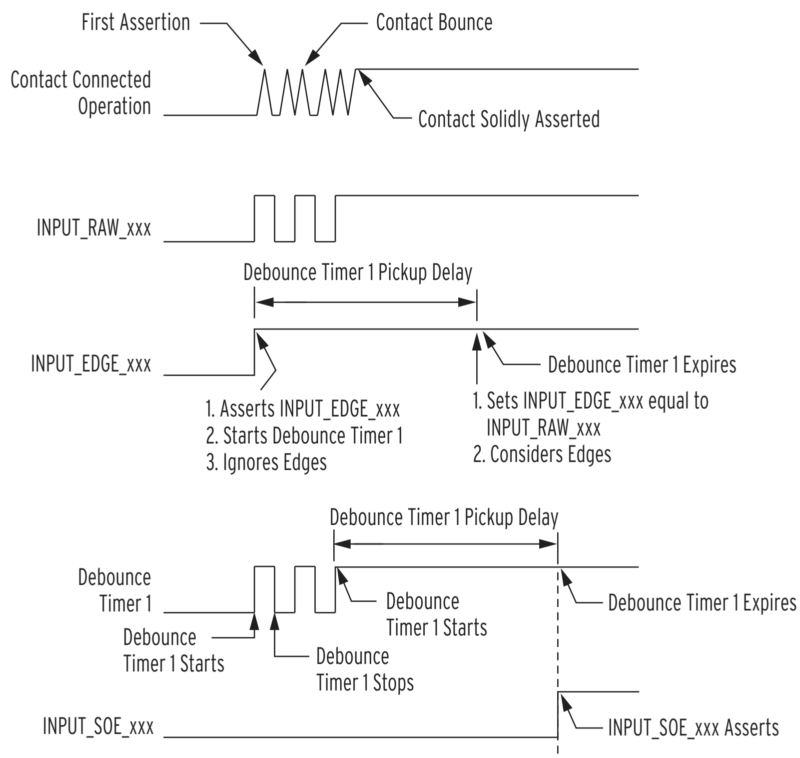


Figure 2.76 Timing Diagram When Input Changes From the Deasserted State to the Asserted State

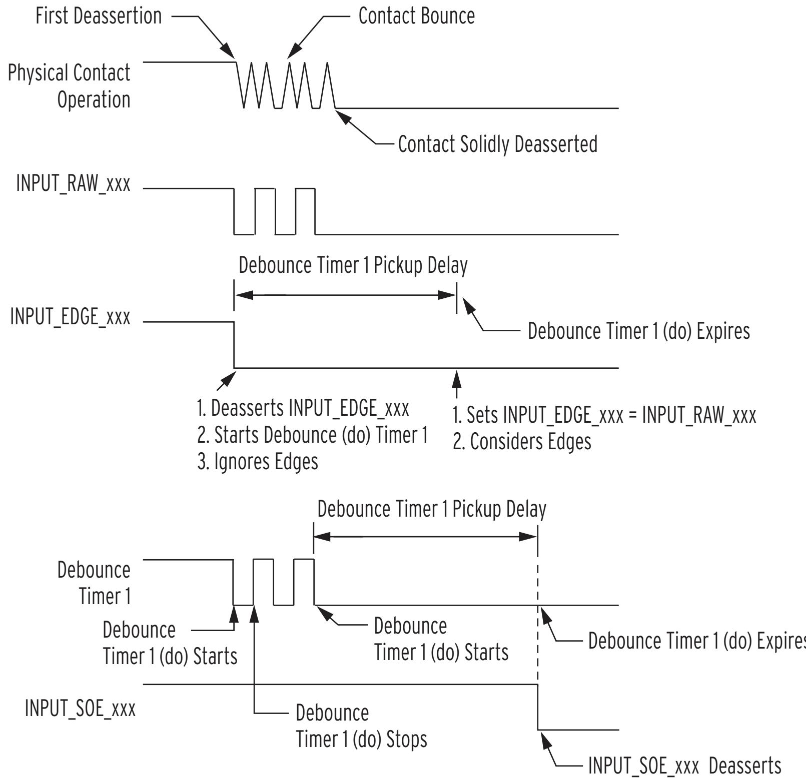


Figure 2.77 Timing Diagram When Input Changes From the Asserted State to the Deasserted State

AC Mode Processing (AC Control Voltage)

NOTE
In R118 and later, the AC/DC mode selection applies to an entire module rather than individual inputs. Converting from an older project with mixed settings will result in the first selection applying to the whole module.

Figure 2.78 shows the logic for the ac voltage type mode of operation, which you select through use of the voltage type setting shown in *Figure 2.74*. The Input_*xxx* and Input_SOE_*xxx* tags share a debounce timer. On assertion, INPUT_RAW_*xxx* (internal variable) starts the debounce timer, producing INPUT_*xxx* and INPUT_SOE_*xxx* after the debounce time delay. The ac mode differs from the dc mode in having only delayed time information available. There are also no adjustable time settings for the debounce timer in the ac mode: the Pick Up Delay is fixed at 2 ms, and the Drop Out Delay is fixed at 16 ms.

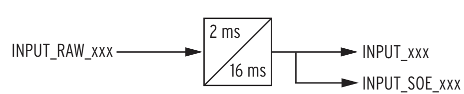


Figure 2.78 AC Mode Processing

Figure 2.79 shows a timing diagram for the ac mode of operation. On the rising edge of INPUT_RAW_*xxx*, the pickup timer starts timing (points marked 1 in *Figure 2.79*). If INPUT_RAW_*xxx* deasserts (points marked 2 in *Figure 2.79*) before expiration of the Pick Up Delay time, INPUT_*xxx* and INPUT_SOE_*xxx* will not assert, and they remain at logical 0. If, however, INPUT_RAW_xxx remains asserted for a period longer than the Pick Up Delay time, then both INPUT_*xxx* and INPUT_SOE_*xxx* assert to a logical 1.

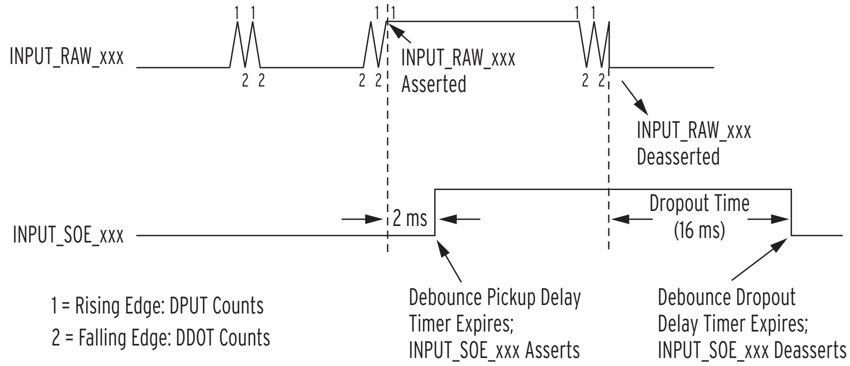


Figure 2.79 Timing Diagram for Debounce Timer Operation When Operating in AC Mode

Deassertion follows the same logic. On the falling edge of INPUT_RAW_*xxx*, the Drop Out Delay timer starts timing. If INPUT_RAW_*xxx* remains deasserted for a period longer than the Drop Out Delay setting, then INPUT_*xxx* and INPUT_SOE_*xxx* deassert to a logical 0.

Be aware that ac mode digital inputs require more processing time in the RTAC than dc mode digital inputs. As a general rule, you should not install more than two nodes full of ac digital inputs in a single Axion system. However, because of processor burden for other protocols and user logic, the best method to validate system performance is to test your project and verify that the EtherCAT message processing task does not regularly exceed 500 microseconds to run. You can check this by going online in ACSELERATOR RTAC, select **EtherCAT I/O Network** from the project tree view, and monitor the tags Client_Max_us and Client_Last_us. As discussed in *Task Cycle Time on page 6*, if you overload a task you will see poor performance in the web HMI or web server application. One way to reduce system burden is to distribute Axion I/O between multiple RTAC modules.

Creating a Counter From a Digital Input

You can use the input tags in any custom logic you want. Refer to the *Section 9: Custom Logic* for examples and *Appendix B: IEC 61131-3 Programming Reference* for IEC 61131-3 programming reference material. As a brief example, we will evaluate how to create a simple accumulator from a digital input. You could use this function to accumulate metering values or store the number of times an apparatus performs a specific sequence of operation. In this example, we used a CFC diagram, as shown in *Figure 2.80*. We use the stVal attribute of digital input INPUT_002 (refer to SPS data type in the IEC 61131-3 Programming Reference) in conjunction with a Rising Edge Trigger and an Incrementing Counter function block to create an output representing the number of times the input has changed from the deasserted state to the asserted state. Remember that debouncing of INPUT_*xxx* values occurs as we discussed in the previous section. Add automatic or manual counter reset logic as necessary.

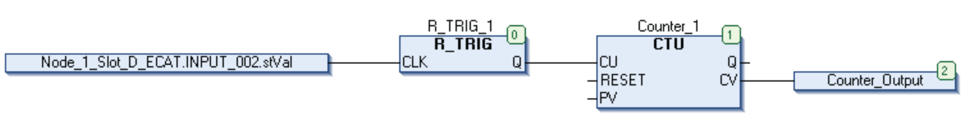


Figure 2.80 Creating a Counter Function From a Digital Input

SEL-2244 16 Digital Output and 10 Fast High-Current Digital Output Modules

The SEL-2244 Digital Output Module includes 16 independent standard contact outputs or 10 independent fast high-current contact outputs. You can order the module to have all Form A or all Form B contacts; they are not software selectable. Also, you can order a module that contains half Form A and half Form B outputs. Use the **Insert** ribbon selection, as shown in *Figure 2.81*, to include an instance of this module in your project. Be careful to select the output type you require for each module. You can configure an Axion node to have output modules with a variety of contact types.

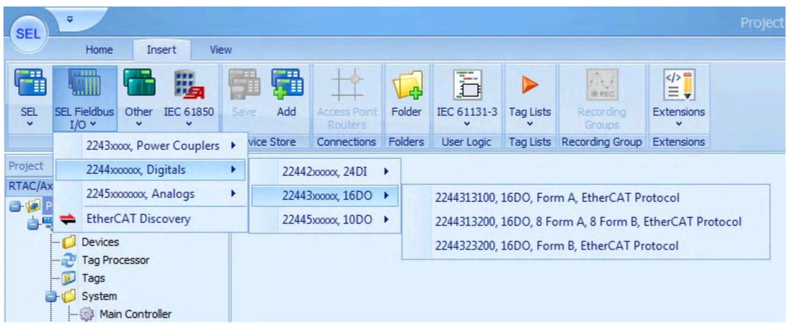


Figure 2.81 Insert an SEL-2244 Digital Output Module

Once you add a digital output module to a project, click the module name in the project view to open the connection editor. The **Properties** and **Diagnostics** tabs (see *Figure 2.73*) include the common EtherCAT module data discussed previously. The **Tags** tab lists all available and enabled tags from the module. Each of the 16 outputs in a module is controlled by an Input/Output Control (IOC) type tag. Refer to the IEC 61131-3 Programming Reference for detailed information on this tag type. Notice in *Figure 2.88* that each output will include a status, operSet, and operClear attribute. The physical output will assert when you set the operSet attribute. The physical output will deassert when you set the operClear attribute. The status attribute displays the latest commanded value for the output. Each output tag is enabled by default. Disable any unneeded tags to optimize processing resources.

Output Contact Pulse

EtherCAT Digital outputs and the optional SEL-3530 digital outputs execute pulse operations without custom logic. Enter pulse configuration data directly from custom programs or map the **TRIP**, **CLOSE**, or **PULSE** command from a DNP Binary Output by using the Tag Processor. When the operPulse.ctlVal is toggled from FALSE to TRUE, the Output contact asserts for the length defined by operPulse.pulseConfig.OnDur and deasserts for the length of time defined by operPulse.pulseConfig.offDur. The pulse repeats the number of times defined by operPulse.pulseConfig.numPls. Assertions and deassertions occur in multiples of the RTE Cycle. On and Off durations that are not multiples of the RTE cycle time execute on the RTE cycle after the transition occurs. For example, if the RTE cycle is set to 100 ms, the onDur is set to 97 ms, and the offDur is set to 103 ms, the output contact will assert for 100 ms and then deassert for 200 ms (see *Figure 2.82*).

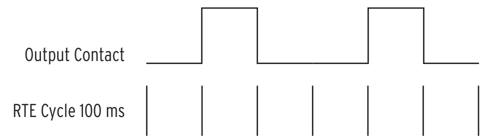


Figure 2.82 Pulse Train Duration

If an output is already asserted and a pulse command is received, the first state transition occurs at a time of onDur after the pulse command is processed (see *Figure 2.83*).

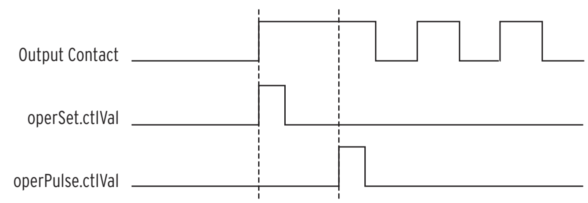


Figure 2.83 Pulsing an Asserted Output

If a pulse operation is in progress and the operClear.ctlVal is set to TRUE, the output deasserts and pulse operation ceases (see *Figure 2.84*).

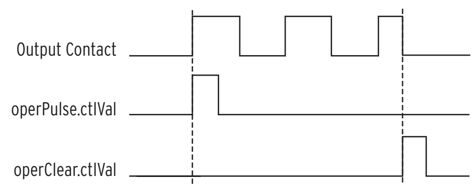


Figure 2.84 Interrupting a Pulse With a Clear Command

If a pulse operation is in progress and a new pulse command is received with a pulseConfig. onDur = 0, the output deasserts and pulse operation ceases (see *Figure 2.85*).

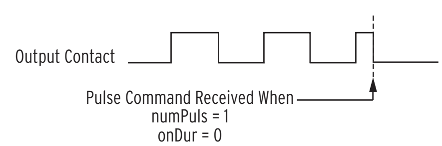


Figure 2.85 New Pulse Config onDur = 0

If a pulse operation is in progress and a new pulse command is received with a pulseConfig. numPls = 0, the asserted output remains asserted until onDur expires and pulse operation ceases. A deasserted output remains deasserted and pulse operation ceases (see *Figure 2.86*).

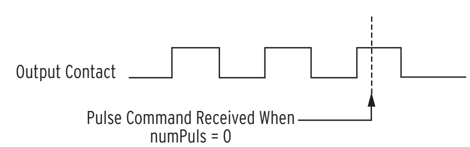


Figure 2.86 New Pulse Config numPls = 0

If a pulse command is received with a pulseConfig.offDur = 0, the output asserts for a time of onDur * numPls (see *Figure 2.87*).

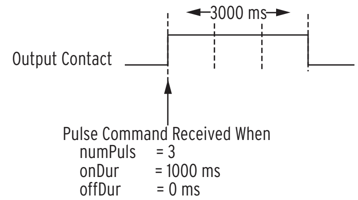
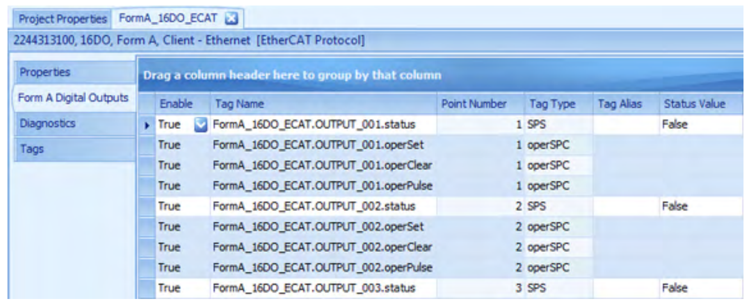


Figure 2.87 New Pulse Config offDur = 0

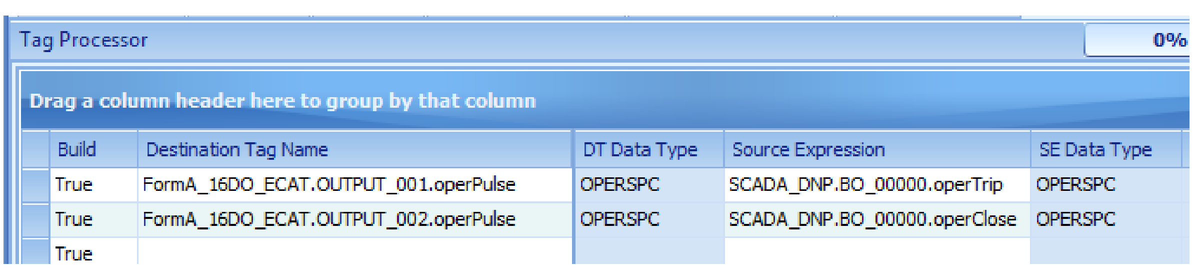


Enable	Tag Name	Point Number	Tag Type	Tag Alias	Status Value
True	FormA_16DO_ECAT.OUTPUT_001.status	1	SPS		False
True	FormA_16DO_ECAT.OUTPUT_001.operSet	1	operSPC		
True	FormA_16DO_ECAT.OUTPUT_001.operClear	1	operSPC		
True	FormA_16DO_ECAT.OUTPUT_001.operPulse	1	operSPC		
True	FormA_16DO_ECAT.OUTPUT_002.status	2	SPS		False
True	FormA_16DO_ECAT.OUTPUT_002.operSet	2	operSPC		
True	FormA_16DO_ECAT.OUTPUT_002.operClear	2	operSPC		
True	FormA_16DO_ECAT.OUTPUT_002.operPulse	2	operSPC		
True	FormA_16DO_ECAT.OUTPUT_003.status	3	SPS		False

Figure 2.88 Digital Output Module Tag Attributes

Creating Paired Outputs

You can use the output tags in any custom logic you want. Refer to *Section 9: Custom Logic* for examples and *Appendix B: IEC 61131-3 Programming Reference* for IEC 61131-3 programming reference material. As an example, we will evaluate how to combine two physical outputs in an SEL-2244 Digital Output Module into a trip-close pair when used in conjunction with a DNP binary output tag. For this example, the RTAC module is connected to the SEL-2244 Digital Output Module and also operates as a DNP server (slave). This example associates one binary output DNP tag's trip and close attributes with two physical outputs (refer to *Figure 2.89* for the tag processor example). Configure the operTrip.pulseConfig and operClose.pulseConfig attributes for the tag SCADA_DNP.BO.00000 the same, setting the onDur := 3000ms, offDur := 1000ms, and numPls := 1. When the RTAC receives a **TRIP** command from the DNP Client, OUTPUT_001 asserts for three seconds. When the RTAC receives a **CLOSE** command from the DNP Client, OUTPUT_002 asserts for three seconds.



Build	Destination Tag Name	DT Data Type	Source Expression	SE Data Type
True	FormA_16DO_ECAT.OUTPUT_001.operPulse	OPERSPC	SCADA_DNP.BO_00000.operTrip	OPERSPC
True	FormA_16DO_ECAT.OUTPUT_002.operPulse	OPERSPC	SCADA_DNP.BO_00000.operClose	OPERSPC
True				

Figure 2.89 Tag Processor for Trip-Close Pair Example

SEL-2245-2 16 DC Analog Input Module

The SEL-2245-2 DC Analog Input Module includes 16 inputs for measuring low-level dc signals. The inputs are user configurable in pairs to measure signals within ±20 mA, ±2 mA, or ±10 V ranges. Use the **Insert** ribbon as shown to add an instance of this module to your project. The displayed pop-up will allow you to change the module name if necessary. When finished, click **Insert**.

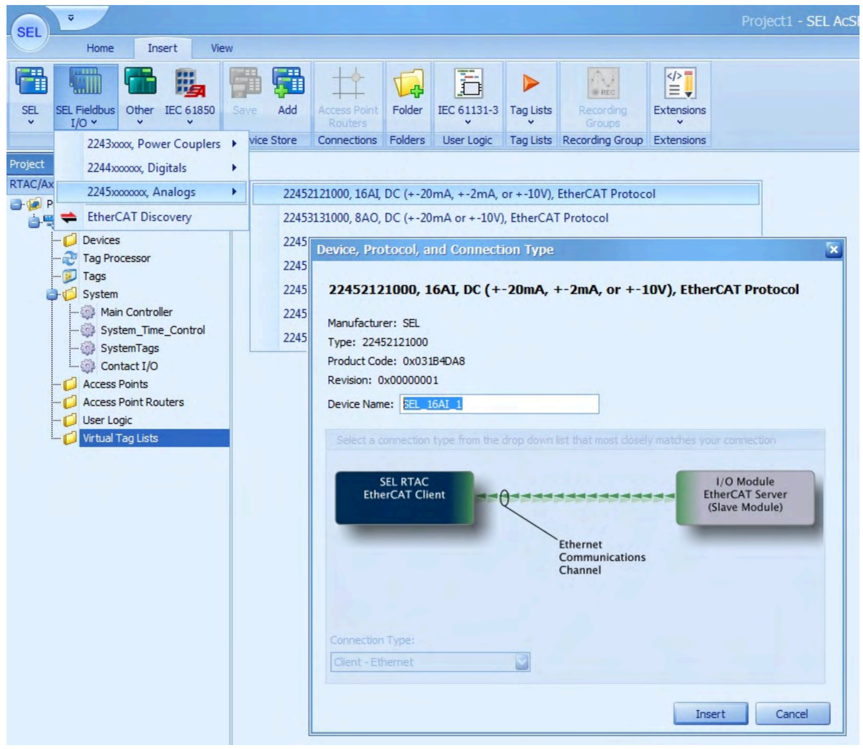
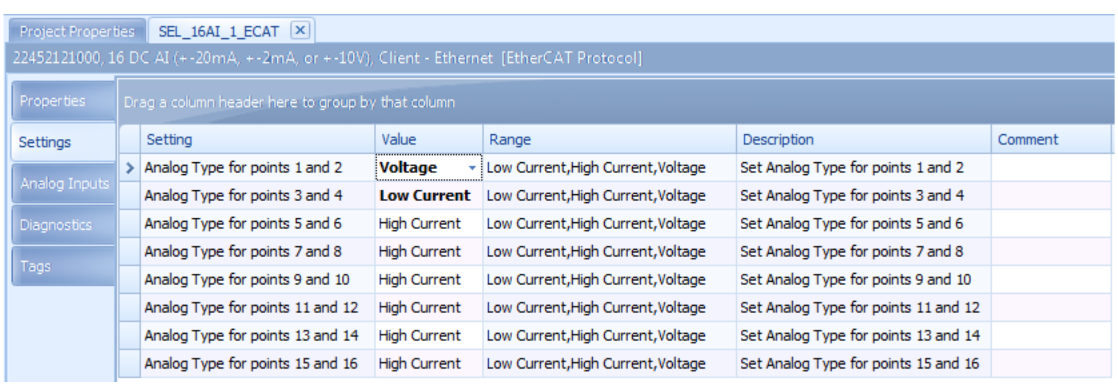


Figure 2.90 Insert an SEL-2245 DC Analog Input Module

Configure the Analog Input Module

In the project tree, select the module to open the module settings dialog window. Go to the **Settings** tab to select the input range of the analog inputs type (High Current, Low Current, and Voltage). See the specifications section for input specifications. Each input range selection is for a pair of inputs.



Setting	Value	Range	Description	Comment
Analog Type for points 1 and 2	Voltage	Low Current,High Current,Voltage	Set Analog Type for points 1 and 2	
Analog Type for points 3 and 4	Low Current	Low Current,High Current,Voltage	Set Analog Type for points 3 and 4	
Analog Type for points 5 and 6	High Current	Low Current,High Current,Voltage	Set Analog Type for points 5 and 6	
Analog Type for points 7 and 8	High Current	Low Current,High Current,Voltage	Set Analog Type for points 7 and 8	
Analog Type for points 9 and 10	High Current	Low Current,High Current,Voltage	Set Analog Type for points 9 and 10	
Analog Type for points 11 and 12	High Current	Low Current,High Current,Voltage	Set Analog Type for points 11 and 12	
Analog Type for points 13 and 14	High Current	Low Current,High Current,Voltage	Set Analog Type for points 13 and 14	
Analog Type for points 15 and 16	High Current	Low Current,High Current,Voltage	Set Analog Type for points 15 and 16	

Figure 2.91 Analog Input Range Selection

The **Analog Inputs** tab allows you to configure the range of the inputs, alarm points and sampling parameters.

Enable

When True, this tag will be included when the program is compiled. When False, it will not be included. Disable all tags not in use to optimize processing resources.

Tag Name

Programming designation of a resource.

Tag Alias

Input an alternative name to use in place of the Tag Name while programming (for example, SEL_16AI_1_ECAT.INPUT_VALUE_001 could be replaced by "coolant_temp").

Engineering Units High and Engineering Units Low

The real-world quantities that the analog signal represents: temperature, pressure, etc.

Physical High and Physical Low

The high and low signal levels expected.

Example settings for a –50degC to 150degC 4–20mA transducer would be as follows:
Engineering Units High: 150
Engineering Unit Low: –50
Physical High: 20
Physical Low: 4

The software will then create a scaling factor to relate the current value to a real world temperature; for example 6 mA = –25 degrees C.

There are four set points: **Low Alarm**, **Low Warning**, **High Alarm**, **High Warning**. These fields are of the same type as the Engineering Units: temperature, psi, etc. When the analog value is exceeded the corresponding tag is set.

Example:

Setting **Low Alarm** to 10 would set the SEL_16AI_1_ECAT.INPUT_LOW_ALARM_001 tag when Temp ≤ 10 degrees C

Setting **High Alarm** to 90 would set the SEL_16AI_1_ECAT.INPUT_HIGH_ALARM_001 tag when Temp ≥ 90 degrees C

Dead Band

The value is set in engineering units and is the amount the analog value has to change before a change is reported.

Filter Type

None	No filters are applied. This will yield the fastest signal response.
Filter A	Slower response than None but has higher noise immunity. (See Specifications.)
Filter B	Medium response and medium immunity. (See Specifications.)
Filter C[a]	Slowest response but has the highest immunity. (See Specifications.)

Rate of Change
The amount the analog value has to change in the time specified (Rate of Change Sample Duration) before the rate of change alarm tag is set. Rate of Change is in engineering units and Rate of Change Sample Duration is in milliseconds.

[a]Filter C is only supported in SEL-2245-2 firmware R102 or later.

Diagnostics

STATUS	String representing the present operational status of the module.
Operational	Normal Operation
Non-Operational	Module is not communicating on the EtherCAT network.
ERROR	String representing the network state for a module.
No Error	Normal Operation
Cable Error	The network configuration has changed and does not match settings. This error could arise either from the movement of a server module or an Ethernet cable change.

Tags

INPUT_VALUE_*xxx* is of type MV and the value is of type REAL.[a]
All other tags are of type SPS and the value is of type BOOL
INPUT_HIGH_ALARM_*xxx*
INPUT_HIGH_WARN_*xxx*
INPUT_LOW_ALARM_*xxx*
INPUT_LOW_WARN_*xxx*
INPUT_RATE_CHANGE_ALARM_*xxx*

[a]xxx refers to the physical input designation within a module.

All alarm tags will return to false on the next cycle after an alarm condition is no longer present.

Waveform Recording

NOTE
Waveform reading is only supported in SEL-2245-2 firmware R102 or later.

Waveforms will be recording in COMTRADE format with an accompanying configuration file. File names are stored as follows:

StationName_ModuleID_Protocol_YYYYMMDD_HHMMSSMSC.yyy

where:

yyy is one of two file name extensions: CFG for configuration files or DAT for data files.

Protocol is the name of the protocol used (e.g., ECAT).

StationName is defined by the RTAC hostname, which is configurable through the web interface. When the RTAC hostname is changed, the EtherCAT network must be disabled and re-enabled for the new StationName to appear in the file name of waveform records.

A maximum of 1024 COMTRADE events can be stored in the RTAC with the oldest record being deleted when a new event is triggered and 1024 is exceeded. The SEL-2245-2 can record two back-to-back events. A new record can be triggered as soon as the current one has finished recording. A trigger condition that occurs before the current recording finishes is ignored. These records are first stored locally to prevent data loss and then transferred to the RTAC for downloading via the built-in web server.

Inputs

The following analog inputs are recorded in the COMTRADEfiles generated by the SEL-2245-2 modules:

AI1–AI16

Downloading Events Via the Web Interface

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **https://** followed by the RTAC Ethernet address.

Step 4. Log in to the RTAC web interface.

Step 5. Click on **Event Collection** under **Reports** in the navigation pane on the left.

Step 6. Sort events by selecting a time frame from **Timestamp Query** and the **Device Filter**.

Step 7. Select the desired events and click **Download**. You can also delete events by clicking **Delete**.

Table 2.59 Waveform Settings

Setting	Description
VoltageInputConfiguration	Voltage input configuration
FrequencyNominal	Nominal system frequency
OscillographyCaptureRate	Sampling rate: 1 kHz
OscillographyRecordLength	Total record length: 0.5–144 seconds
OscillographyPreTriggerLength	Pre-trigger record length: 0.05 seconds minimum to a maximum of (record length – 0.05) seconds
StationName	RTAC hostname
ModuleID	Module identifier (module name in ACSELERATOR RTAC)
Module_name.Event_Trigger	Boolean input that triggers waveform recording in the module and only responds to rising edges

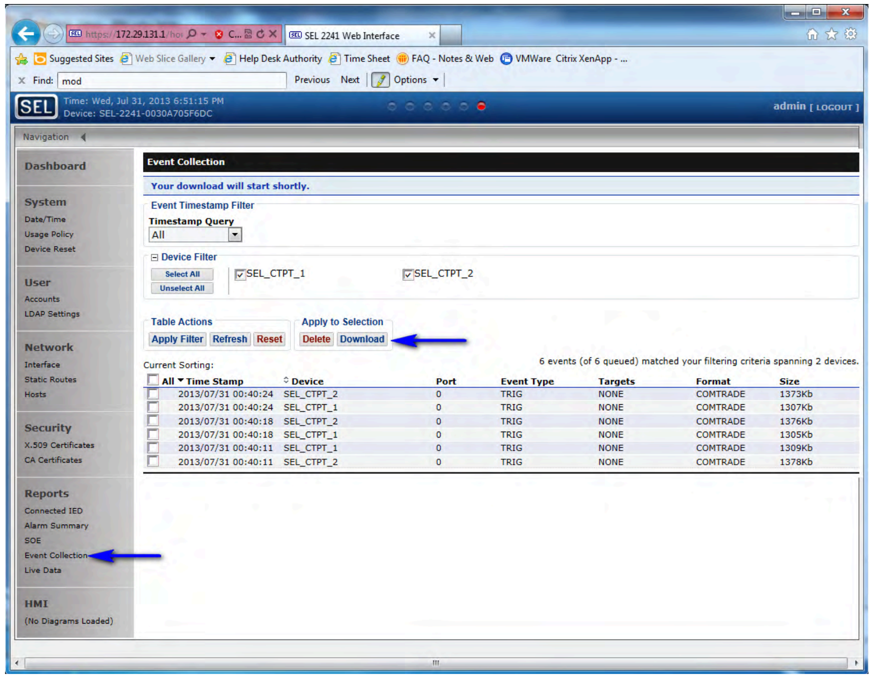


Figure 2.92 Waveform Record Retrieval

Viewing Waveforms and Event Files

Events files created by the SEL-2245-2 are in COMTRADE format. Event viewing software such as SEL-5601-2 SYNCHROWAVE® Event Software is required to open these files.

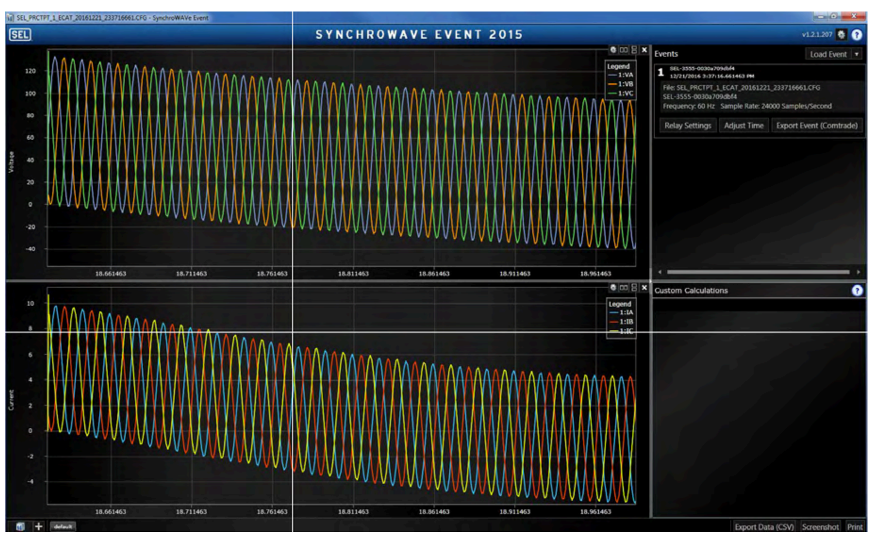


Figure 2.93 SYNCHROWAVE Event Display Waveform

Field Calibration Procedure

Step 1. Turn the Axion with the SEL-2245-2 on and allow it to warm up for a few minutes.

Step 2. Set the analog inputs for each analog channel to the desired range (e.g., 4–20 mA), using the value on the **Settings** tab and **Physical High**, and **Physical Low** on the **Analog Input** tab. Set **Engineering Units High** equal to **Physical High** and **Engineering Units Low** equal to **Physical Low**.

Step 3. Using a calibrated source, drive the signal line from the transducer end to the low value (for example, 4 mA).

Record ten measurements from the analog input value, then calculate the average of the ten measurements by adding the ten values algebraically, and dividing the sum by ten. This is the actual low value (for example, 3.9 mA).

Step 4. Enter this value in **Physical Low**.

Step 5. Drive the line to the high value (for example, 20 mA).

Step 6. Repeat *Step 3*.

Step 7. This is the actual high value (for example, 20.012 mA). Enter this value in **Physical High**.

Step 8. Set **Engineering Units High** and **Engineering Units Low** to desired values (for example, –50F to +150F).

SEL-2245-22 Four Analog Input Extended Range Module

The SEL-2245-22 Analog Input Extended Range Module includes four inputs for measuring high-level dc or ac signals. Refer to *Specifications in the SEL-2240 Axion Instruction Manual* for details on the input range for DC or AC Mode. Use the **Insert** ribbon as shown to add an instance of this module to your project. The displayed popup will allow you to change the module name if necessary. When finished, click **Insert**.

NOTE
AC metering mode was added to the SEL-2245-22 module firmware in R101 and to the RTAC in R141.

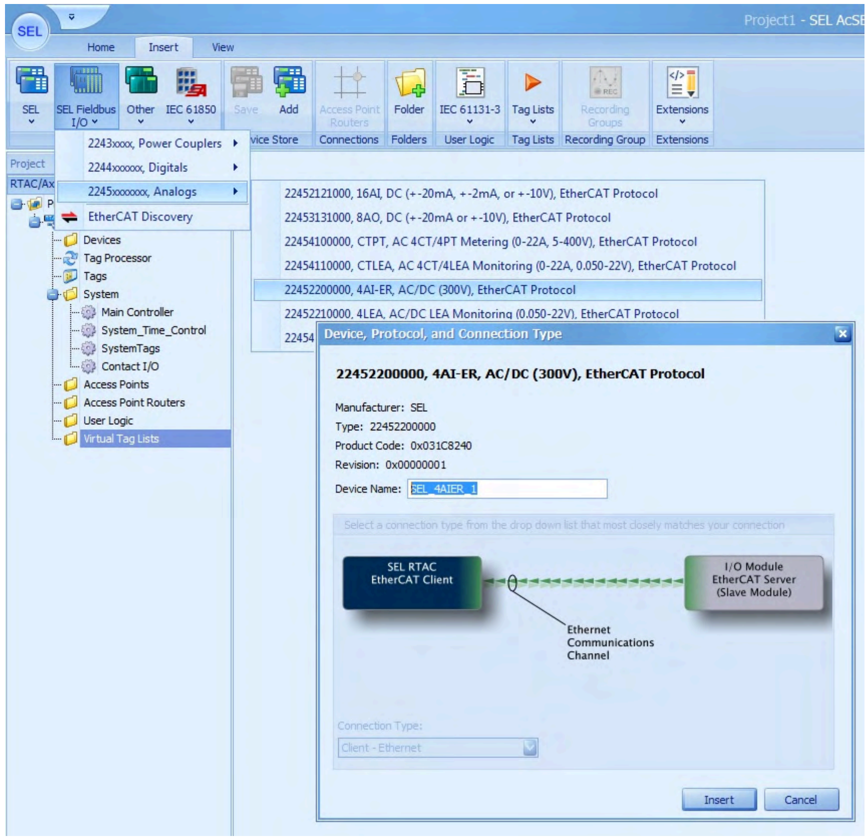


Figure 2.94 Insert an SEL-2245-22 DC Analog Input Extended Range Module

Configuring the Analog Input Module (DC Mode)

In the **Settings** tab, select DC for the Input Mode. This will enable the dc analog measurements and the associated settings.

The **Settings** tab also includes voltage parameter settings for AC Mode as well as Oscillography and Synchrophasor settings.

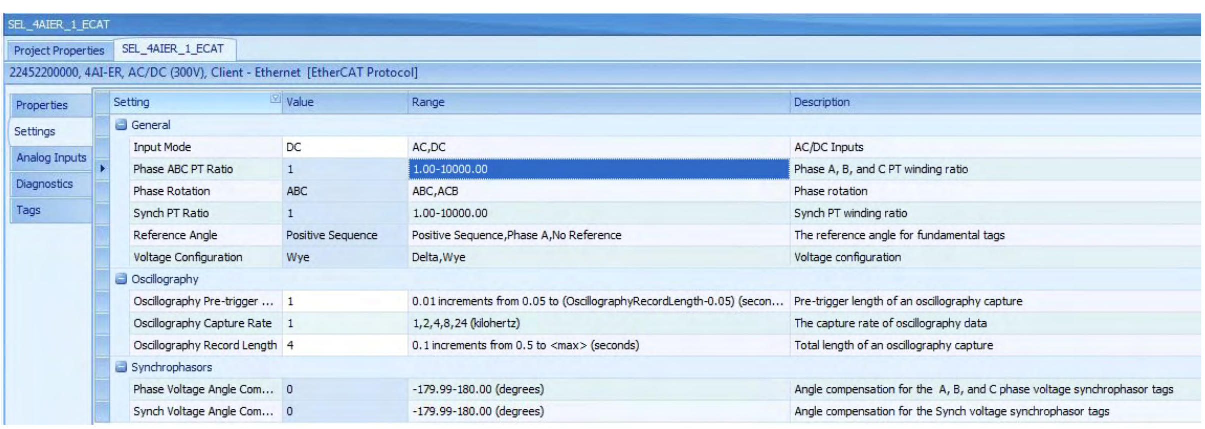


Figure 2.95 SEL-2245-22 Settings Tab (DC Mode)

The **Analog Inputs** tab allows you to configure the range of the inputs, alarm points and sampling parameters. The following table shows the settings of the **Analog Inputs** tab in DC Mode.

Enable	
When True, this tag will be included when the program is compiled. When False, it will not be included. Disable all tags not in use to optimize processing resources.	
Tag Name	
Programming designation of a resource.	
Tag Alias	
Input an alternative name to use in place of the Tag Name while programming (for example, SEL_4AIER_1_ECAT.INPUT_VALUE_001 could be replaced by "coolant_temp").	
Engineering Units High and Engineering Units Low	
The real-world quantities that the analog signal represents: temperature, pressure, etc.	
Physical High and Physical Low	
The high and low signal levels expected. Example settings: Engineering Units: 300 Engineering Units Low: 0 Physical High: 300 Physical Low: 0 The software will then create a scaling factor to relate the current value to a real-world value.	

There are four set points: **Low Alarm**, **Low Warning**, **High Alarm**, **High Warning**. These fields are of the same type as the Engineering Units: voltage, temperature, psi, etc. When the analog value is exceeded the corresponding tag is set.

Example:

Setting **Low Alarm** to 100 would set the SEL_4AIER_1_ECAT.INPUT_LOW_ALARM_001 tag when Volts ≥ 100 Vdc

Setting **High Alarm** to 130 would set the SEL_4AIER_1_ECAT.INPUT_HIGH_ALARM_001 tag when Volts ≥ 130 Vdc

Dead Band	
The value is set in engineering units and is the amount the analog value has to change before a change is reported.	
Filter Type	
None	No filters are applied. This will yield the fastest signal response.
Filter A	Slower response than None but has higher noise immunity. (See Specifications.)
Filter B	Medium response and medium immunity. (See Specifications.)
Filter C	Slowest response but has the highest immunity. (See Specifications.)
Rate of Change	
The amount the analog value has to change in the time specified (Rate of Change Sample Duration) before the rate of change alarm tag is set. Rate of Change is in engineering units and Rate of Change Sample Duration is in milliseconds.	

Diagnostics

STATUS	String representing the present operational status of the module.
Operational	Normal Operation
Non-Operational	Module is not communicating on the EtherCAT network.
ERROR	String representing the network state for a module.
No Error	Normal Operation
Cable Error	The network configuration has changed and does not match settings. This error could arise either from the movement of a server module or an Ethernet cable change.

Tags

INPUT_VALUE_*xxx* is of type MV and the value is of type REAL.[a]
All other tags are of type SPS and the value is of type BOOL
INPUT_HIGH_ALARM_*xxx*
INPUT_HIGH_WARN_*xxx*
INPUT_LOW_ALARM_*xxx*
INPUT_LOW_WARN_*xxx*
INPUT_RATE_CHANGE_ALARM_*xxx*

[a] xxx refers to the physical input designation within a module.

All alarm tags will return to false on the next cycle after an alarm condition is no longer present.

Configuring the Analog Input Module (AC Mode)

In the **Settings** tab, select AC for the Input Mode. This will enable the ac metering inputs and its associated settings.

The **Settings** tab also includes Voltage parameter settings for DC Mode as well as Oscillography and Synchrophasor settings.

Figure 2.96 SEL-2245-22 Settings Tab (AC Mode)

The **Analog Inputs** tab allows you to configure the range of the inputs, alarm points and sampling parameters. *Table 2.60* shows the settings of the **Analog Inputs** tab in AC Mode.

NOTE

RMS voltage quantities have units of kilovolts and Fundamental voltage quantities are in units of volts.

Table 2.60 Analog Inputs Settings

Setting	Configuration Options
Phase Rotation	ABC or ACB
Voltage Configuration	Configure inputs for three-wire Delta or four-wire Wye
Phase ABC PT ratio	1.00–10000.00
Synch PT Ratio	1.00–10000.00
Reference Angle	Select reference for Fundamental Vector quantities
Oscillography Capture Rate	Capture rate of oscillography data
Oscillography Record Length	Length of oscillography capture. Capture rate affects maximum record length.
Oscillography Pre-Trigger Length	Length of time before trigger to start capture
Station Name	1–255 characters
Voltage Phasor Data Set	All, Phase, Positive Sequence, None
PMU Id	1–65534
Global PMU Id	$0–2^{128 - 1}$

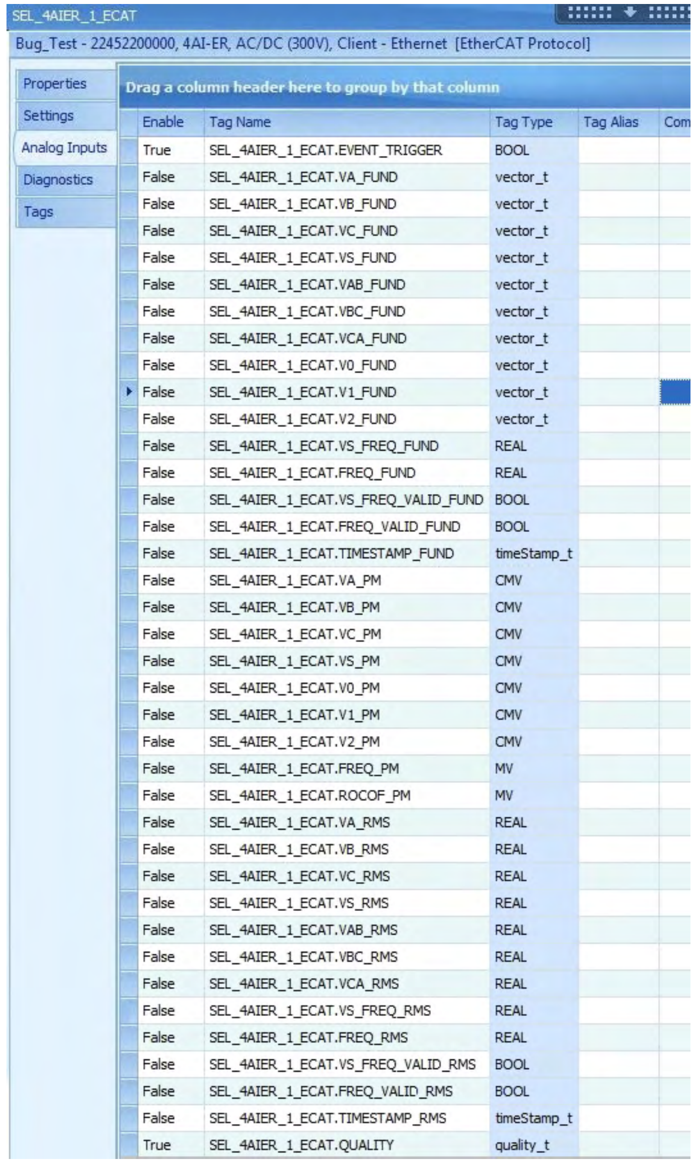


Figure 2.97 AC Mode Tag Settings

Analog Input	Description
Enable	When True, this tag will be included when the program is compiled. When False it will not be included. Disable all tags not in use to optimize processing resources. If there are too many analog data on the EtherCAT backplane, you will receive a message that the bandwidth has been exceeded (shown in *Figure 2.113*).
Tag Name	Programming designation of a resource.
Tag Alias	Input an alternative name to use in place of the Tag Name while programming (for example, SEL_CTPT_1_ECAT.IA_FUND could be replaced by IA_Bus1).

Waveform Recording

Waveforms will be recording in COMTRADE format with an accompanying configuration file. File names are stored as follows:

StationName_ModuleID_Protocol_YYYYMMDD_HHMMSSMSC.yyy

where:

yyy	is one of two file name extensions: CFG for configuration files or DAT for data files.
Protocol	is the name of the protocol used (e.g., ECAT).
StationName	is defined by the RTAC hostname, which is configurable through the web interface. When the RTAC hostname is changed, the EtherCAT network must be disabled and re-enabled for the new StationName to appear in the file name of waveform records.

A maximum of 1024 COMTRADE events can be stored in the RTAC with the oldest record being deleted when a new event is triggered and 1024 is exceeded. The SEL-2245-22 can record two back-to-back events. A new record can be triggered as soon as the current one has finished recording. A trigger condition that occurs before the current recording finishes is ignored. These records are first stored locally to prevent data loss and then transferred to the RTAC for downloading via the built-in web server.

Inputs

The following analog inputs are recorded in the COMTRADE files generated by the SEL-2245-22 modules in DC Mode:

AI1–AI4

The following analog inputs are recorded in the COMTRADE files generated by the SEL-2245-22 modules in AC Mode:

VA, VB, VC, VS (Wye)

VAB, VBC, VCA (Delta)

Downloading Events Via the Web Interface

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **https://** followed by the RTAC Ethernet address.

Step 4. Log in to the RTAC web interface.

Step 5. Click on **Event Collection** under **Reports** in the navigation pane on the left.

Step 6. Sort events by selecting a time frame from **Timestamp Query** and the **Device Filter**.

Step 7. Select the desired events and click **Download**. You can also delete events by clicking **Delete**.

Table 2.61 Waveform Settings

Setting	Description
VoltageInputConfiguration	Voltage input configuration
FrequencyNominal	Nominal system frequency
OscillographyCaptureRate	Sampling rate: 1, 2, 4, 8, 24 kHz; software-selectable
OscillographyRecordLength	Total record length: 6 seconds at 24 kHz 18 seconds at 8 kHz 36 seconds at 4 kHz 72 seconds at 2 kHz 144 seconds at 1 kHz
OscillographyPreTriggerLength	Pre-trigger record length: 0.05 seconds minimum to a maximum of (record length – 0.05) seconds
PTRatio	Phase A, B, C PT winding ratio
PTRatioS	Synch PT winding ratio
StationName	RTAC hostname
ModuleID	Module identifier (module name in ACSELERATOR RTAC)
module_name.Event_Trigger	Boolean input that triggers waveform recording in the module and only responds to rising edges

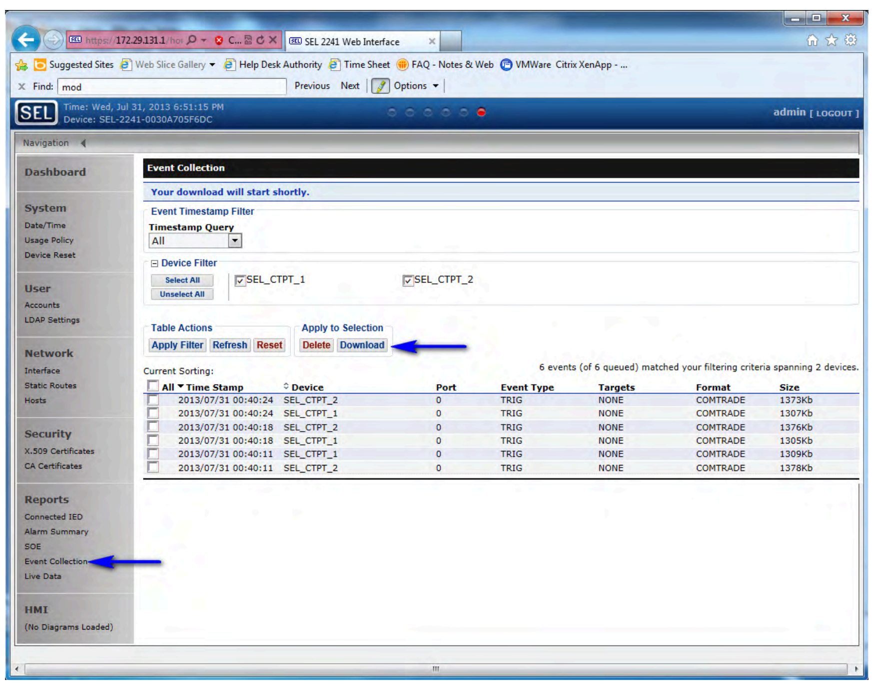


Figure 2.98 Waveform Record Retrieval

Viewing Waveforms and Event Files

Events files created by the SEL-2245-22 are in COMTRADE format. Event viewing software such as SYNCHROWAVE Event is required to open these files.

Figure 2.99 SYNCHROWAVE Event Display Waveform

Field Calibration Procedure

Step 1. Turn the Axion with the SEL-2245-22 on and allow it to warm up for a few minutes.

Step 2. Set the analog inputs for each analog channel to the desired range (e.g., 0–300 V), using the value on the **Physical High**, and **Physical Low** on the **Analog Input** tab. Set **Engineering Units High** equal to **Physical High** and **Engineering Units Low** equal to **Physical Low**.

Step 3. Using a calibrated source, drive the signal line from the transducer end to the low value (for example, 0 V).

Step 4. Record ten measurements from the analog input value, then calculate the average of the ten measurements by adding the ten values algebraically, and dividing the sum by ten. This is the actual low value (for example, 0.01 V).

Step 5. Enter this value in **Physical Low**.

Step 6. Drive the line to the high value (for example, 300 V).

Step 7. Repeat *Step 4*.

Step 8. This is the actual high value (for example, 300.1 V). Enter this value in **Physical High**.

Step 9. Set **Engineering Units High** and **Engineering Units Low** to desired values (for example, 0 V to 300 V).

SEL-2245-221 Low-Voltage (LEA) Monitoring Module

The SEL-2245-221 Low-Voltage (LEA) Monitoring Module (4 LEA) includes four low-voltage inputs with isolated returns for measuring ac signals. Refer to *Specifications in the SEL-2240 Axion Instruction Manual* for details on the input range. Use the **Insert** ribbon as shown in *Figure 2.104* to add an instance of this module to your project. The displayed pop-up will allow you to change the module name if necessary. When finished, click **Insert**.

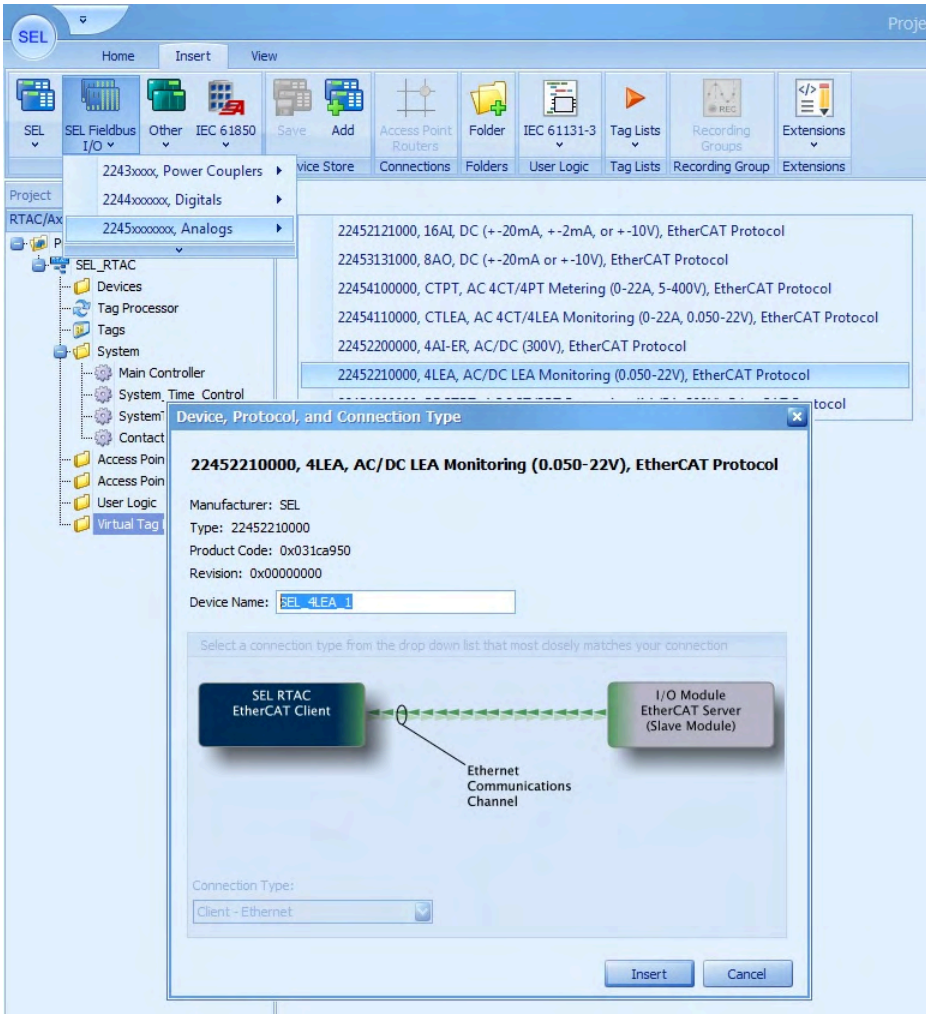


Figure 2.100 Insert an SEL-2245-221 4 LEA Module

Configuring the LEA Monitoring Module

In the project tree, select the module to open the module settings window. Click the **Settings** tab to configure the module.

NOTE

Only individual tags enabled per the LEA Monitoring Module are included in the EtherCAT message frame to reduce bandwidth. This allows for the use of more modules if fewer tags are needed. The EtherCAT connection tab shows the amount of bandwidth used by your current configuration.


SEL_4LEA_1_ECAT

22452210000, 4LEA, AC/DC LEA Monitoring (0.050-22V), Client - Ethernet [EtherCAT Protocol]

Properties | Settings | Analog Inputs | Diagnostics | Tags

Setting	Value	Range	Description
General			
Input Mode	AC	AC,DC	AC/DC Inputs
Phase Rotation	ABC	ABC,ACB	Phase rotation
Phase ABC PT Ratio	35	1.00-10000.00	Phase A, B, and C PT winding ratio
Synch PT Ratio	35	1.00-10000.00	Synch PT winding ratio
Voltage Configuration	Wye	Delta,Wye	Voltage configuration
Reference Angle	Positive Sequence	Positive Sequen...	The reference angle for fundamental tags
LEA			
VA Voltage Magnitude Com...	1	0.500-50.000	Magnitude Compensation for the VA Voltage Channel
VB Voltage Magnitude Com...	1	0.500-50.000	Magnitude Compensation for the VB Voltage Channel
VC Voltage Magnitude Com...	1	0.500-50.000	Magnitude Compensation for the VC Voltage Channel
VS Voltage Magnitude Com...	1	0.500-50.000	Magnitude Compensation for the VS Voltage Channel
VA Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VA Voltage channel
VB Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VB Voltage channel
VC Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VC Voltage channel
VS Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VS Voltage channel
Oscillography			
Oscillography Capture Rate	1	1,2,4,8,24 (kilo...	The capture rate of oscillography data
Oscillography Record Length	4	0.1 increments ...	Total length of an oscillography capture
Oscillography Pre-trigger L...	1	0.01 increments...	Pre-trigger length of an oscillography capture
Oscillography Channel Units	Volts	Volts,Amps	Oscillography Channel Units for COMTRADE Configuration File
PMU			
Station Name	SEL_4LEA_1_ECAT	1-255 (characte...	The PMU station name
PMU Id	1	1-65534	Identifier for this PMU
Voltage Phasor Data Set	Phase	All,Phase,Positi...	Voltage Phasor channels for PMU data set
Synchrophasors			
Phase Voltage Angle Comp...	0	-179.99-180.00...	Angle compensation for the A, B, and C phase voltage synchrophasor tags
Synch Voltage Angle Comp...	0	-179.99-180.00...	Angle compensation for the Synch voltage synchrophasor tags

Figure 2.101 LEA Module Settings

Setting	Configuration Options
Phase Rotation	ABC or ACB
Voltage Configuration	Configure inputs for three-wire Delta or four-wireWye
Phase ABC PT Ratio	1.00–10000.00
Synch PT Ratio	1.00–10000.00
Enable RMS Tags	Shows rms quantities in the **Analog Inputs** tab
Enable Fundamental Tags	Shows Fundamental quantities in the **Analog Inputs** tab
Reference Angle	Select reference for Fundamental Vector quantities
Oscillography Capture Rate	Capture rate of oscillography data
Oscillography Record Length	Length of oscillography capture. Capture rate affects maximum record length (see *Specifications* in the *SEL-2240 Instruction Manual*).
Oscillography Pre-Trigger Length	Length of time before trigger to start capture
VA Voltage Magnitude Compensation	0.500 to 1.500
VB Voltage Magnitude Compensation	0.500 to 1.500

Setting	Configuration Options
VC Voltage Magnitude Compensation	0.500 to 1.500
VS Voltage Magnitude Compensation	0.500 to 1.500
VA Voltage Angle Compensation	–10.00 to 10.00
VB Voltage Angle Compensation	–10.00 to 10.00
VC Voltage Angle Compensation	–10.00 to 10.00
VS Voltage Angle Compensation	–10.00 to 10.00
Phase Voltage Angle Compensation	–179.99 to 180.00
Synch Voltage Angle Compensation	–179.99 to 180.00

NOTE
RMS voltage quantities have units of kilovolts and fundamental voltage quantities are in units of volts.

Figure 2.102 LEA Tag Enable Settings

Analog Input	Description
Enable	When True, this tag will be included when the program is compiled. When False, it will not be included. Disable all tags not in use to optimize processing resources. If there are too many analog data on the EtherCAT backplane, you will receive a message that the bandwidth has been exceeded (shown in *Figure 2.103*).
Tag Name	Programming designation of a resource.
Tag Alias	Input an alternative name to use in place of the Tag Name while programming (for example, SEL_CTLEA_1_ECAT.IA_FUND could be replaced by IA_Bus1).

NOTE
If you receive the message "EtherCAT Bandwidth Exceeded," you will need to disable a number of RMS or Fundamental tags on one or more LEA modules to reduce the bandwidth in use.

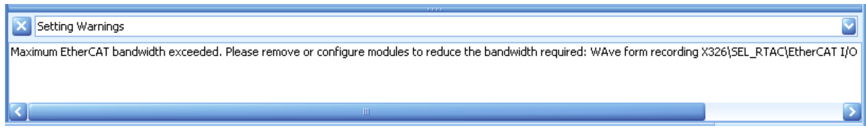


Figure 2.103 EtherCAT Bandwidth Exceeded

LEA Module Simple Tags

The LEA measurements are in a simplified format referred to as "Simple Tags." Complex measured values (CMV) display as the vector_t data subtype containing the angle and magnitude quantities. Measured values (MV) display as the REAL numeric type.

TimeStamp_t and quality_t tags are still available at the module level and automatically insert when mapping a Simple Tag to another tag type in the tag processor. Switching to the code view in the tag processor allows you to modify the default quality and time source if desired.

Waveform Recording

Waveforms are recorded in COMTRADE format with an accompanying configuration file. File names are stored in the following format:

StationName_ModuleID_Protocol_YYYYMMDD_HHMMSSMSC.yyy

where:

yyy is one of two file name extensions: CFG for configuration files or DAT for data files.

Protocol is the name of the protocol used (e.g., ECAT).

StationName is defined by the RTAC hostname, which is configurable through the web interface. When the RTAC hostname is changed, the EtherCAT network must be disabled and re-enabled for the new StationName to appear in the file name of waveform records.

A maximum of 1024 COMTRADE events can be stored in the RTAC, with the oldest record being deleted when a new event is triggered and 1024 events are exceeded. The SEL-2245-221 can record two back-to-back events. A new record can be triggered as soon as the present one has finished recording. A trigger condition that occurs before the current recording finishes is ignored. These records are first stored locally to prevent data loss and then transferred to the RTAC for downloading via the built-in web server.

Inputs

The following voltage and current inputs are recorded in the COMTRADE files generated by the SEL-2245-221 modules:

- IA, IB, IC, IN
- VA, VB, VC, VS (Wye)
- VAB, VBC, VCA (Delta)

NOTE
Waveform settings are configured in the individual LEA module settings tabs.

NOTE
Oscillography Pre-Trigger Length is relative to when the module receives the trigger through EtherCAT. Two or more processing intervals of the configured RTAC task cycle rate should be added to the desired pre-trigger length to ensure all desired data are captured.

Table 2.62 Waveform Settings

Setting	Description
VoltageInputConfiguration	Voltage input configuration
FrequencyNominal	Nominal system frequency
OscillographyCaptureRate	Sampling rate: 1, 2, 4, 8, 24 kHz; software-selectable
OscillographyRecordLength	Total record length: 6 seconds at 24 kHz 18 seconds at 8 kHz 36 seconds at 4 kHz 72 seconds at 2 kHz 144 seconds at 1 kHz
OscillographyPreTriggerLength	Pre-trigger record length: 0.05 seconds minimum to a maximum of (record length – 0.05) seconds
PTRatio	Phase A, B, C PT winding ratio
PTRatioS	Synch PT winding ratio
StationName	RTAC hostname
ModuleID	Module identifier (module name in ACSELERATOR RTAC)
module_name.Event_Trigger	Boolean input that triggers waveform recording in the module and only responds to rising edges

Downloading Events Via the Web Interface

NOTE
To reduce bandwidth, only individually enabled LEA module tags are included in the EtherCAT message frame. This allows the use of more LEA modules if fewer tags are needed. The EtherCAT connection tab shows the amount of bandwidth used by your current configuration.

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **HTTPS://** followed by the RTAC Ethernet address.

Step 4. Log in to the RTAC web interface.

Step 5. Click **Event Collection** under **Reports** in the navigation pane on the left.

Step 6. Sort events by selecting a time frame from **Timestamp Query** and the **Device Filter**.

Step 7. Select the desired events and click **Download**. You can also delete events by clicking **Delete**.

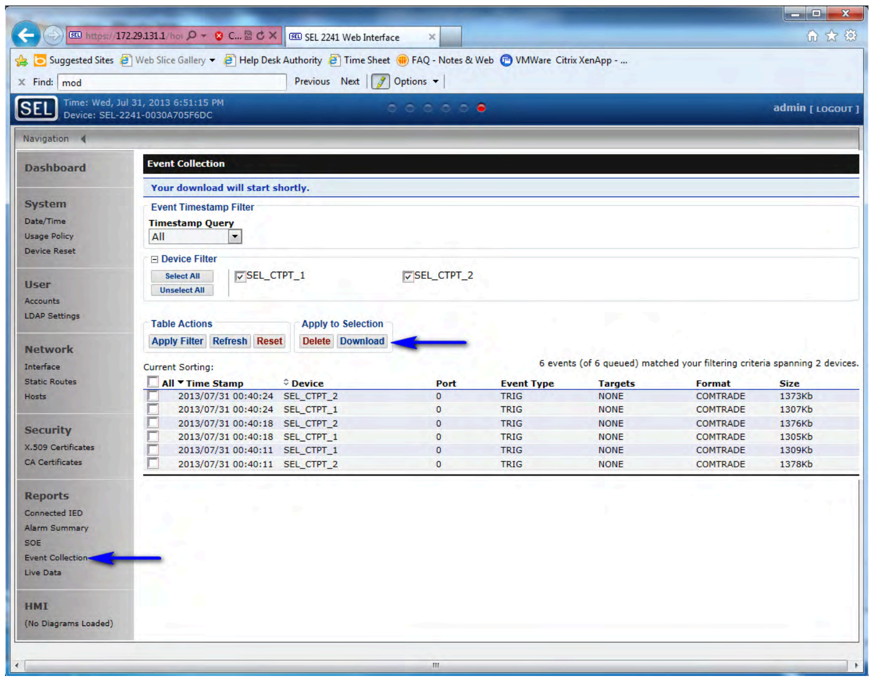


Figure 2.104 Waveform Record Retrieval

Viewing Waveforms and Event Files

Event files created by the SEL-2245-221 are in COMTRADE format. Event viewing software such as SYNCHROWAVE Event is required to open these files.

Figure 2.105 SYNCHROWAVE Event Display Waveform

SEL-2245-3 Eight DC Analog Output Module

The SEL-2245-3 DC Analog Output Module has eight sourcing dc outputs. Each output is user-configurable via a software switch between ±20 mA or ±10 Vdc. Use the **Insert** ribbon as shown in *Figure 2.110* to add an instance of this module to your project. The displayed pop-up will allow you to change the module name if necessary. When finished, click **Insert**.

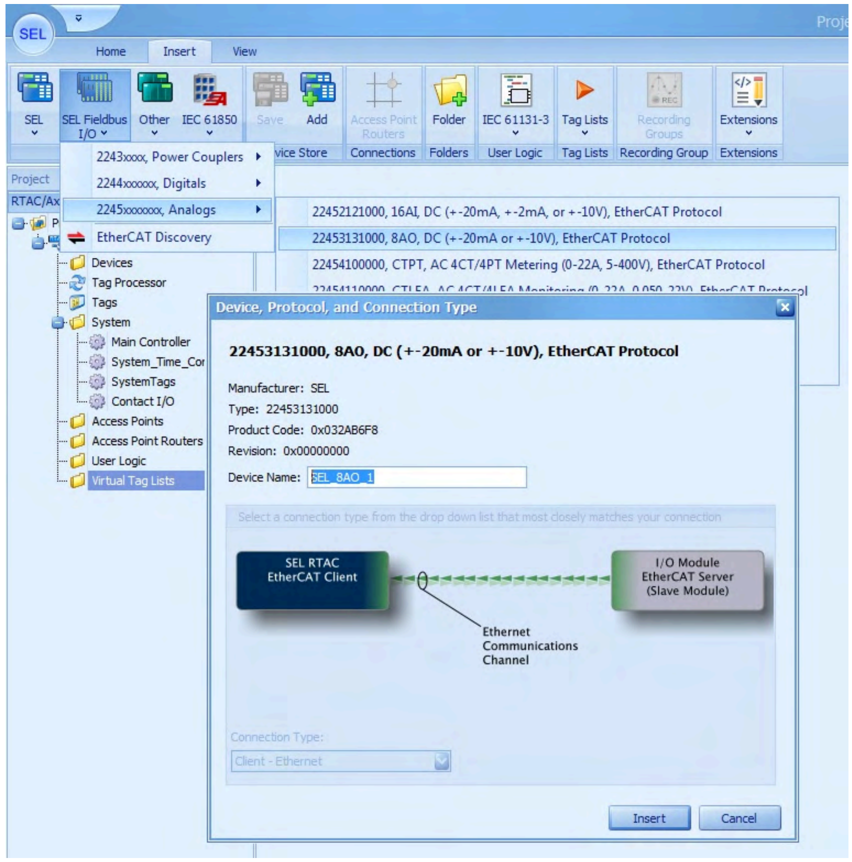


Figure 2.106 Insert an SEL-2245-3 DC Analog Output Module

Configuring the DC Analog Output Module

In the project tree, select the module to open the **Analog Outputs** dialog window. Go to the **Analog Outputs** tab to configure the analog output parameters. See *Specifications* in the *SEL-2240 Instruction Manual* for output specifications.

Enable
When TRUE, this tag will be included when the program is compiled. When FALSE, it will not be included. Disable all tags not in use to optimize processing resources.
Tag Name
Programming designation of a resource.
Tag Alias
Input an alternative name to use in place of the Tag Name while programming (for example, SEL_8AO_1_ECAT.AO_VALUE_001 could be replaced by "Conveyor_Speed").
Analog Type
Software switch for selecting voltage and current output modes.
Engineering Units High and Engineering Units Low
The real-world quantities that the analog signal represents: temperature, pressure, speed, etc.

Physical High and Physical Low

The high and low signal levels produced at the outputs in milliamperes (mA) or volts.

NOTE: The Analog Output is not restricted from driving above or below these values. To limit the output level, see Clamp Enable.

Example: Settings for a Variable Frequency Drive (VFD) connected to a conveyor with a maximum speed of 100 Feet Per Minute (FPM) based upon a 0 to +10 Vdc voltage reference would be as follows:
  Engineering Units High: 100
  Engineering Unit Low: 0
  Physical High: 10
  Physical Low: 0

The software will use the Engineering Units and Physical to create a slope equation to relate the real conveyor speed to a voltage for the VFD. In this example, setting AO_Value_*xxx* to 60 FPM will cause Output *xxx* to drive to +6 Vdc. However, if Clamp_ENABLE is FALSE and AO_VALUE_*xxx* is set to –90 FPM, the Analog Output will drive to –9 Vdc.

Safe Output

State applied to output when the module loses communication with the EtherCAT client for 100 ms or more. Options are **Last Value** or **User Defined**.

User Safe Output

Output is driven to this value when Safe Output is User Defined and the module loses EtherCAT communication for 100 ms or more. The output will be driven to this value even if it is outside the optional Clamp High or Clamp Low settings. This value is in engineering units.

Clamp Enable

Enables or disables software limits on output values.

Clamp Low

Lower limit on output when Clamp Enable is True. Value is in engineering units.

Clamp High

Upper limit on output when Clamp Enable is True. Value is in engineering units.

Comment

Editable description field.

Input TAGS	
AO_VALUE_*xxx*	Output value of the analog output in engineering units.
RAMP_ENABLE_*xxx*	Boolean input to enable ramping on an output channel.
RAMP_HOLD_*xxx*	Boolean input to hold the output at the current value during a ramp operation.
RAMP_TARGET_*xxx*	Value for the output to ramp to from AO_VALUE_*xxx* in engineering units.
RAMP_TIME_*xxx*	Number of milliseconds to ramp from VALUE_*xxx* to RAMP_TARGET_ *xxx*.
RAMP_TRIGGER_*xxx*	Boolean rising-edge trigger that latches AO_VALUE_*xxx*, RAMP_TARGET_*xxx*, and RAMP_TIME_*xxx*; begins or restarts ramp operation from VALUE_*xxx*.

Output TAGS	
AO_VALUE_ACTUAL_*xxx*	Actual value being output after ramping and clamping functions are applied. Value is in engineering units.
RAMPING_*xxx*	Boolean output indicating a ramp is in progress but the output has not reached RAMP_TARGET_*xxx*.
OUTPUT_CLAMPING_*xxx*	Boolean output indicating AO_VALUE_*xxx* is outside Clamp High and Clamp Low settings and is being clamped.

Field Calibration Procedure

Step 1. Set the Physical High and Physical Low values to the desired output levels (e.g., +10 V and –10 V).

Step 2. Set Engineering Units High equal to Physical High and Engineering Units Low equal to Physical Low.

Step 3. Force AO_VALUE_*xxx* to the High value (+10 V).

Step 4. Record the value displayed on your calibrated meter. This will be MEAS_high.

Step 5. Force AO_VALUE_*xxx* to the Low value (–10 V).

Step 6. Record the value displayed on your calibrated meter. This will be MEAS_low.

Step 7. Set Physical High value to (Physical High – MEAS_high + Physical High).

Step 8. Set Physical Low to (Physical Low – MEAS_low + Physical Low).

Step 9. Set the Engineering High and Low units to the desired range for your project.

Ramping Functions

If RAMP_ENABLE_*xxx* is TRUE and there is a rising edge on RAMP_TRIGGER_*xxx*, RAMP_TARGET_*xxx*, RAMP_TIME_*xxx*, andAO_VALUE_*xxx* are latched and the ramp process begins.

The output ramps from AO_VALUE_*xxx* to RAMP_TARGET_*xxx* over the time interval of RAMP_TIME_*xxx*.

If RAMP_ENABLE_*xxx* becomes FALSE, the output will return to the AO_VALUE_*xxx* that was latched.

If RAMP_HOLD_*xxx* becomes TRUE, the ramp timer is paused and the output will hold the current value until RAMP_HOLD_*xxx* or RAMP_ENABLE_*xxx* becomes FALSE.

When RAMP_TARGET_*xxx* is reached, the output will remain at that value until RAMP_ENABLE_*xxx* becomes FALSE.

If a rising edge is detected on RAMP_TRIGGER_*xxx* during the ramp process, RAMP_TARGET_*xxx*, RAMP_TIME_*xxx*, and AO_VALUE_*xxx* are latched and the ramp process restarts with the new values.

RAMP_TRIGGER_*xxx* is ignored if RAMP_HOLD_*xxx* is TRUE.

Figure 2.107 is an example of RAMP_HOLD_*xxx* asserting and deasserting during a ramp process.

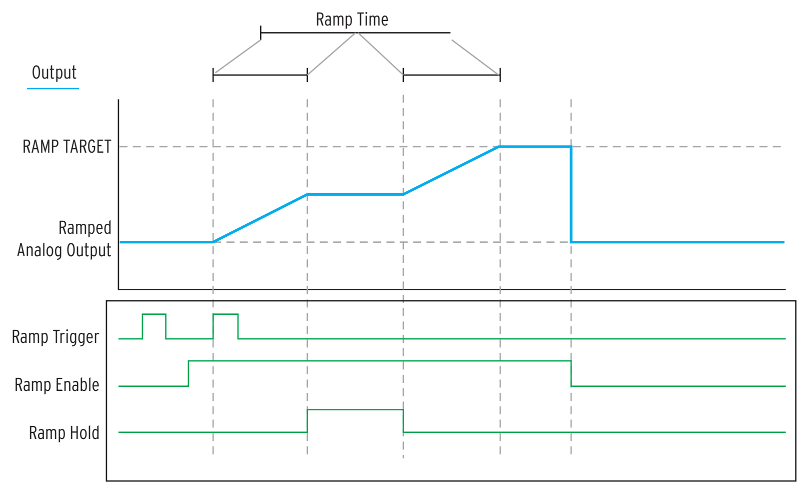


Figure 2.107 Hold During a Ramp Process

Figure 2.108 is an example of retriggering a ramp operation.

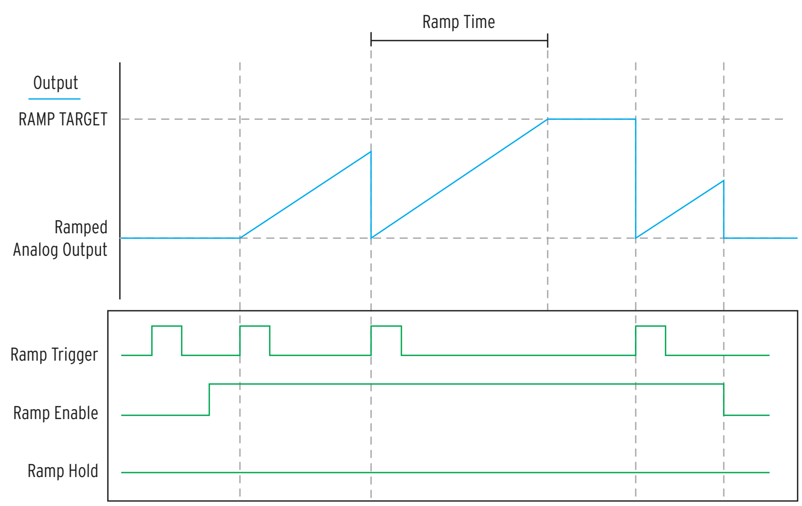


Figure 2.108 Retriggering a Ramp During a Ramp Process

SEL-2245-4 AC Metering Module

The SEL-2245-4 AC Metering Module (CTPT) includes four Potential Transformer inputs and four Current Transformer inputs with isolated returns for measuring ac signals. Refer to *Specifications in the SEL-2240 Axion Instruction Manual* for details on the input range. Use the **Insert** ribbon as shown in *Figure 2.113* to add an instance of this module to your project. The displayed pop-up will allow you to change the module name if necessary. When finished, click on **Insert**.

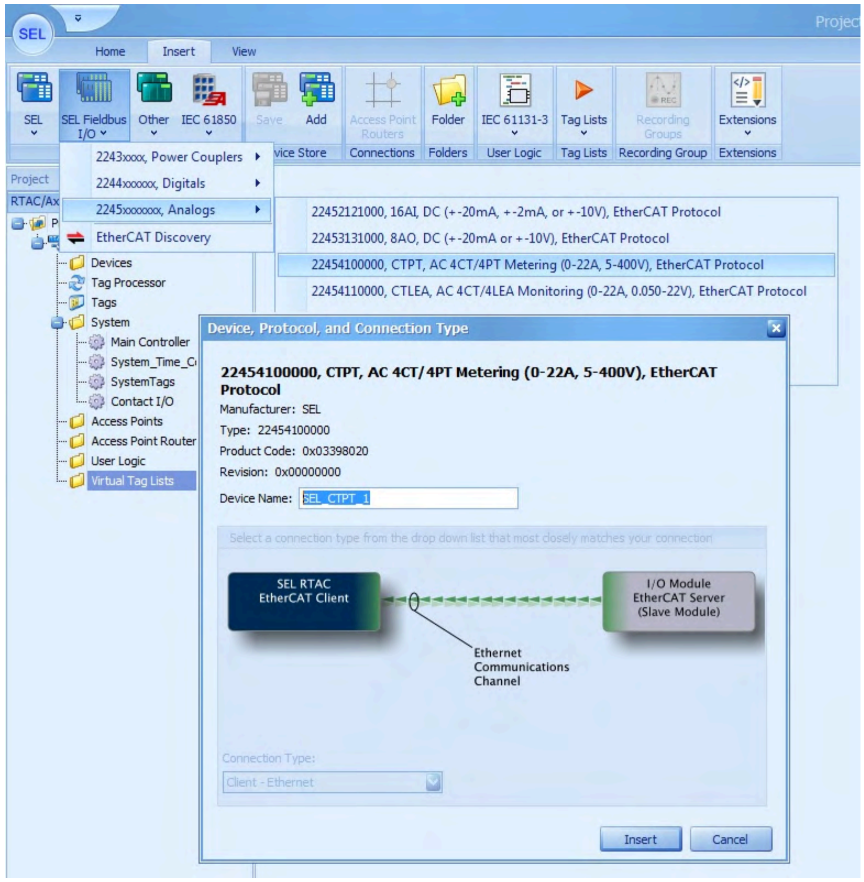


Figure 2.109 Insert an SEL-2245-4 CT/PT Analog Input Module

Configuring the AC Analog Input Module

In the project tree, select the module to open the module settings dialog window. Click on the **Settings** tab to configure the module.

SEL_CTPT_1_ECAT

22454100000, CTPT, AC 4CT/4PT Metering (0-22A, 5-400V), Client - Ethernet [EtherCAT Protocol]

Properties | Settings | Analog Inputs | Diagnostics | Tags

Setting	Value	Range	Description
General			
Phase ABC CT Ratio	250	1.00-50000.00	Phase A, B, and C CT winding ratio
Neutral CT Ratio	250	1.00-50000.00	Neutral CT winding ratio
Voltage Configuration	Wye	Delta,Wye	Voltage configuration
Phase ABC PT Ratio	35	1.00-10000.00	Phase A, B, and C PT winding ratio
Synch PT Ratio	35	1.00-10000.00	Synch PT winding ratio
Phase Rotation	ABC	ABC,ACB	Phase rotation
Enable RMS Tags	True	True,False	Enables the creation of the RMS tags
Enable Fundamental Tags	False	True,False	Enables the creation of the fundamental tags
Enable Synchrophasor Tags	False	True,False	Enables the creation of the synchrophasor tags
Reference Angle	Positive Sequence	Positive Sequen...	The reference angle for fundamental tags
Oscillography			
Oscillography Record Length	1	0.1 increments ...	Total length of an oscillography capture
Oscillography Pre-trigger L...	0.05	0.01 increments...	Pre-trigger length of an oscillography capture
Oscillography Capture Rate	8	1,2,4,8,24 (kilo...	The capture rate of oscillography data
PMU			
Station Name	SEL_CTPT_1_ECAT	1-255 (characte...	The PMU station name
PMU Id	1	1-65534	Identifier for this PMU
Voltage Phasor Data Set	Phase	All,Phase,Positi...	Voltage Phasor channels for PMU data set
Current Phasor Data Set	Phase	All,Phase,Positi...	Current Phasor channels for PMU data set
Synchrophasors			
Phase Voltage Angle Comp...	0	-179.99-180.00...	Angle compensation for the A, B, and C phase voltage synchrophasor tags
Synch Voltage Angle Comp...	0	-179.99-180.00...	Angle compensation for the Synch voltage synchrophasor tags
Phase Current Angle Comp...	0	-179.99-180.00...	Angle compensation for the A, B, and C phase current synchrophasor tags
Neutral Current Angle Com...	0	-179.99-180.00...	Angle compensation for the neutral current synchrophasor tags

Figure 2.110 CT/PT Module Settings

Setting	Configuration Options
Phase Rotation	ABC or ACB
Voltage Configuration	Configure inputs for three-wire Delta or four-wire Wye
Phase ABC CT Ratio	1.00–50000.00
Neutral CT Ratio	1.00–50000.00
Phase ABC PT Ratio	1.00–10000.00
Synch PT Ratio	1.00–10000.00
Enable RMS Tags	Shows rms quantities in the **Analog Inputs** tab
Enable Fundamental Tags	Shows Fundamental quantities in the **Analog Inputs** tab
Enable Synchrophasor Tags	Shows Synchrophasor quantities in the **Analog Inputs** tab
Reference Angle	Select reference for Fundamental Vector quantities
Oscillography Capture Rate	Capture rate of oscillography data
Oscillography Record Length	Length of oscillography capture. Capture rate affects maximum record length (see Specifications).
Oscillography Pre-Trigger Length	Length of time before trigger to start capture

NOTE

In firmware R141 or later, only individual tags enabled per AC Metering Module are included in the EtherCAT message frame to reduce bandwidth. This allows the use of more AC Metering Modules if fewer tags are needed. The EtherCAT connection tab shows the amount of bandwidth used by your current configuration. Prior to R141, enabling the RMS, Fundamental, or Synchrophasor tag streams added all tags of each stream to the EtherCAT bandwidth.

NOTE

RMS voltage quantities have units of kilovolts and Fundamental voltage quantities are in units of volts.

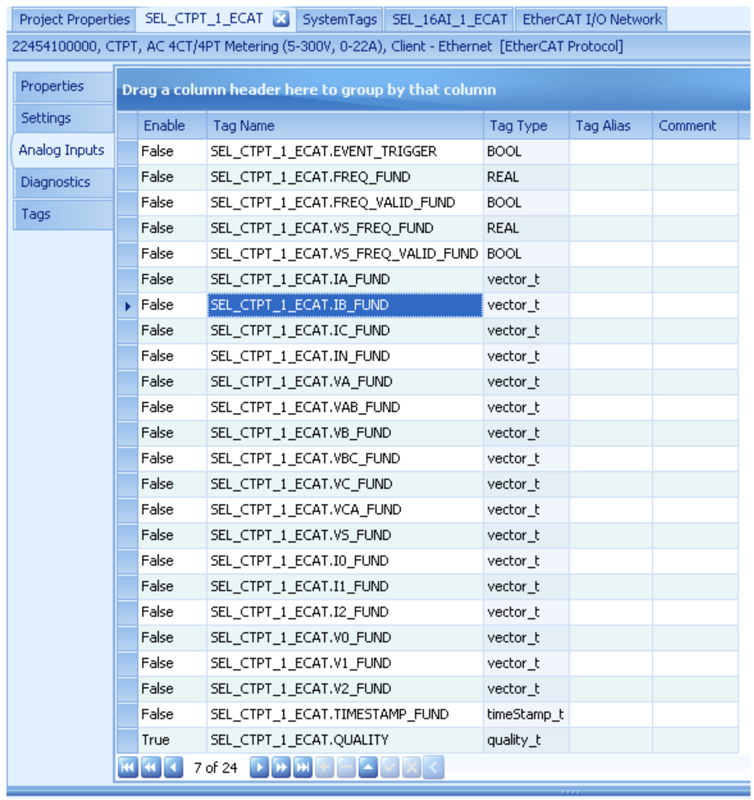


Figure 2.111 CT/PT Tag Enable Settings

Analog Input	Description
Enable	When True, this tag will be included when the program is compiled. When False it will not be included. Disable all tags not in use to optimize processing resources. If there are too many analog data on the EtherCAT backplane, you will receive a message that the bandwidth has been exceeded (shown in *Figure 2.112*).
Tag Name	Programming designation of a resource.
Tag Alias	Input an alternative name to use in place of the Tag Name while programming (for example, SEL_CTPT_1_ECAT.IA_FUND could be replaced by IA_Bus1).

NOTE

If you receive the message "EtherCAT Bandwidth Exceeded," you will need to disable a number of RMS, Fundamental, or Synchrophasor tags on one or more CT/PT modules to reduce the bandwidth in use.

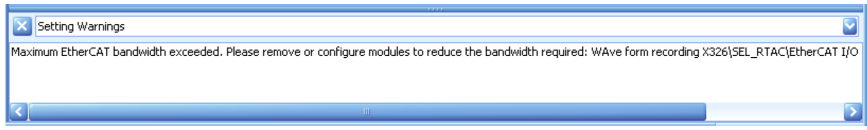


Figure 2.112 EtherCAT Bandwidth Exceeded

AC Module Simple Tags

The CT/PT measurements are in a simplified format referred to as "Simple Tags." Complex measured values (CMV) display as the vector_t data subtype containing the angle and magnitude quantities. Measured values (MV) display as the REAL numeric type.

TimeStamp_t and quality_t tags are still available at the module level and automatically insert when mapping a Simple Tag to another tag type in the tag processor. Switching to the code view in the tag processor will allow you to modify the default quality and time source if desired.

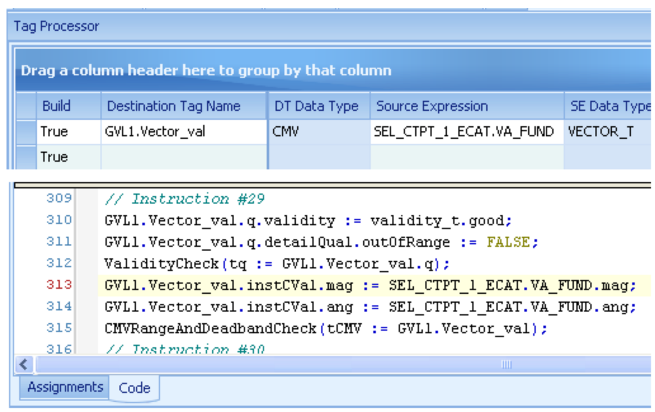


Figure 2.113 CT/PT Simple Tag to Complex Tag Mapping

Waveform Recording

Waveforms will be recording in COMTRADE format with an accompanying configuration file. File names are stored as follows:

StationName_ModuleID_Protocol_YYYYMMDD_HHMMSSMSC.yyy

where:

yyy	is one of two file name extensions: CFG for configuration files or DAT for data files.
Protocol	is the name of the protocol used (e.g., ECAT).
StationName	is defined by the RTAC hostname, which is configurable through the web interface. When the RTAC hostname is changed, the EtherCAT network must be disabled and re-enabled for the new StationName to appear in the file name of waveform records.

A maximum of 1024 COMTRADE events can be stored in the RTAC with the oldest record being deleted when a new event is triggered and 1024 is exceeded. The SEL-2245-4 can record two back-to-back events. A new record can be triggered as soon as the current one has finished recording. A trigger condition that occurs before the current recording finishes is ignored. These records are first stored locally to prevent data loss and then transferred to the RTAC for downloading via the built-in web server.

Inputs

The following voltage and current inputs are recorded in the COMTRADE files generated by the SEL-2245-4 modules:

- IA, IB, IC, IN
- VA, VB, VC, VS (Wye)
- VAB, VBC, VCA (Delta)

NOTE
Waveform settings are configured in the individual CT/PT module settings tabs.

NOTE
Oscillography Pre-Trigger Length is relative to when the module receives the trigger through EtherCAT. Two or more processing intervals of the configured RTAC task cycle rate should be added to the desired pre-trigger length to ensure all desired data are captured.

Table 2.63 Waveform Settings

Setting	Description
VoltageInputConfiguration	Voltage input configuration
FrequencyNominal	Nominal system frequency
OscillographyCaptureRate	Sampling rate: 1, 2, 4, 8, 24 kHz; software-selectable
OscillographyRecordLength	Total record length: 6 seconds at 24 kHz 18 seconds at 8 kHz 36 seconds at 4 kHz 72 seconds at 2 kHz 144 seconds at 1 kHz
OscillographyPreTriggerLength	Pre-trigger record length: 0.05 seconds minimum to a maximum of (record length – 0.05) seconds
CTRatio	A-, B-, and C-phase CT winding ratio
CTRatioN	Neutral CT winding ratio
PTRatio	Phase A, B, C PT winding ratio
PTRatioS	Synch PT winding ratio
StationName	RTAC hostname
ModuleID	Module identifier (module name in ACSELERATOR RTAC)
module_name.Event_Trigger	This is a Boolean input that triggers waveform recording in the module and only responds to rising edges.

Downloading Events Via the Web Interface

NOTE

In firmware R140 or later, only individual tags enabled per AC Protection Module are included in the EtherCAT message frame to reduce bandwidth. This allows the use of more AC Protection Modules if fewer tags are needed. The EtherCAT connection tab shows the amount of bandwidth used by your current configuration. Prior to R140, enabling the RMS, Fundamental, or Synchrophasor tag streams added all tags of each stream to the EtherCAT bandwidth.

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **https://** followed by the RTAC Ethernet address.

Step 4. Log in to the RTAC web interface.

Step 5. Click on **Event Collection** under **Reports** in the navigation pane on the left.

Step 6. Sort events by selecting a time frame from **Timestamp Query** and the **Device Filter**.

Step 7. Select the desired events and click **Download**.

Step 8. You can also delete events by clicking **Delete**.

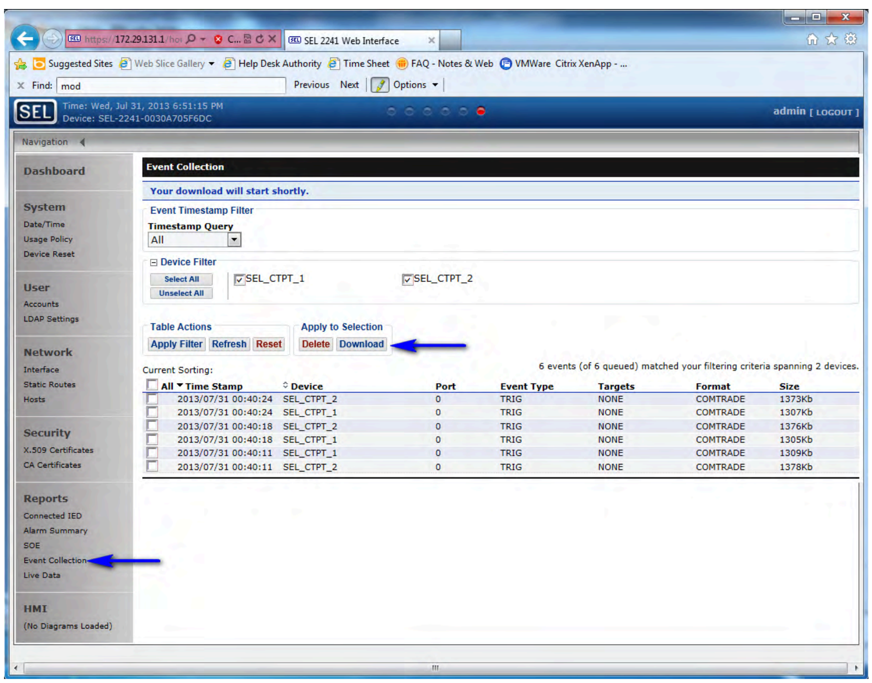


Figure 2.114 Waveform Record Retrieval

Viewing Waveforms and Event Files

Events files created by the SEL-2245-4 are in COMTRADE format. Event viewing software such as SYNCHROWAVE Event is required to open these files.

Figure 2.115 SYNCHROWAVE Event Display Waveform

SEL-2245-411 Standard Current and Low-Voltage (LEA) Monitoring Module

The SEL-2245-411 Standard Current and Low-Voltage (LEA) Monitoring Module (4 CT/4 LEA) includes four low-voltage inputs and four current transformer inputs with isolated returns for measuring ac signals. Refer to *Specifications in the SEL-2240 Axion Instruction Manual* for details on the input range. Use the **Insert** ribbon as shown in *Figure 2.116* to add an instance of this module to your project. The displayed pop-up will allow you to change the module name if necessary. When finished, click **Insert**.

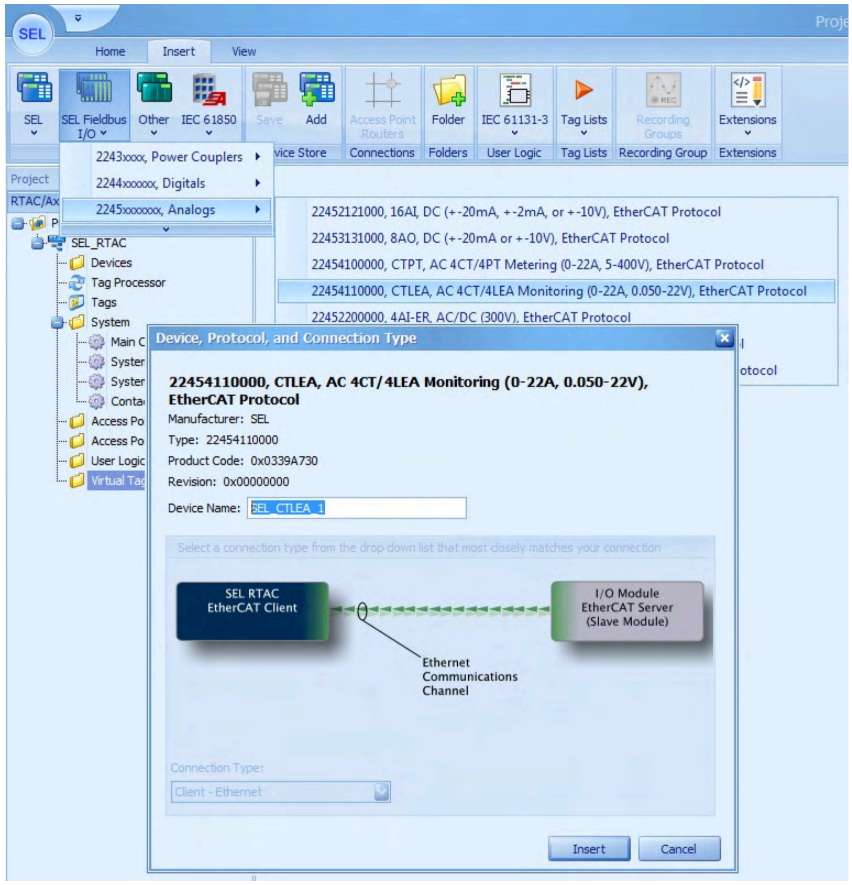


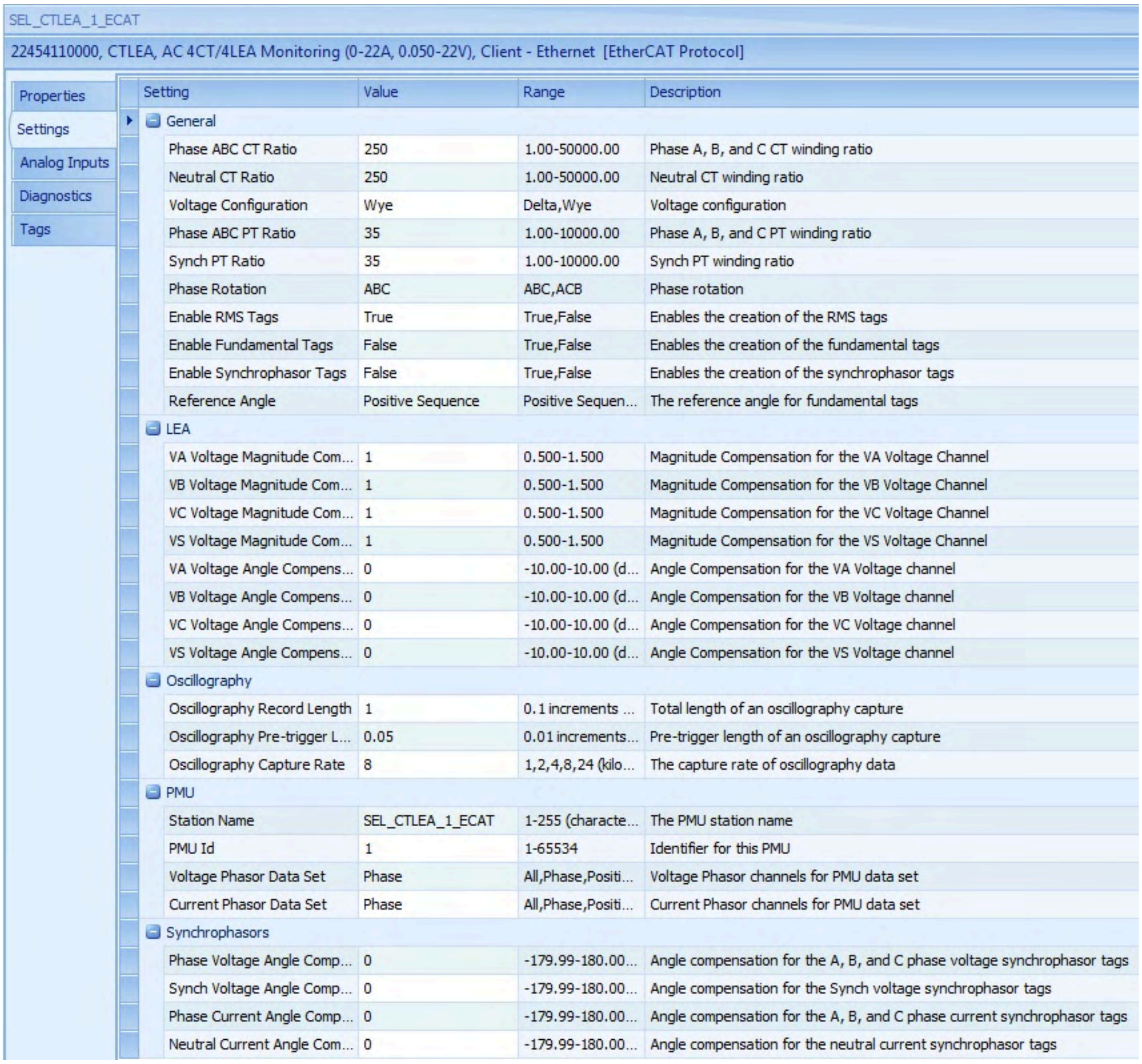
Figure 2.116 Insert an SEL-2245-411 4 CT/4 LEA Module

Configuring the CT/LEA Monitoring Module

In the project tree, select the module to open the module settings window. Click the **Settings** tab to configure the module.

NOTE

Only individual tags enabled per the CT/LEA Monitoring Module are included in the EtherCAT message frame to reduce bandwidth. This allows for the use of more modules if fewer tags are needed. The EtherCAT connection tab shows the amount of bandwidth used by your current configuration.



SEL_CTLEA_1_ECAT

22454110000, CTLEA, AC 4CT/4LEA Monitoring (0-22A, 0.050-22V), Client - Ethernet [EtherCAT Protocol]

Properties
Settings
Analog Inputs
Diagnostics
Tags

Setting	Value	Range	Description
General			
Phase ABC CT Ratio	250	1.00-50000.00	Phase A, B, and C CT winding ratio
Neutral CT Ratio	250	1.00-50000.00	Neutral CT winding ratio
Voltage Configuration	Wye	Delta,Wye	Voltage configuration
Phase ABC PT Ratio	35	1.00-10000.00	Phase A, B, and C PT winding ratio
Synch PT Ratio	35	1.00-10000.00	Synch PT winding ratio
Phase Rotation	ABC	ABC,ACB	Phase rotation
Enable RMS Tags	True	True,False	Enables the creation of the RMS tags
Enable Fundamental Tags	False	True,False	Enables the creation of the fundamental tags
Enable Synchrophasor Tags	False	True,False	Enables the creation of the synchrophasor tags
Reference Angle	Positive Sequence	Positive Sequen...	The reference angle for fundamental tags
LEA			
VA Voltage Magnitude Com...	1	0.500-1.500	Magnitude Compensation for the VA Voltage Channel
VB Voltage Magnitude Com...	1	0.500-1.500	Magnitude Compensation for the VB Voltage Channel
VC Voltage Magnitude Com...	1	0.500-1.500	Magnitude Compensation for the VC Voltage Channel
VS Voltage Magnitude Com...	1	0.500-1.500	Magnitude Compensation for the VS Voltage Channel
VA Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VA Voltage channel
VB Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VB Voltage channel
VC Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VC Voltage channel
VS Voltage Angle Compens...	0	-10.00-10.00 (d...	Angle Compensation for the VS Voltage channel
Oscillography			
Oscillography Record Length	1	0.1 increments ...	Total length of an oscillography capture
Oscillography Pre-trigger L...	0.05	0.01 increments...	Pre-trigger length of an oscillography capture
Oscillography Capture Rate	8	1,2,4,8,24 (kilo...	The capture rate of oscillography data
PMU			
Station Name	SEL_CTLEA_1_ECAT	1-255 (characte...	The PMU station name
PMU Id	1	1-65534	Identifier for this PMU
Voltage Phasor Data Set	Phase	All,Phase,Positi...	Voltage Phasor channels for PMU data set
Current Phasor Data Set	Phase	All,Phase,Positi...	Current Phasor channels for PMU data set
Synchrophasors			
Phase Voltage Angle Comp...	0	-179.99-180.00...	Angle compensation for the A, B, and C phase voltage synchrophasor tags
Synch Voltage Angle Comp...	0	-179.99-180.00...	Angle compensation for the Synch voltage synchrophasor tags
Phase Current Angle Comp...	0	-179.99-180.00...	Angle compensation for the A, B, and C phase current synchrophasor tags
Neutral Current Angle Com...	0	-179.99-180.00...	Angle compensation for the neutral current synchrophasor tags

Figure 2.117 CT/LEA Module Settings

Setting	Configuration Options
Phase Rotation	ABC or ACB
Voltage Configuration	Configure inputs for three-wire Delta or four-wireWye
Phase ABC CT Ratio	1.00–50000.00
Neutral CT Ratio	1.00–50000.00
Phase ABC PT Ratio	1.00–10000.00
Synch PT Ratio	1.00–10000.00
Enable RMS Tags	Shows rms quantities in the **Analog Inputs** tab
Enable Fundamental Tags	Shows Fundamental quantities in the **Analog Inputs** tab
Reference Angle	Select reference for Fundamental Vector quantities
Oscillography Capture Rate	Capture rate of oscillography data

Setting	Configuration Options
Oscillography Record Length	Length of oscillography capture. Capture rate affects maximum record length (see *Specifications* in the *SEL-2240 Instruction Manual*).
Oscillography Pre-Trigger Length	Length of time before trigger to start capture
VA Voltage Magnitude Compensation	0.500 to 1.500
VB Voltage Magnitude Compensation	0.500 to 1.500
VC Voltage Magnitude Compensation	0.500 to 1.500
VS Voltage Magnitude Compensation	0.500 to 1.500
VA Voltage Angle Compensation	–10.00 to 10.00
VB Voltage Angle Compensation	–10.00 to 10.00
VC Voltage Angle Compensation	–10.00 to 10.00
VS Voltage Angle Compensation	–10.00 to 10.00
Phase Voltage Angle Compensation	–179.99 to 180.00
Synch Voltage Angle Compensation	–179.99 to 180.00
Phase Current Angle Compensation	–179.99 to 180.00
Neutral Current Angle Compensation	–179.99 to 180.00

NOTE

RMS voltage quantities have units of kilovolts and fundamental voltage quantities are in units of volts.

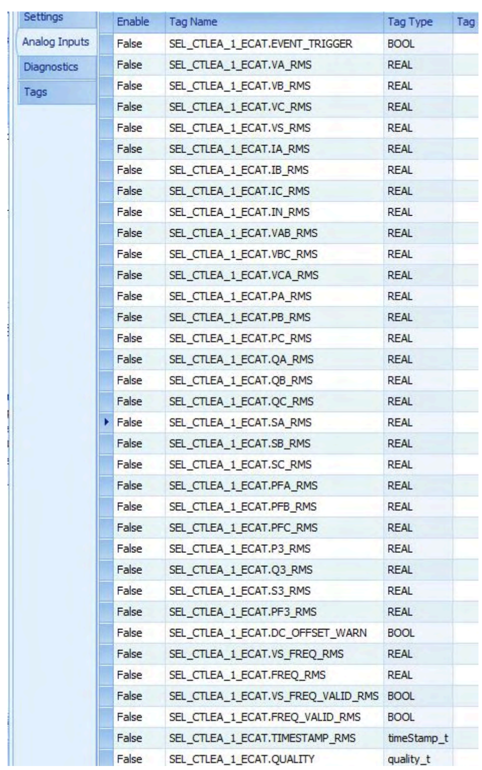


Figure 2.118 CT/LEA Tag Enable Settings

Analog Input	Description
Enable	When True, this tag will be included when the program is compiled. When False, it will not be included. Disable all tags not in use to optimize processing resources. If there are many analog data on the EtherCAT backplane, you will receive a message that the bandwidth has been exceeded (shown in *Figure 2.119*).
Tag Name	Programming designation of a resource.
Tag Alias	Input an alternative name to use in place of the Tag Name while programming (for example, SEL_CTLEA_1_ECAT.IA_FUND could be replaced by IA_Bus1).

NOTE
If you receive the message "EtherCAT Bandwidth Exceeded," you will need to disable a number of RMS or Fundamental tags on one or more CT/LEA modules to reduce the bandwidth in use.

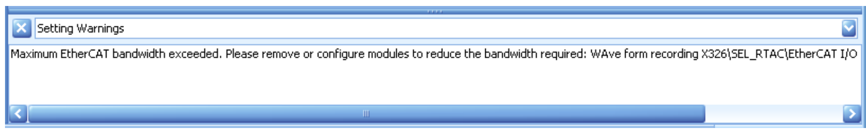


Figure 2.119 EtherCAT Bandwidth Exceeded

CT/LEA Module Simple Tags

The CT/LEA measurements are in a simplified format referred to as "Simple Tags." Complex measured values (CMV) display as the vector_t data subtype containing the angle and magnitude quantities. Measured values (MV) display as the REAL numeric type.

TimeStamp_t and quality_t tags are still available at the module level and automatically insert when mapping a Simple Tag to another tag type in the tag processor. Switching to the code view in the tag processor will allow you to modify the default quality and time source if desired.

Waveform Recording

Waveforms will be recorded in COMTRADE format with an accompanying configuration file. File names are stored in the following format:

StationName_ModuleID_Protocol_YYYYMMDD_HHMMSSMSC.yyy

where:

yyy is one of two file name extensions: CFG for configuration files or DAT for data files.

Protocol is the name of the protocol used (e.g., ECAT).

StationName is defined by the RTAC hostname, which is configurable through the web interface. When the RTAC hostname is changed, the EtherCAT network must be disabled and re-enabled for the new StationName to appear in the file name of waveform records.

A maximum of 1024 COMTRADE events can be stored in the RTAC with the oldest record being deleted when a new event is triggered and 1024 events are exceeded. The SEL-2245-411 can record two back-to-back events. A new record can be triggered as soon as the present one has finished recording. A trigger condition that occurs before the current recording finishes is ignored. These records are first stored locally to prevent data loss and then transferred to the RTAC for downloading via the built-in web server.

Inputs

The following voltage and current inputs are recorded in the COMTRADE files generated by the SEL-2245-411 modules:

- IA, IB, IC, IN
- VA, VB, VC, VS (Wye)
- VAB, VBC, VCA (Delta)

NOTE
Waveform settings are configured in the individual CT/LEA module settings tabs.

NOTE
Oscillography Pre-Trigger Length is relative to when the module receives the trigger through EtherCAT. Two or more processing intervals of the configured RTAC task cycle rate should be added to the desired pre-trigger length to ensure all desired data are captured.

Table 2.64 Waveform Settings

Setting	Description
VoltageInputConfiguration	Voltage input configuration
FrequencyNominal	Nominal system frequency
OscillographyCaptureRate	Sampling rate: 1, 2, 4, 8, 24 kHz; software-selectable
OscillographyRecordLength	Total record length: 6 seconds at 24 kHz 18 seconds at 8 kHz 36 seconds at 4 kHz 72 seconds at 2 kHz 144 seconds at 1 kHz
OscillographyPreTriggerLength	Pre-trigger record length: 0.05 seconds minimum to a maximum of (record length – 0.05) seconds
CTRatio	A-, B-, and C-phase CT winding ratio
CTRatioN	Neutral CT winding ratio
PTRatio	Phase A, B, C PT winding ratio
PTRatioS	Synch PT winding ratio
StationName	RTAC hostname
ModuleID	Module identifier (module name in ACSELERATOR RTAC)
module_name.Event_Trigger	This is a Boolean input that triggers waveform recording in the module and only responds to rising edges

Downloading Events Via the Web Interface

NOTE
Only individually enabled CT/LEA module tags are included in the EtherCAT message frame to reduce bandwidth. This allows the use of more CT/LEA modules if fewer tags are needed. The EtherCAT connection tab shows the amount of bandwidth used by your current configuration.

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **https://** followed by the RTAC Ethernet address.

Step 4. Log in to the RTAC web interface.

Step 5. Click on **Event Collection** under **Reports** in the navigation pane on the left.

Step 6. Sort events by selecting a time frame from **Timestamp Query** and the **Device Filter**.

Step 7. Select the desired events and click **Download**.

Step 8. You can also delete events by clicking **Delete**.

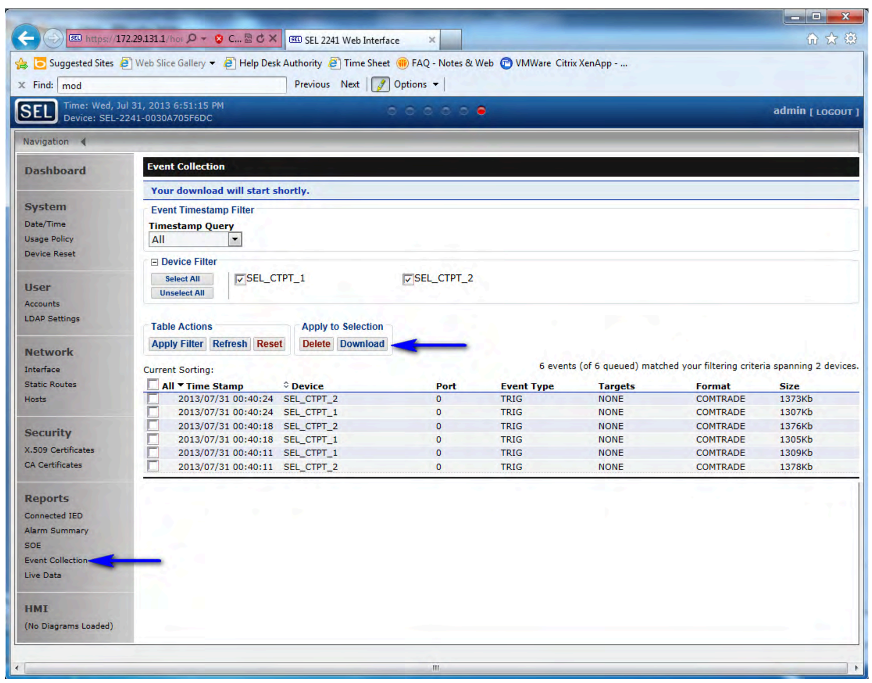


Figure 2.120 Waveform Record Retrieval

Viewing Waveforms and Event Files

Event files created by the SEL-2245-411 are in COMTRADE format. Event viewing software such as SYNCHROWAVE Event is required to open these files.

Figure 2.121 SYNCHROWAVE Event Display Waveform

SEL-2245-42 AC Protection Module

The SEL-2245-42 AC Protection Module (PRCTPT) includes three Potential Transformer inputs and three Current Transformer inputs with isolated returns for measuring ac signals. See *Specifications in the SEL-2240 Axion Instruction Manual* for details on the input range. Use the **Insert** ribbon, as shown in *Figure 2.122*, to add an instance of this module to your project. The displayed pop-up will allow you to change the module name if necessary. When finished, click **Insert**.

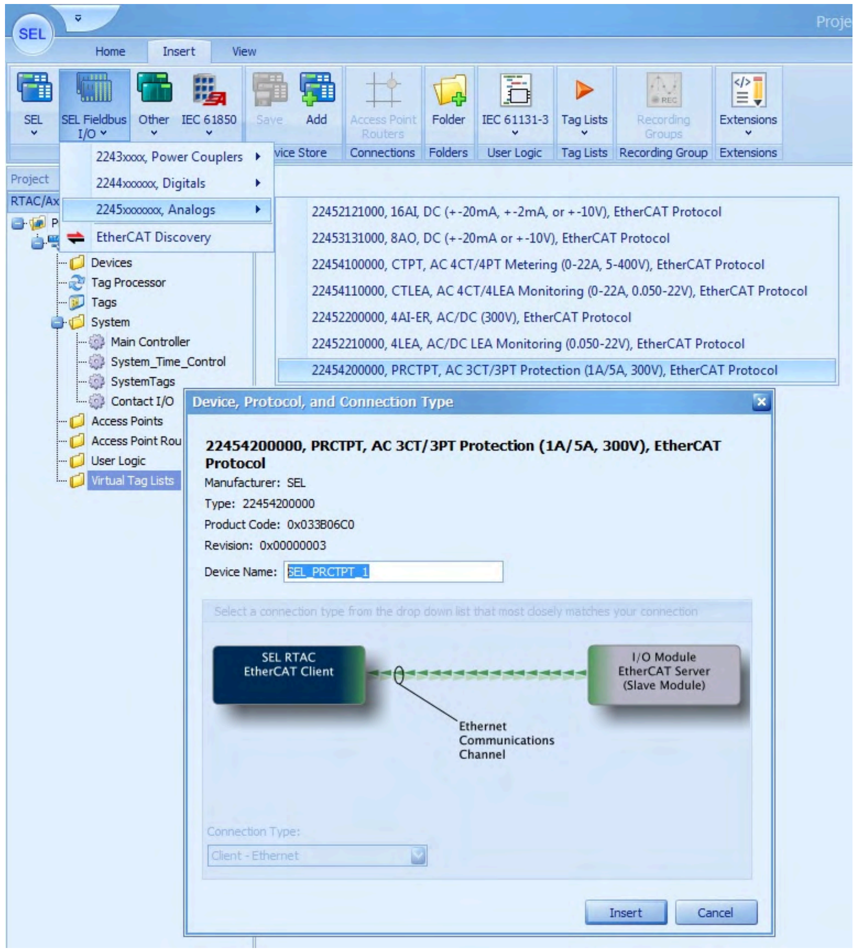
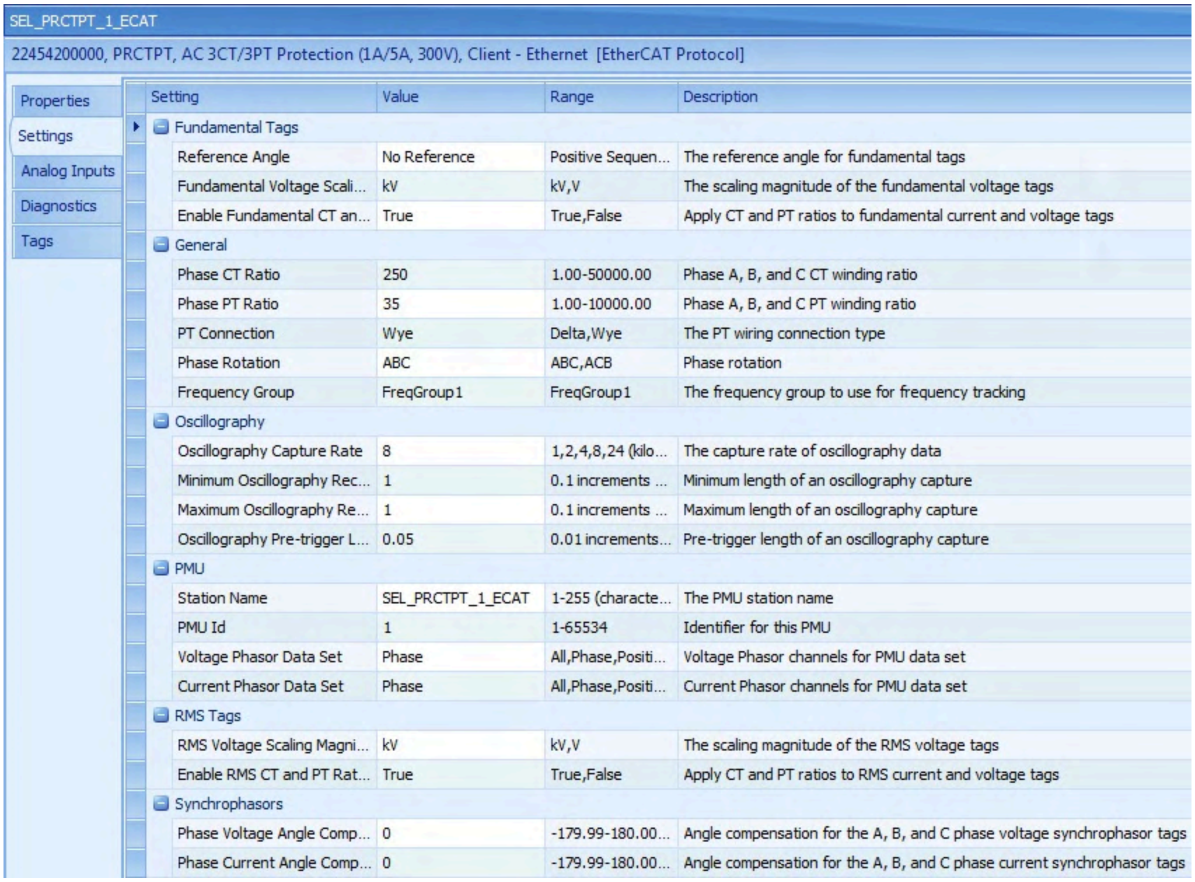


Figure 2.122 Insert an SEL-2245-42 AC Protection Module

Configuring the AC Protection Module

In the project tree, select the module to open the module settings dialog window. Click the **Settings** tab to configure the module.


SEL_PRCTPT_1_ECAT

22454200000, PRCTPT, AC 3CT/3PT Protection (1A/5A, 300V), Client - Ethernet [EtherCAT Protocol]

Properties
Settings
Analog Inputs
Diagnostics
Tags

Setting	Value	Range	Description
Fundamental Tags			
Reference Angle	No Reference	Positive Sequen...	The reference angle for fundamental tags
Fundamental Voltage Scali...	kV	kV,V	The scaling magnitude of the fundamental voltage tags
Enable Fundamental CT an...	True	True,False	Apply CT and PT ratios to fundamental current and voltage tags
General			
Phase CT Ratio	250	1.00-50000.00	Phase A, B, and C CT winding ratio
Phase PT Ratio	35	1.00-10000.00	Phase A, B, and C PT winding ratio
PT Connection	Wye	Delta,Wye	The PT wiring connection type
Phase Rotation	ABC	ABC,ACB	Phase rotation
Frequency Group	FreqGroup1	FreqGroup1	The frequency group to use for frequency tracking
Oscillography			
Oscillography Capture Rate	8	1,2,4,8,24 (kilo...	The capture rate of oscillography data
Minimum Oscillography Rec...	1	0.1 increments ...	Minimum length of an oscillography capture
Maximum Oscillography Re...	1	0.1 increments ...	Maximum length of an oscillography capture
Oscillography Pre-trigger L...	0.05	0.01 increments...	Pre-trigger length of an oscillography capture
PMU			
Station Name	SEL_PRCTPT_1_ECAT	1-255 (characte...	The PMU station name
PMU Id	1	1-65534	Identifier for this PMU
Voltage Phasor Data Set	Phase	All,Phase,Positi...	Voltage Phasor channels for PMU data set
Current Phasor Data Set	Phase	All,Phase,Positi...	Current Phasor channels for PMU data set
RMS Tags			
RMS Voltage Scaling Magni...	kV	kV,V	The scaling magnitude of the RMS voltage tags
Enable RMS CT and PT Rat...	True	True,False	Apply CT and PT ratios to RMS current and voltage tags
Synchrophasors			
Phase Voltage Angle Comp...	0	-179.99-180.00...	Angle compensation for the A, B, and C phase voltage synchrophasor tags
Phase Current Angle Comp...	0	-179.99-180.00...	Angle compensation for the A, B, and C phase current synchrophasor tags

Figure 2.123 PRCTPT Module Settings

Table 2.65 PRCTPT Module Settings Options

Setting	Configuration Options
Fundamental Tags	
Reference Angle	Positive Sequence, Phase A, No Reference
Fundamental Voltage Scaling Magnitude	kV or V
Enable Fundamental CT and PT Ratios	True, False
General	
Phase CT Ratio	1.00–50000.00
Phase PT Ratio	1.00–6000.00
PT Connection	Delta, Wye
Phase Rotation	ABC or ACB
Frequency Group	Assign the frequency group to use for frequency tracking
Oscillography	
Oscillography Capture Rate	1, 2, 4, 8, 24 kHz
Minimum Oscillography Record Length	0.5–72 seconds in 0.1 increments

Setting	Configuration Options
Maximum Oscillography Record Length	1–72 seconds in 0.1 increments
Oscillography Pre-trigger Length	0.05–0.95 seconds in 0.01 increments
RMS Tags	
RMS Voltage Scaling Magnitude	kV, V
Enable RMS CT and PT Ratios	True, False
Synchrophasors	
Phase Voltage Angle Compensation	–179.99–180.00
Phase Current Angle Compensation	–179.99–180.00
PMU	
Station Name	1–255 characters
Current Phasor Data Set	All, Phase, Positive Sequence, None
Voltage Phasor Data Set	All, Phase, Positive Sequence, None
PMU Id	1–65534

NOTE
To conserve EtherCAT bandwidth, enable only the necessary tags. An error message will appear in the diagnostic window if the EtherCAT bandwidth is exceeded.

Figure 2.124 PRCTPT Tag Enable Settings

NOTE

If you receive the message "EtherCAT Bandwidth Exceeded," you will need to disable a number of RMS, Fundamental, or Synchrophasor tags on one or more CT/PT modules to reduce the bandwidth in use.

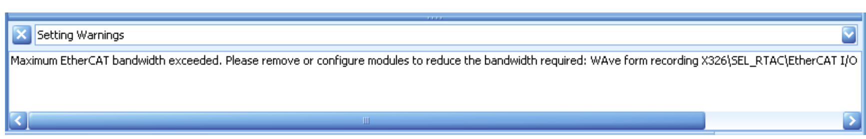


Figure 2.125 EtherCAT Bandwidth Exceeded

Table 2.66 Tag Settings

Analog Input	Description
Enable	When True, this tag will be included when the program is compiled. When False it will not be included. Disable all tags not in use to optimize processing resources. If there are too many analog data on the EtherCAT backplane, you will receive a message that the bandwidth has been exceeded (shown in *Figure 2.125*).
Tag Name	Programming designation of a resource.
Tag Alias	Input an alternative name to use in place of the Tag Name while programming (for example, SEL_PRCTPT_1_ECAT.IA_FUND could be replaced by IA_Bus1).

Configure Frequency Groups for the AC Protection Modules in the EtherCAT I/O network to define which module will be the reference for frequency tracking.

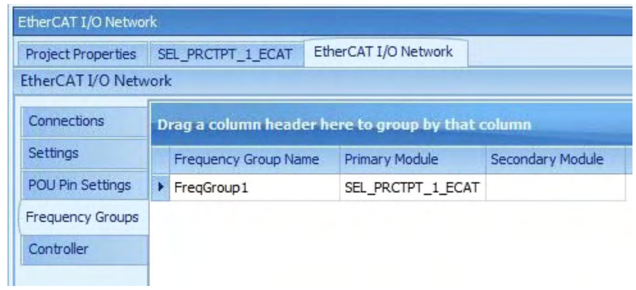


Figure 2.126 Configure PRCTPT Frequency Groups

AC Protection Module Simple Tags

CT/PT measurements are provided in a simplified format referred to as Simple Tags. Complex measured values (CMV) display as the vector_t data subtype containing the angle and magnitude quantities. Measured values (MV) display as the REAL numeric type.

TimeStamp_t and quality_t tags are still available at the module level and automatically insert when you map a Simple Tag to another tag type in the tag processor. Switching to the code view in the tag processor will allow you to modify the default quality and time source if needed.

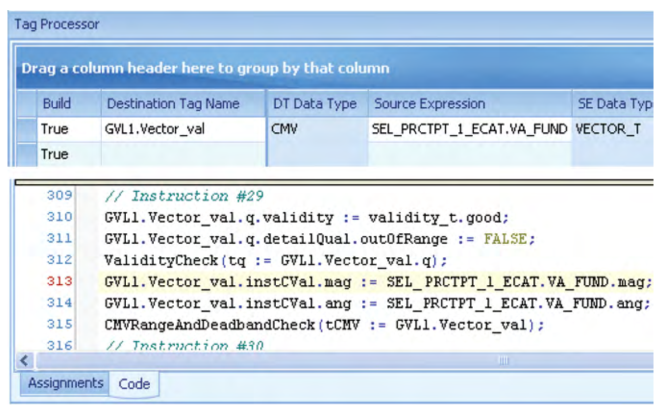


Figure 2.127 AC Protection Module Simple Tag to Complex Tag Mapping

Waveform Recording

Waveforms are recording in COMTRADE format with an accompanying configuration file. File names are stored as follows:

StationName_ModuleID_Protocol_YYYYMMDD_HHMMSSMSC.yyy

where:

yyy is one of two file name extensions: CFG for configuration files or DAT for data files.

Protocol	is the name of the protocol used (e.g., ECAT).
StationName	is defined by the RTAC hostname, which is configurable through the web interface. When the RTAC hostname is changed, the EtherCAT network must be disabled and re-enabled for the new StationName to appear in the file name of waveform records.

NOTE
Waveform settings are configured in the individual AC Protection Module setting tabs or in the assigned Recording Group Settings.

A maximum of 1024 COMTRADE events can be stored in the RTAC, with the oldest record being deleted when a new event is triggered and 1024 is exceeded. The SEL-2245-42 can record two back-to-back events. A new record can be triggered as soon as the current one has finished recording. These records are first stored locally to prevent data loss and then transferred to the RTAC for downloading via the built-in web server.

NOTE
The Trigger Time Alignment setting is set to RTAC Assertion and hidden by default. To change this setting to Module Assertion, select the Advanced Settings check box to view and modify the setting.

Inputs

The following voltage and current inputs are recorded in the COMTRADE files generated by the SEL-2245-42 modules:

- IA, IB, IC
- VA, VB, VC (Wye)
- VAB, VBC, VCA (Delta)

Oscillography Triggering

The AC Protection Module offers a very flexible trigger mechanism with extendable recording lengths. *Figure 2.128* illustrates the three recording length settings pertaining to the module output event file. The first setting, Oscillography Pre-Trigger Length, is the length in seconds of the prerecorded data prior to the initiating trigger of the event. The second setting, Minimum Oscillography Record Length, is the minimum total recording length of the event in seconds and is equivalent to the length of Pre-Trigger plus the length of Post-Trigger. The last setting, Maximum Oscillography Record Length, is the absolute maximum total recording length that the event will not exceed, based on the applied trigger and Extended Post-Trigger length.

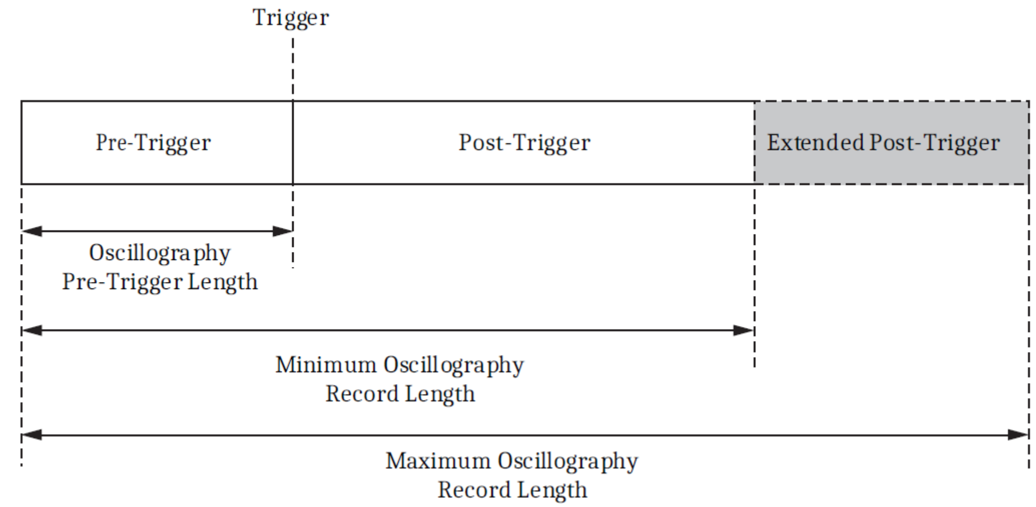


Figure 2.128 Event Record for the SEL-2245-42 AC Protection Module

By responding to either rising-edge or level triggers, the AC Protection Module provides flexibility for recording faults by using a variety of trigger techniques. *Figure 2.129* illustrates the basic rising-edge trigger implementation for the module when the time between triggers is greater than the minimum oscillography record length setting. In this simple rising-edge trigger example, the module generates two separate event files in response to two different rising-edge trigger occurrences. Each event file length matches the minimum oscillography record length setting without any Extended Post-Trigger data.

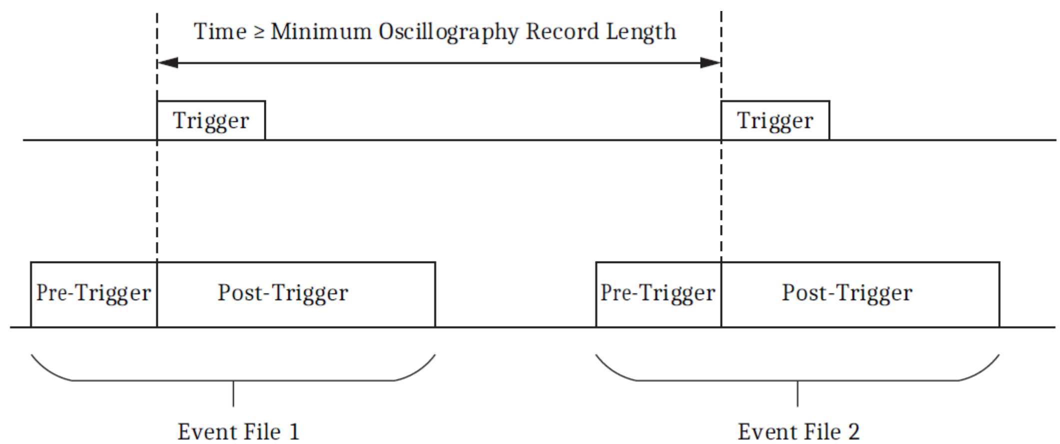


Figure 2.129 Simple Rising-Edge Triggered Event Recordings

Figure 2.130 illustrates another example with two rising-edge triggers, but in this case the second rising-edge trigger occurs before the minimum oscillography length expires. This action extends the length of event recording by the length of Post-Trigger time at the instance of the second trigger. This recording extension can continue with subsequent triggers until either the triggers stop or the total length of the event recording reaches the maximum oscillography record length.

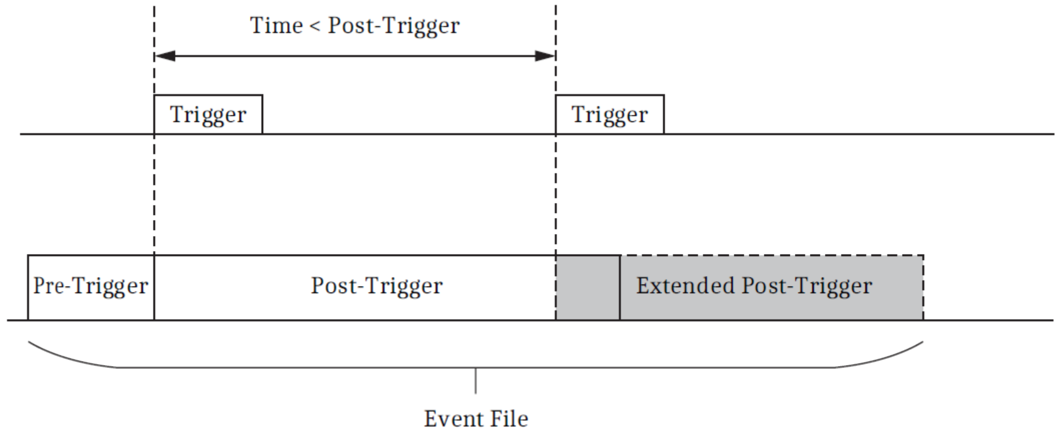


Figure 2.130 Rising-Edge Trigger Extended Event Recording

Figure 2.131 illustrates another example with a level trigger that extends the length of the recording. The level trigger extends the length of the event past the minimum oscillography record length setting. If the level trigger stays high, the module will continue to record until it reaches the maximum oscillography record length setting.

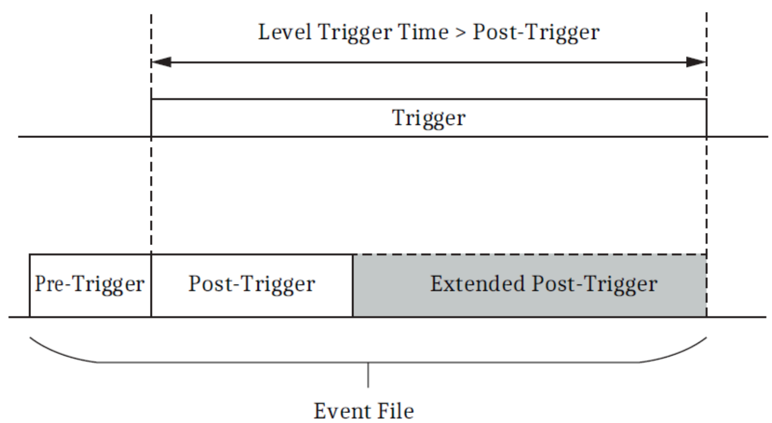


Figure 2.131 Level Trigger Extended Event Recording

Table 2.67 Oscillography Settings

Setting	Description
OscillographyTrigger	On rising edge, generates a waveform trigger event
OscillographyCaptureRate	The capture rate of oscillography data: 1, 2, 4, 8, or 24 kHz
OscillographyRecordLengthMin	Minimum length of an oscillography capture
OscillographyPreTriggerLength	Length of pre-trigger time
OscillographyRecordLengthMax	Maximum length of oscillography capture (in case of multiple trigger events or extended trigger): 24 s at 24 kHz 72 s at 8 kHz 144 s at 4 kHz 288 s at 2 kHz 576 s at 1 kHz

Setting	Description
StationName	RTAC hostname
ModuleID	Module identifier (module name in ACSELERATOR RTAC)
Module_name.Event_Trigger	Boolean input that triggers waveform recording in the module and only responds to rising edges

Downloading Events Via the Web Interface

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **https://** followed by the RTAC Ethernet address.

Step 4. Log in to the RTAC web interface.

Step 5. Click **Event Collection** under **Reports** in the navigation pane on the left.

Step 6. Sort events by selecting a time frame from **Timestamp Query** and the device filter.

Step 7. Select the desired events and click **Download**. You can also delete events by clicking **Delete**.

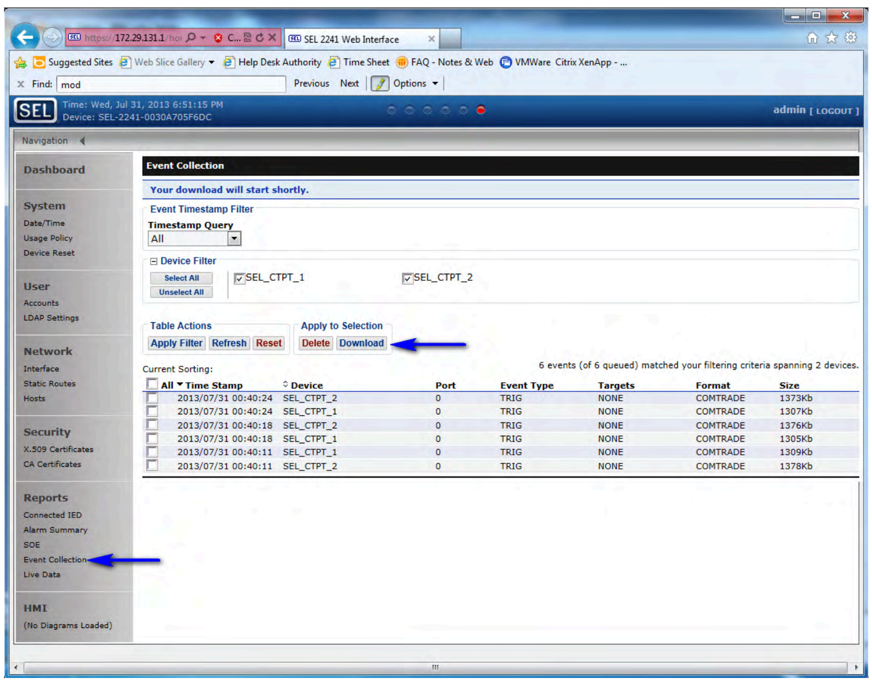


Figure 2.132 Waveform Record Retrieval

Viewing Waveforms and Event Files

Events files created by the SEL-2245-42 are in COMTRADE format. Event viewing software such as SYNCHROWAVE Event is required to open these files.

Figure 2.133 SYNCHROWAVE Event Display Waveform

Configuring Axion Recording Groups

NOTE
Recording Groups configured at 24 kHz for 24 seconds on the SEL-2241, SEL-3530, or SEL-3530-4 with more than two SEL-2245-42 Modules will time out before they are complete. An SEL-3555 RTAC should be used for any systems larger than two Modules at 24 kHz for 24 seconds.

Configure recording groups with the Axion for the purpose of combining multiple event reports from SEL-2245-42 AC Protection Modules with digital I/O from SEL-2244 Digital I/O modules in a single COMTRADE file. Additionally, you can include custom channels from the RTAC logic engine in a recording group. A single recording group can be configured for the entire Axion or as many as six separate recording groups can be configured based on how the recording channels make logical sense for analysis. Each recording group generates a single COMTRADE file with the data included for simplified analysis.

Step 1. Select the Recording Groups icon in the Insert menu in ACSELERATOR RTAC.

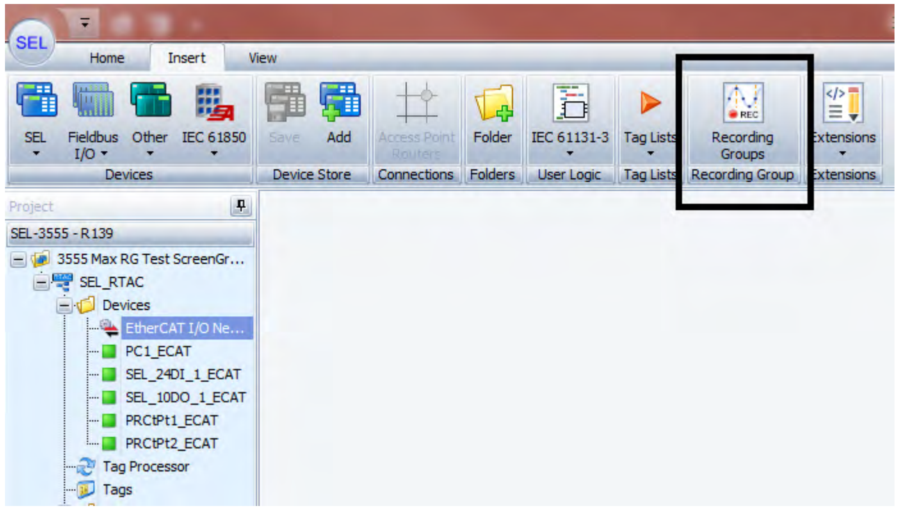


Figure 2.134 Insert a New Recording Group

Step 2. Enter a name for the new Recording Group or use the default, RecordingGroupX. Select the SEL-2245-42 and SEL-2244 modules you wish to include in the recording group.

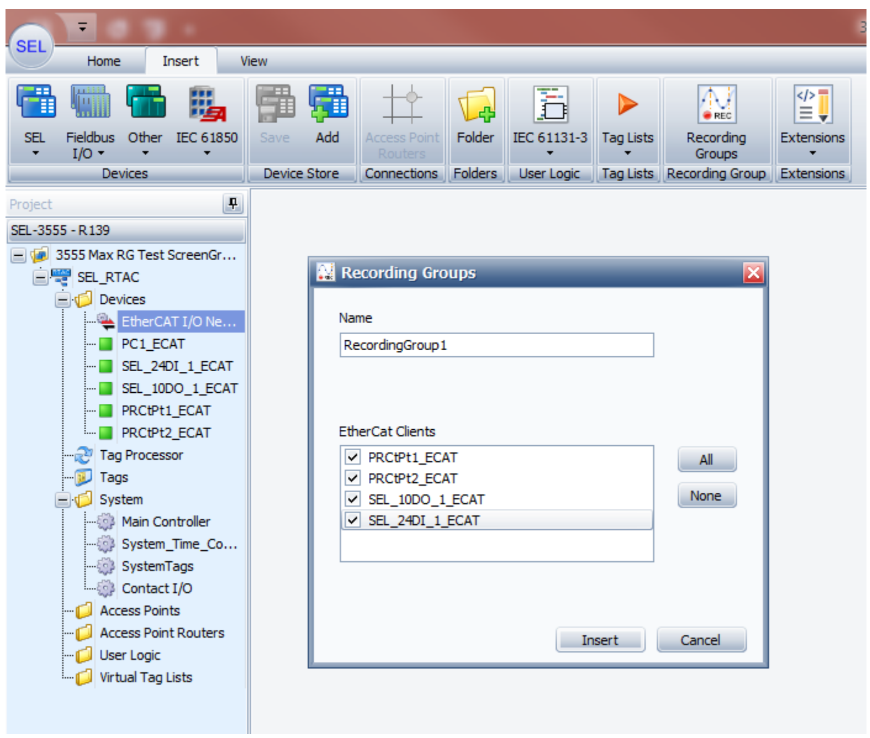


Figure 2.135 Select Modules for Recording Group

Step 3. Select the new RecordingGroupX and go to the **Settings** tab to configure the recording parameters for the recording group.

a. Set the **Station Name** and **Company Name** as desired to indicate your company and the location of the RTAC.

b. Configure the **Oscillography** settings as desired to set the recording lengths and capture rate.

These settings will override any previous Oscillography settings that were configured for all individual SEL-2245-42 Protection modules included in RecordingGroupX.

Figure 2.136 Configure Recording Group Oscillography Settings

Step 4. The **EtherCAT Modules** tab lists the modules included in RecordingGroupX. You can modify the modules included in the recording group here if necessary.

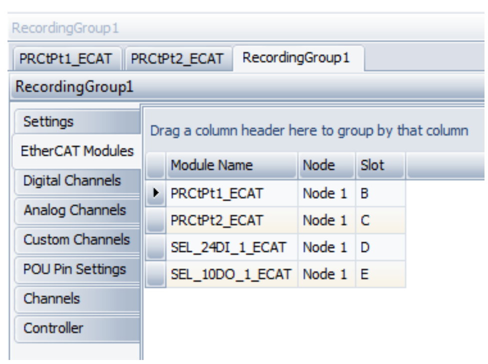


Figure 2.137 EtherCAT Modules Included in the Recording Group

Step 5. Go to the **Digital Channels** tab and configure the **Channel Name** for the inputs or outputs of the SEL-2244 modules included in RecordingGroupX, if desired. Additionally, if there are inputs or outputs you do not wish to include in the COMTRADE file, disable them here.

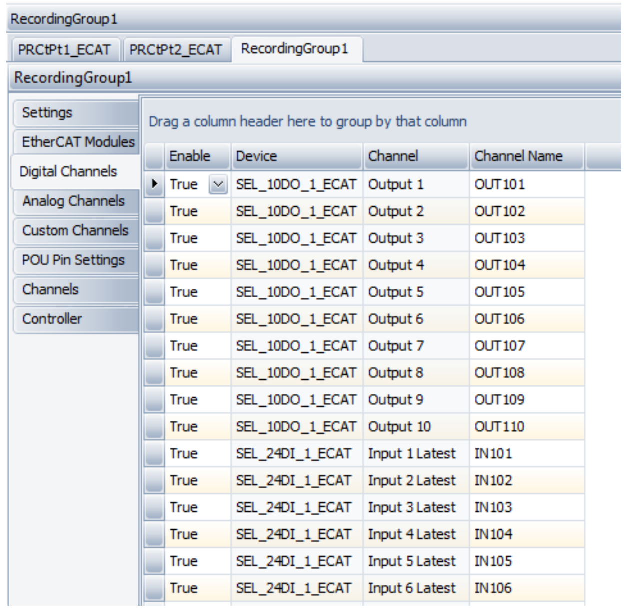


Figure 2.138 Digital Channel Settings

Step 6. Go to the **Analog Channels** tab and configure the **Channel Name** for the analog inputs of the SEL-2245-42 modules included in RecordingGroupX if desired. Additionally, if there are inputs you do not wish to include in the COMTRADE file, you can disable them here.

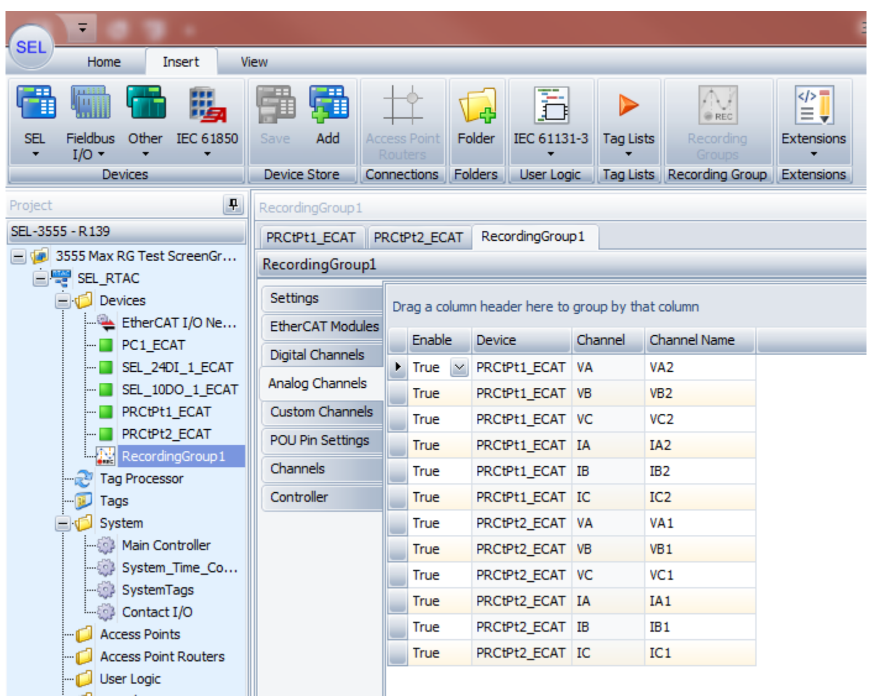


Figure 2.139 Analog Channel Settings

Step 7. Go to Custom Channels and add the number of custom digital or analog tags you wish to add from the RTAC logic engine to include in RecordingGroupX.

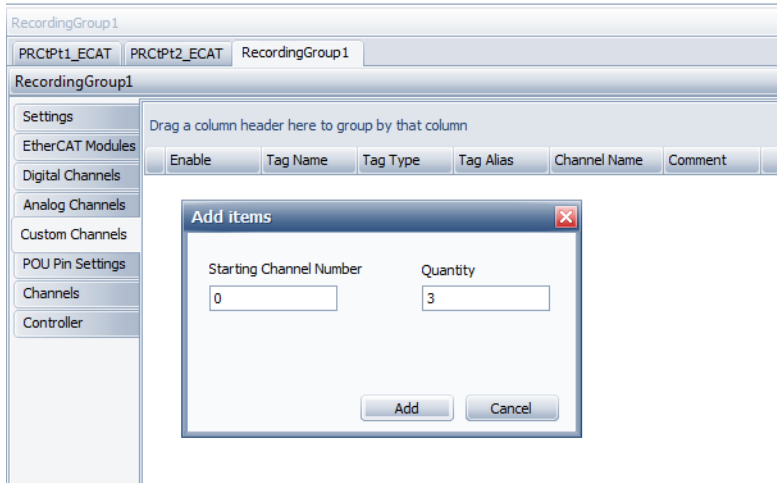


Figure 2.140 Add Custom Channels

Step 8. Configure **Tag Name**, **Channel Name**, and **Tag Alias** for the custom channels included in RecordingGroupX as desired. Additionally, if there are inputs you do not wish to include in the COMTRADE file, disable them here. Go to **Tag Processor** and map source tags from the RTAC logic engine to the new RecordingGroupX.CustomDigitalX or RecordingGroupX.CustomChannelX tags.

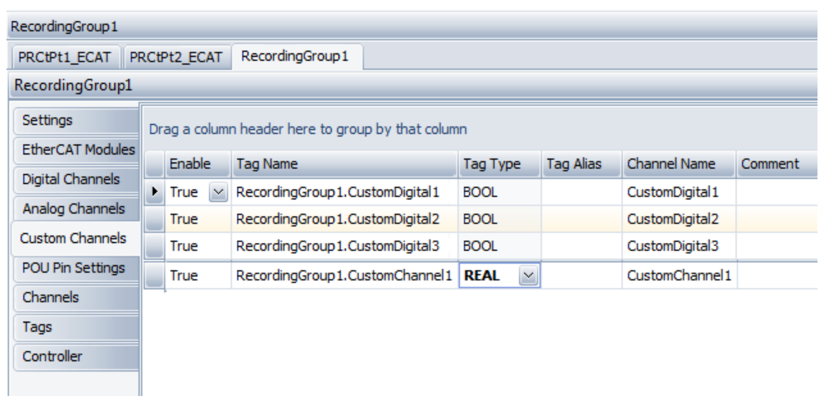
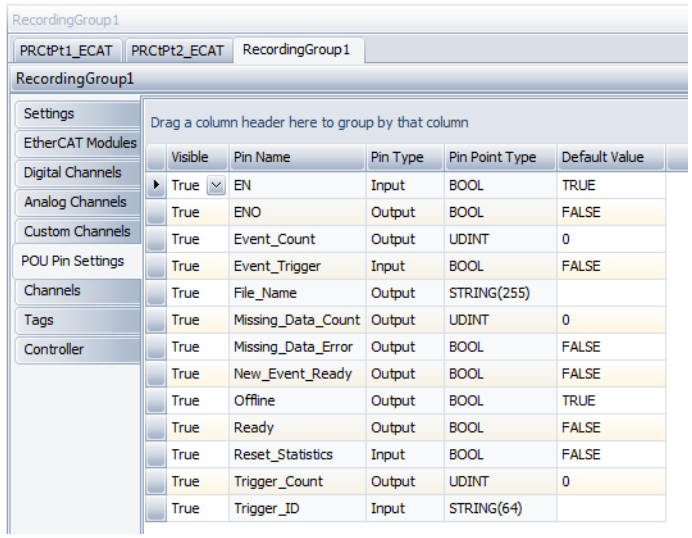


Figure 2.141 Custom Channel Settings

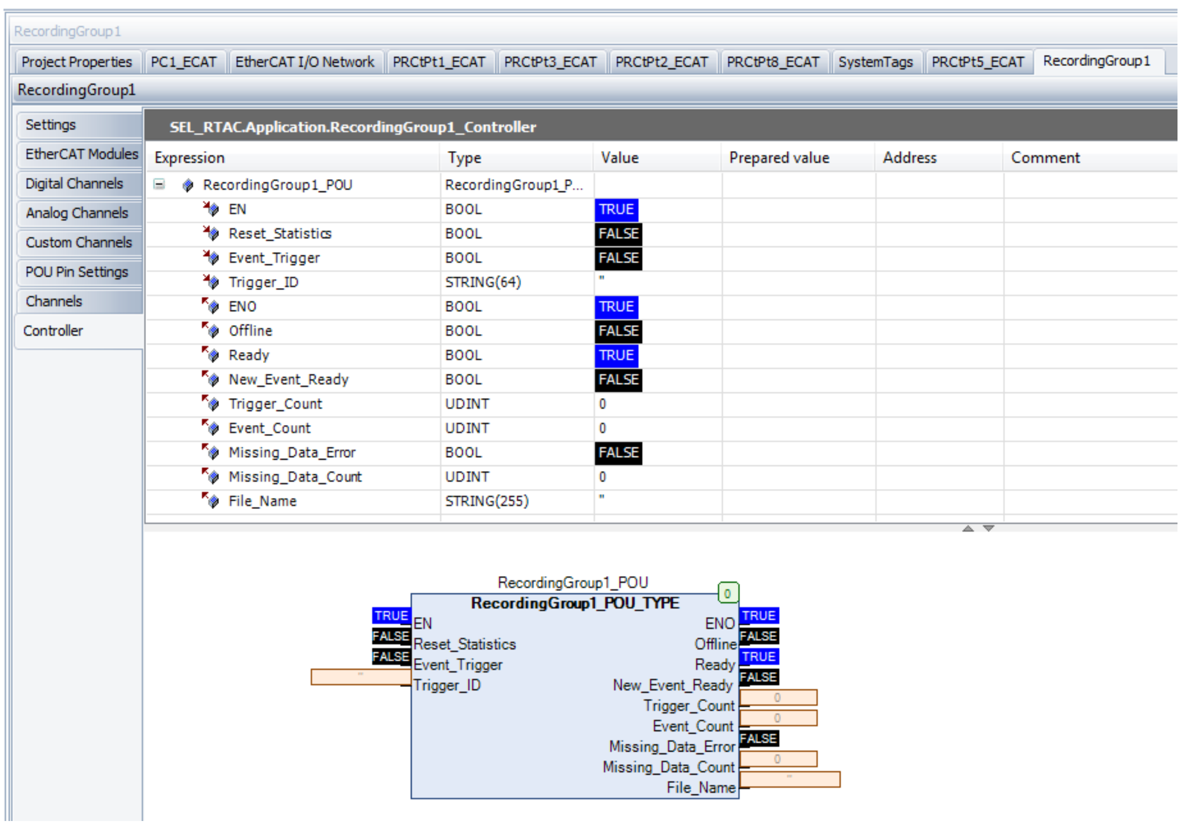
Figure 2.142 shows the POU Pin Settings with default values for RecordingGroupX.



Visible	Pin Name	Pin Type	Pin Point Type	Default Value
True	EN	Input	BOOL	TRUE
True	ENO	Output	BOOL	FALSE
True	Event_Count	Output	UDINT	0
True	Event_Trigger	Input	BOOL	FALSE
True	File_Name	Output	STRING(255)	
True	Missing_Data_Count	Output	UDINT	0
True	Missing_Data_Error	Output	BOOL	FALSE
True	New_Event_Ready	Output	BOOL	FALSE
True	Offline	Output	BOOL	TRUE
True	Ready	Output	BOOL	FALSE
True	Reset_Statistics	Input	BOOL	FALSE
True	Trigger_Count	Output	UDINT	0
True	Trigger_ID	Input	STRING(64)	

Figure 2.142 RecordingGroupX POU Pin Settings

Figure 2.143 shows the Controller with online values for RecordingGroupX.



Expression	Type	Value	Prepared value	Address	Comment
RecordingGroup1_POU	RecordingGroup1_P...				
EN	BOOL	TRUE			
Reset_Statistics	BOOL	FALSE			
Event_Trigger	BOOL	FALSE			
Trigger_ID	STRING(64)	''			
ENO	BOOL	TRUE			
Offline	BOOL	FALSE			
Ready	BOOL	TRUE			
New_Event_Ready	BOOL	FALSE			
Trigger_Count	UDINT	0			
Event_Count	UDINT	0			
Missing_Data_Error	BOOL	FALSE			
Missing_Data_Count	UDINT	0			
File_Name	STRING(255)	''			

Figure 2.143 RecordingGroupX Controller

A rising edge to Event_Trigger simultaneously starts recording of the individual COMTRADE files in each SEL-2245-42 module included in RecordingGroupX. Additionally, it records the digital tags at the RTAC processing interval. If SOE tags from the SEL-2244 Digital I/O modules are included in RecordingGroupX, the assertions that occur between the RTAC processing interval will be included in the combined COMTRADE file for RecordingGroupX.

Axion Wave Server

Overview

The Axion Wave Server is a phasor data concentrator (PDC) that uses the IEEE C37.118 protocol to stream 3 kHz, point-on-wave measurements from SEL-2245-42 AC Protection Modules. It supports Configuration 2 data frames and transmits 3000 messages per second over TCP/IP for as many as 96 Channels. Each SEL-2245-42 AC Protection Module in the configuration functions as a phasor measurement unit (PMU) that provides measurement samples from its AC analog inputs to the Axion Wave Server PDC.

System Components

The Axion Wave Server requires the following components:

- SEL-3555 RTAC or SEL-3560
- SEL-2242 Backplane(s)
- SEL-2243 Power Coupler(s)
- SEL-2245-42 AC Protection Module(s)

Configuration

Step 1. Starting with a new SEL-3555 or SEL-3560 RTAC project, configure the SEL Axion EtherCAT network.

Step 2. Add the Axion Wave Server by selecting it under **SEL Axion** on the **Insert** ribbon.

Step 3. Set the **Server IP Port** in the **Settings** tab. Server IP Port defaults to 4712.

Step 4. Enable the channels in the **Device** tab. For each SEL-2245-42 AC Protection Module, set **Enable Voltages** to **True** to enable voltage channels VA, VB, and VC; and set **Enable Currents** to **True** to enable current channels IA, IB, and IC.

Step 5. Set the PMU's **Station Name** and **PMUID** for each device using the SEL-2245-42 AC Protection Module's settings.

Node Connections Editor

You must configure the EtherCAT network before you can download a new project to an RTAC that contains SEL-2243 Power Couplers, SEL-2244 I/O modules, and SEL-2245 I/O modules. The ACSELERATOR RTAC project must include specific information about the placement of modules within an SEL-2240 node and the network connections between nodes. The Node Connections Editor provides a simple interface you will use to provide this information. Click on the EtherCAT I/O network entry in the project view, as shown in *Figure 2.144*, to access the Node Connections Editor.

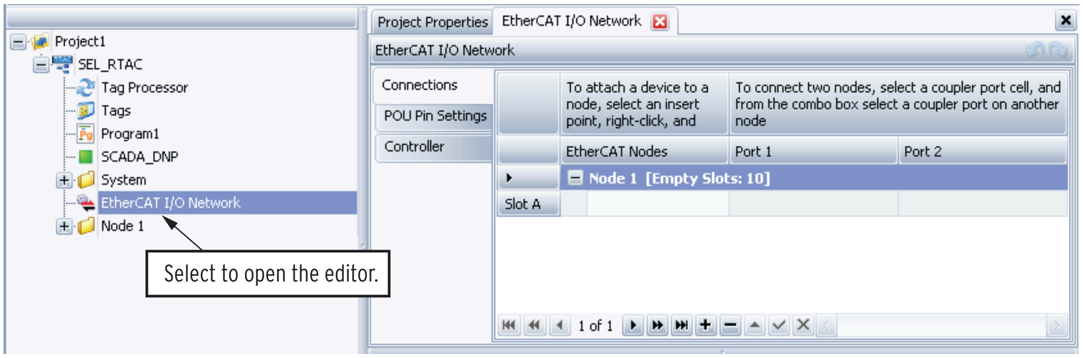


Figure 2.144 Opening the Node Connections Editor

The editor has three tabs: **Connections**, **POU Pin Settings**, and **Controller**. See *POU Pin Settings (Advanced Usage) on page 37* for a description of the **POU Pin Settings** tab. See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab. In online mode, the Node Connections Editor provides valuable diagnostics about the operation and state of the EtherCAT network. Refer to *Section 3: Testing and Troubleshooting* in the *SEL-2240 Instruction Manual* for details.

Controlling EtherCAT Response to Network Errors

There are two options available for operating the EtherCAT network in case of network errors or I/O module failure. By default, all EtherCAT network traffic will shut down, and outputs will return to de-energized state, if the RTAC receives message errors for 25 ms. Optionally, the Axion system can be configured so that the network will remain in operation even if some portions of the network become unavailable.

Control this operation by using a POU pin setting for the EtherCAT I/O Network. View and change the selection, using the following steps in ACSELERATOR RTAC.

Step 1. Select **EtherCAT I/O Network** from the project view pane.

Step 2. Click **POU Pin Settings**.

Step 3. As shown in *Figure 2.145*, find the present value for the setting Disable_On_Network_Error. The value will be TRUE initially for each new project. Change the value to FALSE if you prefer that the network continue to operate even if there are network errors or I/O module failures.

Step 4. Save and download the project.

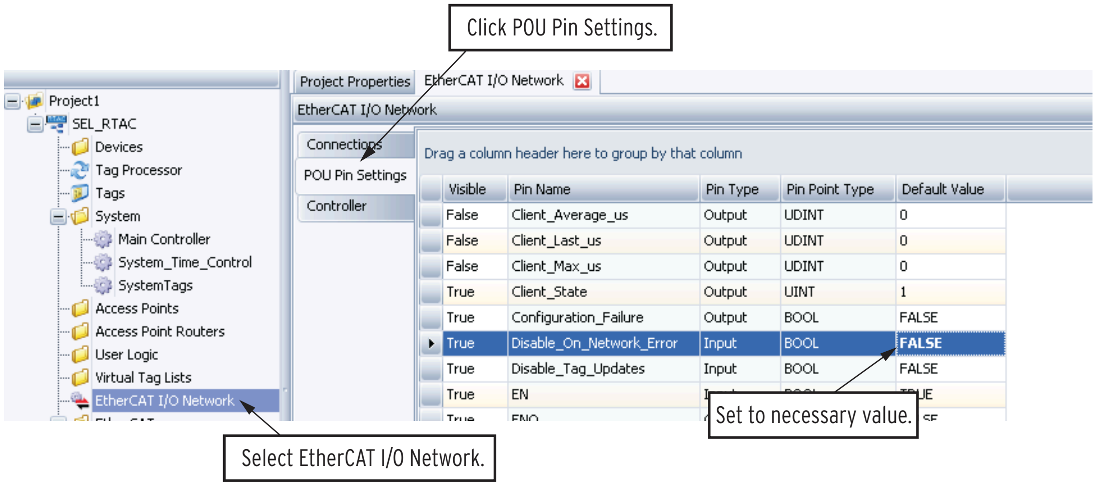


Figure 2.145 Configuring Disable on Network Error Setting

Connections

The **Connections** tab provides a tabular view of the nodes and modules in the project. When you open the Node Connections Editor for the first time, there will be an entry for a single node named Node 1 (you can rename this at any time). Next to the node name you will see the number of empty slots in that node. The primary editor actions are as follows:

- Attaching modules to a node
- Adding a node
- Editing the network connection between nodes
- Renaming a node
- Deleting a node
- Moving modules

NOTE

For ease of initial project configuration and troubleshooting, we recommend that you add only one node at a time to a project. Download and verify EtherCAT network operation prior to adding subsequent nodes.

NOTE

If a node does not include an SEL-2241, then a power coupler should be the initial module in a node. This would be the case if you are using an external SEL-3530 RTAC.

NOTE

If you select multiple modules in the dialog box, ACSELERATOR RTAC will list them in the node in the same sequence they appeared in the dialog box.

NOTE

All RTAC modules in the module list have an "[RTAC]" suffix, and all power couplers have a "[Coupler]" suffix. These suffixes will help you quickly identify these types of modules, regardless of module names.

Attaching Modules to a Node

Modules must be listed in the Node Connections Editor such that the listing matches their physical placement in the node. The EtherCAT network cannot autoconfigure and initialize upon downloading the project if the modules are not listed accurately. If you have an SEL-2241 RTAC module in a node, it must be the initial module added for that node. All subsequent modules must be added so that there are no blank slots between modules in the node. You do not need to fill every slot, but all modules must be installed sequentially and without gaps. As shown in *Figure 2.146*, follow these steps to add the RTAC module to Slot A.

Step 1. Right-click on the empty slot where the module resides.

Step 2. Select **Attach Devices**.

Step 3. The dialog box that appears allows you to select one or more available modules from your project. The selection for an SEL-2241 will only be available so long as no other SEL-2241 has been configured in the project and no network connection has been designated for an external RTAC.

Step 4. Select the SEL-2241 (RTAC) and press **Attach**.

Step 5. In the resulting editor window, the SEL-2241 displays in Slot A and the number of empty slots now shows 9.

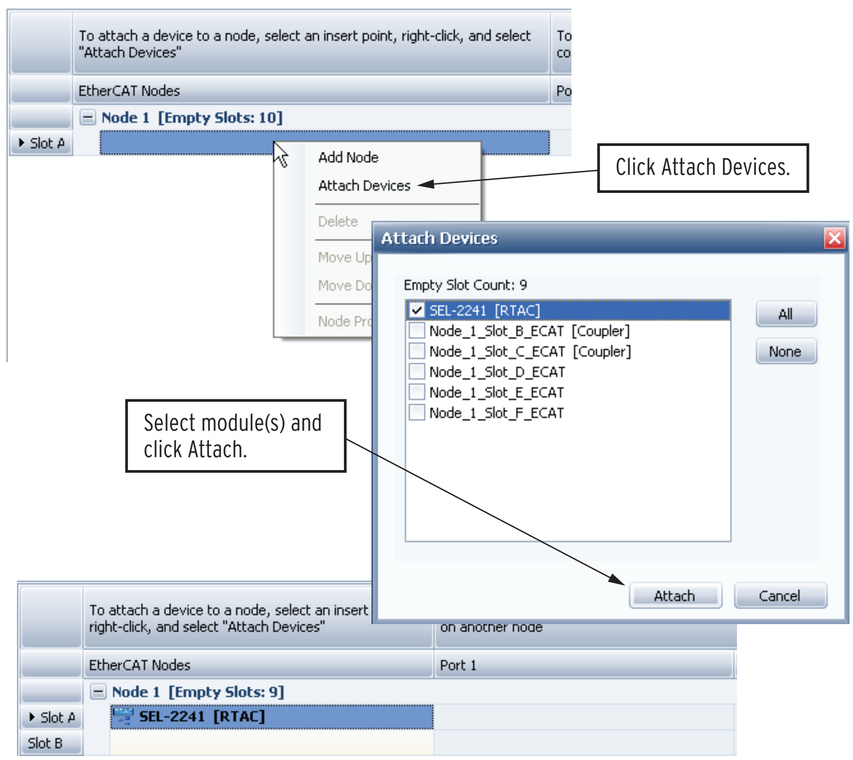


Figure 2.146 Attach a Device to a Node

Adding a Node

There are two methods for adding a new node to the project. First, as shown at the bottom of *Figure 2.147*, select the + control at the bottom of the editor. Second, right-click on any row to select **Add Node**.

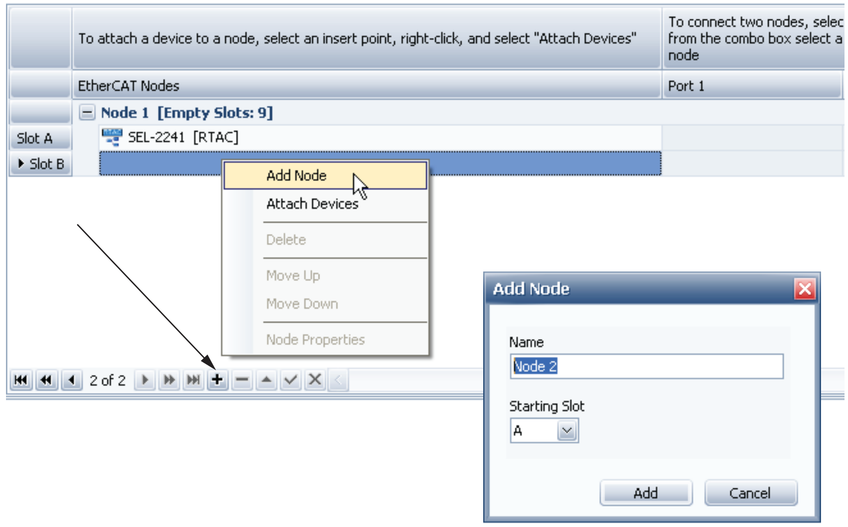


Figure 2.147 Adding a Node to the Project

After you use either selection method, you will see the **Add Node** dialog box. Provide a custom node name, if necessary. There is also a selection for the starting slot of a node. In most cases, the default selection of Slot A will be sufficient. Finally, click the **Add** button to accept a selection.

Editing Internode Connections

The automatic EtherCAT network configuration the RTAC performs during network initialization is dependent not only on accurate information about the sequence of modules within a node, but also on accurate information about the network cabling between nodes. The Axion system supports a number of different network configurations. As you will see, the Node Connections Editor will only allow you to enter settings for a valid network; it will also be helpful for your understanding of the basic networking rules for connections between nodes.

The SEL-2243 Power Couplers provide two Ethernet ports dedicated for EtherCAT. These ports are labeled **PORT 1** and **PORT 2**, respectively. For the following figures, we have used IN and OUT labels to denote the ports operating in the IN mode and the OUT mode for EtherCAT. An **IN** port is the port by which a node is connected to an upstream node or an external RTAC and is designated in the figures by a label of IN. **PORT 2** in a power coupler never operates as an **IN** port. **OUT** ports make a connection to a downstream node and are designated in the figures by a label of **OUT**. Each figure illustrates only a portion of the nodes.

As an initial example, consider a system using an external RTAC and multiple nodes, all with single power coupler modules, as shown in *Figure 2.148*. **PORT 1** of Node 1 is connected to the RTAC and is the **IN** port for Node 1. **PORT 2** is the **OUT** port for Node 1 and is connected to **PORT 1** of Node 2. Finally, **PORT 2** of Node 2 is connected to **PORT 1** of Node 3. You can continue this sequential networking method for more nodes, as necessary (as much as the maximum module limit).

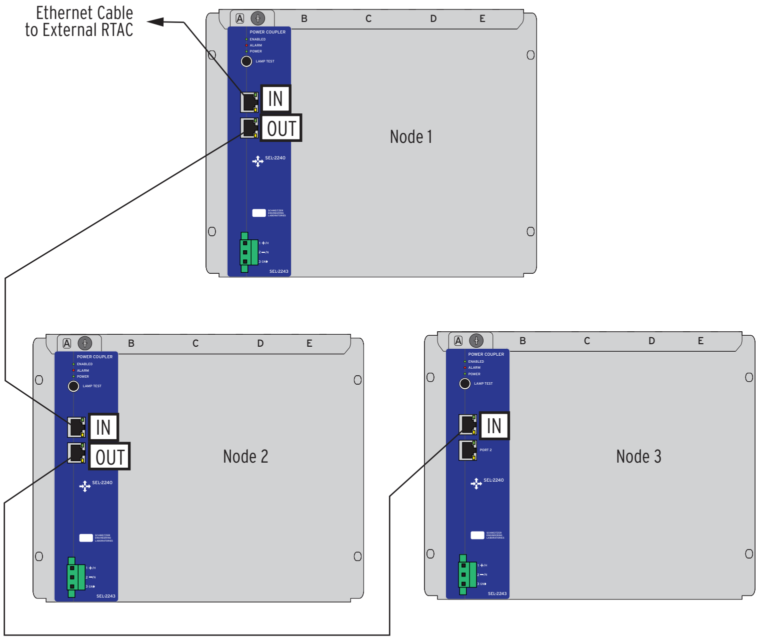


Figure 2.148 Axion System Using an External RTAC and Single Power Couplers

Figure 2.149 illustrates how to begin configuring this system in the Node Connections Editor by identifying the IN connection to an external RTAC. In the figure, notice that we have already created three nodes and attached a power coupler to Slot A of each node. Select the entry cell for PORT 1 of the Node 1 coupler to see the valid input connections. No other node contains an RTAC module connection, so ACSELERATOR RTAC will allow IN connections from an external RTAC, Node 2, or Node 3 at this point. Click on **[IN] SEL-3530 [RTAC]**.

Next, set up the connection between Node 1 and Node 2. As you can see in *Figure 2.150*, you will need to select the Port 2 cell for Node 1 and click the entry for **[OUT] Node 2, Node_2_Slot_A_ECAT, Port 1**. The segments of this entry have the following meanings:

[OUT]—The EtherCAT function of the selected port, either IN or OUT.

Node 2—The name of the node to which you are connecting. This is the node name as listed in the Node Connection Editor; it does not reference a folder or device name in the project view.

Node_2_Slot_A_ECAT—The module name of the power coupler to which you are connecting. You must add power couplers to the project view before the power couplers are available options in the Node Connection Editor.

Port 1—The physical port on the power coupler to which you are connecting.

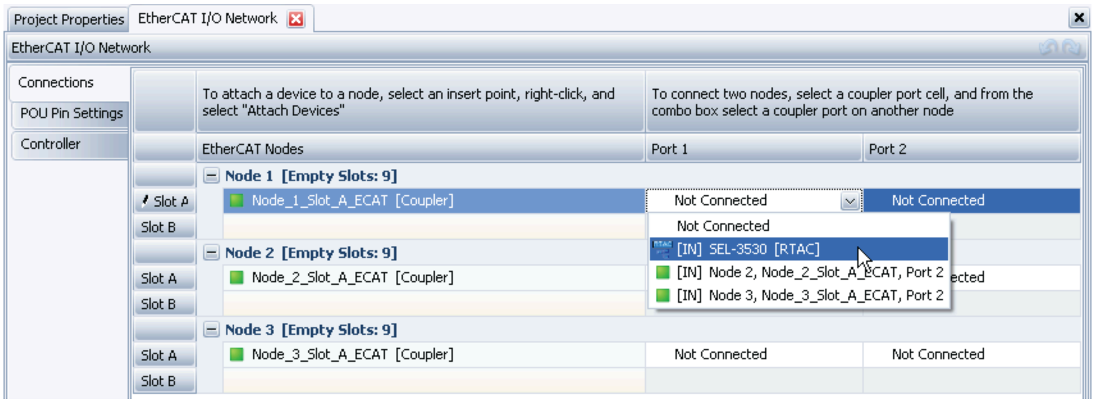


Figure 2.149 Setting an Input Connection to an External RTAC

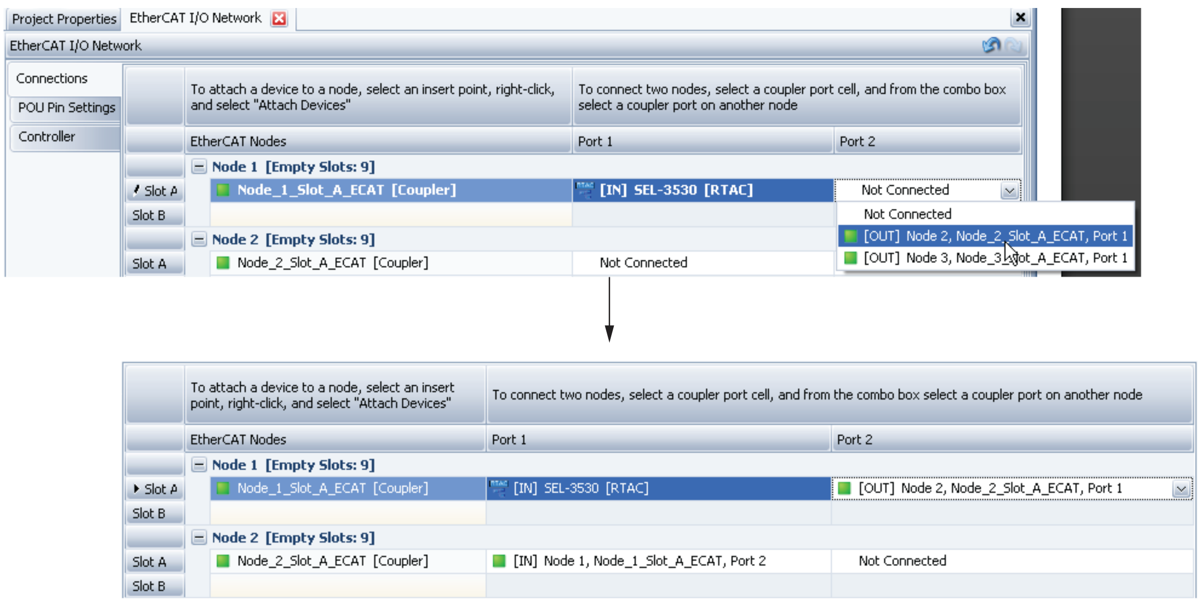


Figure 2.150 Connection From Node 1 to Node 2

Notice in the lower portion of *Figure 2.150* that ACSELERATOR RTAC automatically filled in the corresponding entry in Node 2 to complete the connection. Finally, configure the connection from Node 2 to Node 3. In this case, as shown in *Figure 2.151*, we selected the "IN" port (**PORT 1** on Node 3). The drop-down menu only lists available connections; we already have an RTAC connection in the project, so the list no longer includes an RTAC entry. Once again, the reciprocal entry for **PORT 2** on Node 2 is automatically created, as shown in the lower portion of *Figure 2.151*.

NOTE

ACSELERATOR RTAC validates settings in the Node Connection Editor each time the project is saved. Validation does not occur automatically when a setting changes.

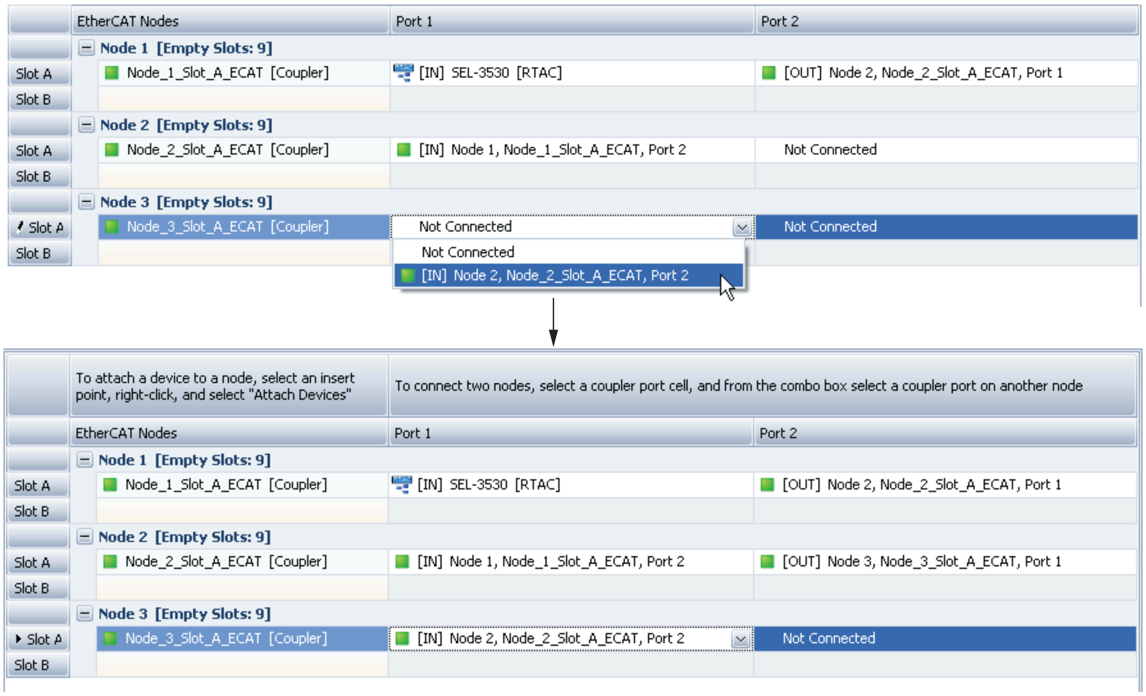


Figure 2.151 Connection From Node 2 to Node 3

For the next example, the system includes an SEL-2241 RTAC module and three nodes, all with a single power coupler. We can configure a network of this type in a sequential manner similar to the previous example. In this case, however, we used a star arrangement as shown in Figure *Figure 2.152*. When you use an SEL-2241 RTAC rather than an external RTAC, the first power coupler in the node containing the RTAC uses the backplane connector as an **IN** port for EtherCAT. Therefore, you can use both ports on the first power coupler in "OUT" mode.

Figure 2.153 displays the Node Connection Editor results for configuring this system. An SEL-2241 RTAC resides in Slot A of Node 1. Both **OUT** ports on the Node 1 power coupler connect with the respective **IN** ports on Node 2 and Node 3.

NOTE
The Node Connection Editor grid is grayed and unavailable for slots that do not contain a power coupler.

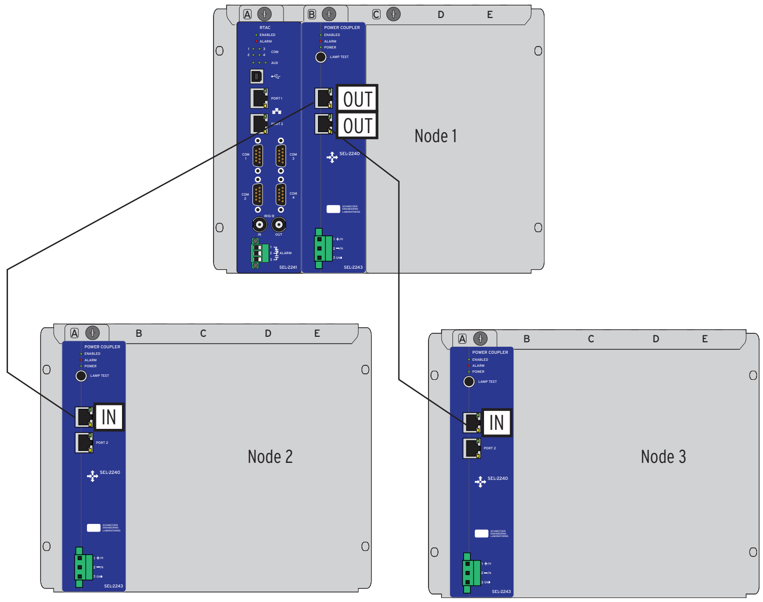


Figure 2.152 Axion System Using an SEL-2241 RTAC and Single Power Couplers

	To attach a device to a node, select an insert point, right-click, and select "Attach Devices"	To connect two nodes, select a coupler port cell, and from the combo box select a coupler port on another node	
	EtherCAT Nodes	Port 1	Port 2
	Node 1 [Empty Slots: 8]		
Slot A	SEL-2241 [RTAC]		
Slot B	Node_1_Slot_B_ECAT [Coupler]	[OUT] Node 2, Node_2_Slot_A_ECAT, Port 1	[OUT] Node 3, Node_3_Slot_A_ECAT, Port 1
Slot C			
	Node 2 [Empty Slots: 9]		
Slot A	Node_2_Slot_A_ECAT [Coupler]	[IN] Node 1, Node_1_Slot_B_ECAT, Port 1	Not Connected
Slot B			
	Node 3 [Empty Slots: 9]		
Slot A	Node_3_Slot_A_ECAT [Coupler]	[IN] Node 1, Node_1_Slot_B_ECAT, Port 2	Not Connected
Slot B			

Figure 2.153 Node Connection Editor Results for System in Figure 2.152

The next two examples deal with systems using redundant power couplers. In all cases, each node can have only one port functioning as an IN port. PORT 1 in the first power coupler will function as the IN port for every node except a node containing an SEL-2241. As we stated previously, the Node Connection Editor helps ensure that each setting is valid.

In *Figure 2.154* you will see a system connected to an external RTAC that uses dual power couplers. This is another example of a star network, with Nodes 2, 3, and 4 all connected to Node 1. *Figure 2.155* shows the settings for this system.

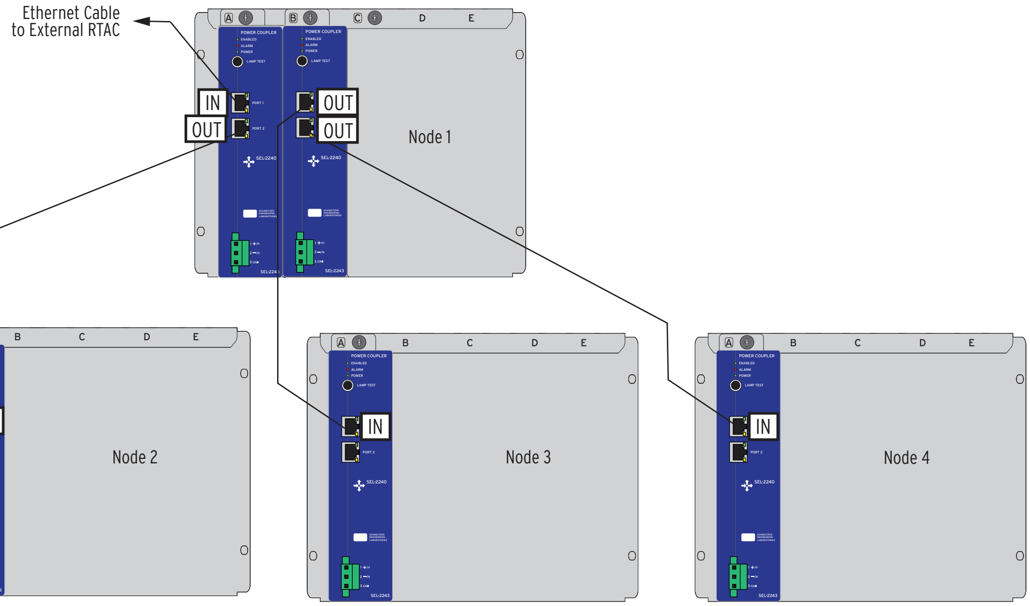


Figure 2.154 Axion System Using an External RTAC and Dual Power Couplers

	Node 1 [Empty Slots: 8]		
Slot A	Node_1_Slot_B_ECAT [Coupler]	[IN] SEL-3530 [RTAC]	[OUT] Node 2, Node_2_Slot_A_ECAT, Port 1
Slot B	Node_1_Slot_C_ECAT [Coupler]	[OUT] Node 3, Node_3_Slot_A_ECAT, Port 1	[OUT] Node 4, Node_4_Slot_A_ECAT, Port 1
Slot C			
	Node 2 [Empty Slots: 9]		
Slot A	Node_2_Slot_A_ECAT [Coupler]	[IN] Node 1, Node_1_Slot_B_ECAT, Port 2	Not Connected
Slot B			
	Node 3 [Empty Slots: 9]		
Slot A	Node_3_Slot_A_ECAT [Coupler]	[IN] Node 1, Node_1_Slot_C_ECAT, Port 1	Not Connected
Slot B			
	Node 4 [Empty Slots: 9]		
Slot A	Node_4_Slot_A_ECAT [Coupler]	[IN] Node 1, Node_1_Slot_C_ECAT, Port 2	Not Connected

Figure 2.155 Node Connection Editor Results for System in Figure 2.154

Figure 2.156 represents the final example. The system has an SEL-2241, two nodes with dual SEL-2243 Power Couplers, and a mixture of sequential and star connections. Notice that the star connections can be tiered, such as exemplified by Node 1 and Node 2. Also note that Node 2 has one star connection (Node 6) and one sequential connection (Node 7). You can mix and match the connections to match the application. *Figure 2.157* shows the configuration results for the system.

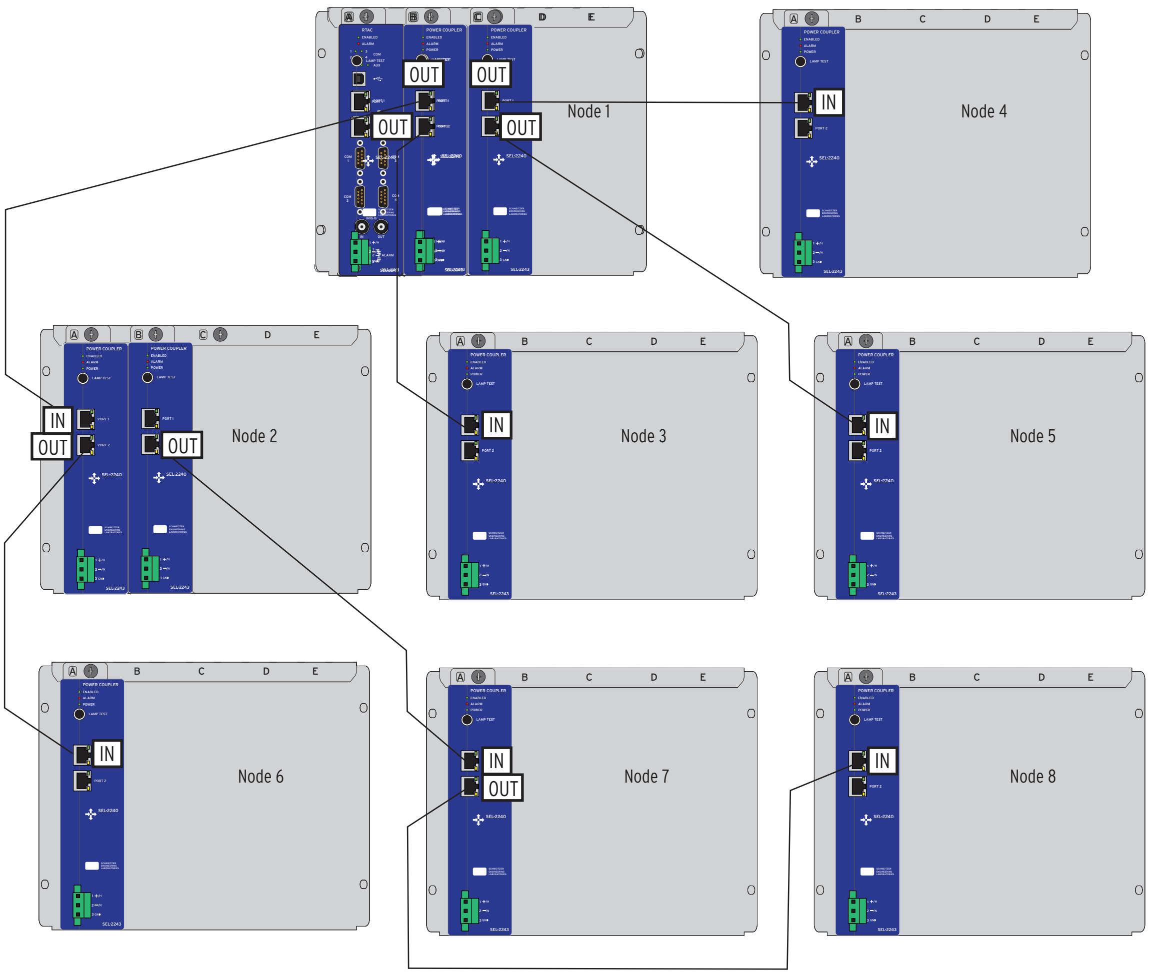


Figure 2.156 Axion System Using an SEL-2241 and Dual Power Couplers

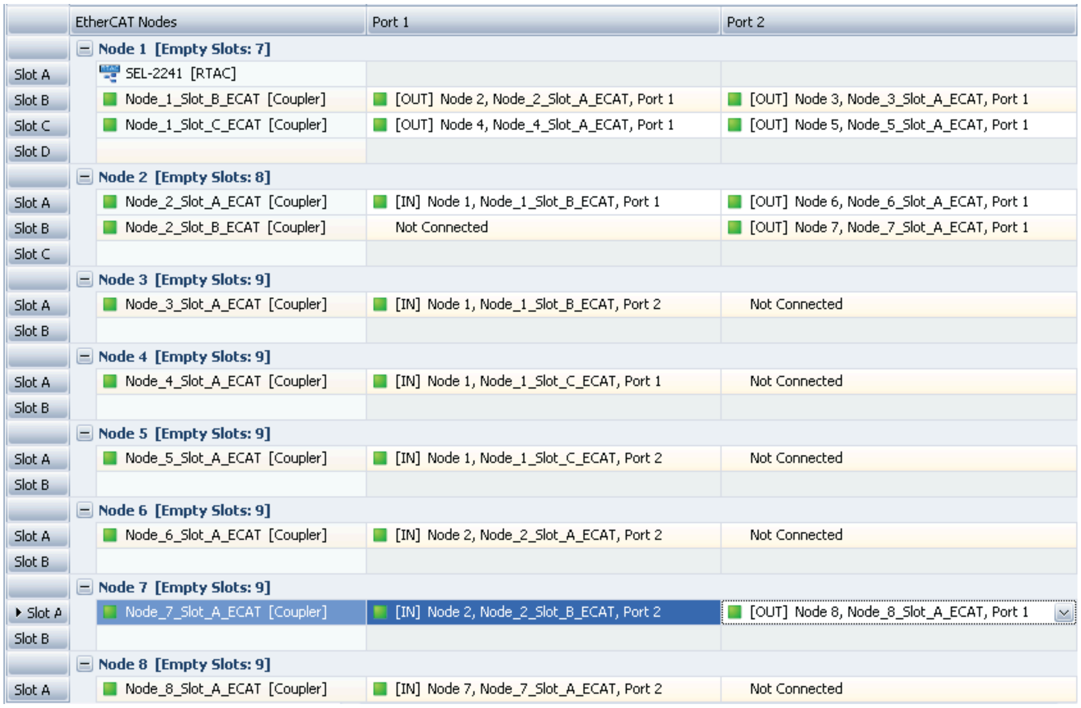


Figure 2.157 Node Connection Editor Results for System in Figure 2.156

Renaming a Node

Rename a node in the Node Connection Editor at any time via the Node Properties function. As shown in *Figure 2.158*, simply right-click on the node name and select **Node Properties** from the menu. Then, in the dialog box, type in the replacement name and click **Set** to complete the operation.

NOTE

Node names must be unique.

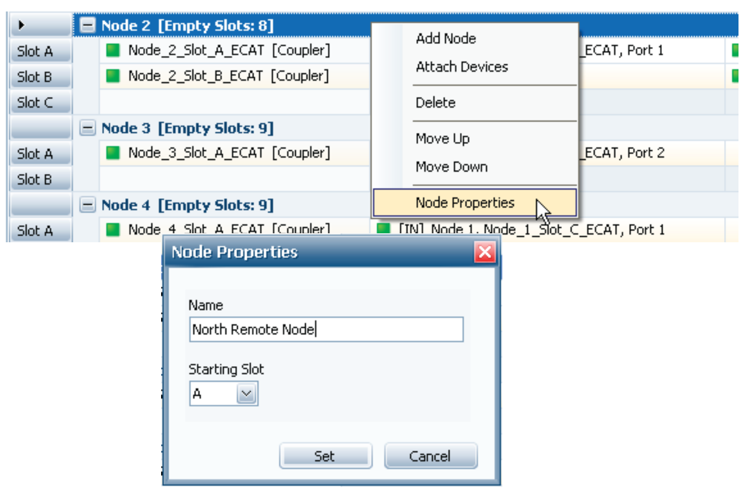


Figure 2.158 Renaming a Node

Deleting a Node

Delete unneeded nodes in the Node Connection Editor at any time. As with the renaming function, you can perform this deletion of unneeded nodes by right-clicking on the node name and selecting **Delete** from the menu. All of the connections referring to this node will also be deleted. However, the modules that are listed as part of that node are not deleted from the project. Attach the modules to the correct node by moving them prior to deleting the node within which they presently reside.

Moving Modules and Nodes

The Node Connection Editor has an intuitive drag-and-drop interface to change the sequence of nodes or modules in the view. Whether you need to change the sequence of modules within a node, move a module from one node to another, or change the displayed sequence of nodes, just select and drag the items to their new location. ACSELERATOR RTAC will not allow for multiple item selection in this view.

RTU Project Examples

There are two project templates in the **RTAC/Axion** RTAC type project category in the **Create Project** dialog box. These RTU projects gather data from one to three SEL Axion nodes and report to a DNP master. Most projects only contain digital inputs and outputs (I/O), but four projects include gathering of dc analog inputs (AI) from an SEL-2411 through IEC 61850 GOOSE messages. Most of the projects use RJ45 EtherCAT connectors for Power Couplers. The projects that have three nodes include variations with LC fiber-optic EtherCAT connectors for Power Couplers. The projects include 125 V digital input and Form A digital output modules.

Each project includes a DNP master connection for all connected I/O, and this connection includes dc AIs if the project includes them. Additionally, all digital I/O map to the internal SOE data for time-stamping of all inputs and outputs. The input debounce time is 5 ms on pickup and dropout.

Each project includes IEC 61131-3 example functions that are not active, but are available for use. These functions include a transfer trip function (TTF), assert digital output (ADO), square-wave generator (SWG), frequency meter (FM), and an analog input alarm (AI_Alm).

Table 2.68 RTU Templates With Included Modules

Project	Nodes	RTAC	PC Mod	DI Mod	DO Mod	AI Mod	CT/ PT Mod	Description
1	Node 1 Node 2 and 3 Node 4	2241	2 RJ45 2 RJ45 2 RJ45	4 5	3 3	2	2	336 DI/144 DO/32 AI/8 CT/8 PT Dual-Redundant RJ45
2	Node 1 Node 2	2241	2 RJ45 2 RJ45	4	4	2	1	96 DI/48 DO/32 AI/4 CT/4 PT Dual-Redundant RJ45

Setting Modifications

There are several settings you may want to change from their defaults to obtain the necessary performance.

- *RTAC Update Rate on page 240*
- *Input Debounce Settings on page 240*
- *SOE Logging Settings on page 241*
- *DNP IP Settings on page 241*
- *IEC 61131-3 Example Function Use on page 242*
- *IEC 61850 GOOSE Analog Settings on page 242*

RTAC Update Rate

The RTAC task cycle time is set at 50 ms in the RTU projects. Adjust this setting in the **Main Controller Options** according to how often you want the information in the RTAC to refresh. To optimize CPU utilization, try to avoid setting the task rate at the fastest settings.

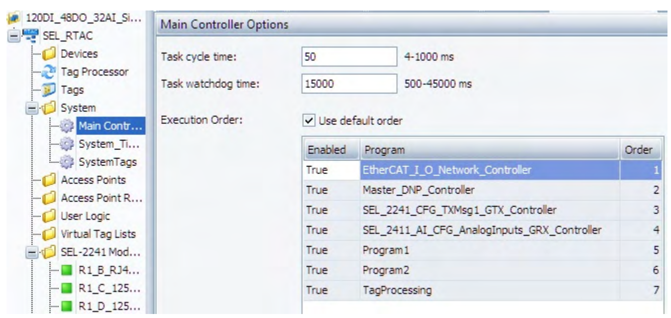


Figure 2.159 RTAC Update Rate

Input Debounce Settings

The input debounce settings are set to 5 ms for pickup and dropout in dc mode. Adjust these settings to the optimum debounce time to match your output contacts, or change to ac mode if you are using ac wetting voltage. Refer to *Digital Input Debounce on page 164* for detailed information.

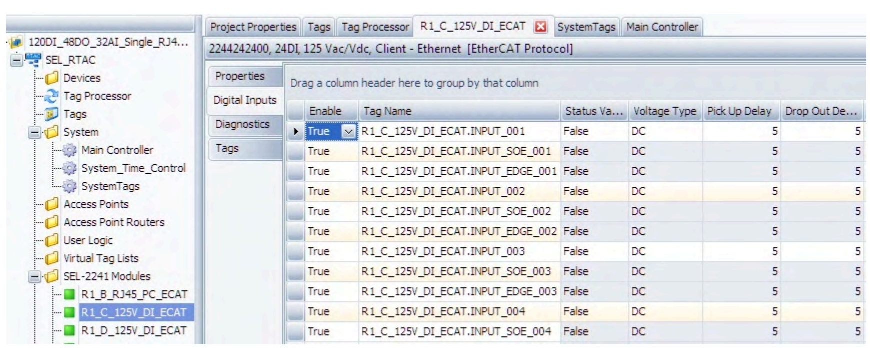


Figure 2.160 Input Debounce Settings

SOE Logging Settings

The Tag Processor contains configuration for SOE logging of all digital inputs and outputs in the projects. Configure logging chatter filters, logging messages, or enable/disable logging for the SOE points. See *Log Settings on page 453* for detailed instructions.

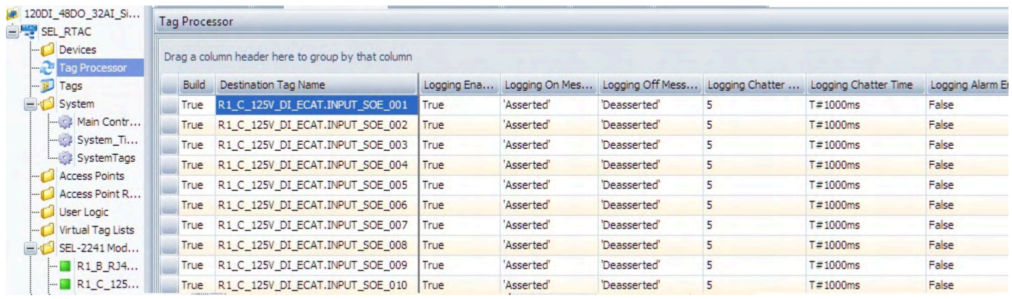


Figure 2.161 SOE Logging Settings

DNP IP Settings

The DNP Server connection has binary inputs/outputs and analog inputs corresponding to the modules and attached devices in each project. Every digital input/output and analog included in each project is mapped through the tag processor to corresponding binary inputs/outputs and analogs in the DNP settings. Modify the network and address settings to match your DNP Master. Refer to *DNP Server Configuration on page 47* for more information.

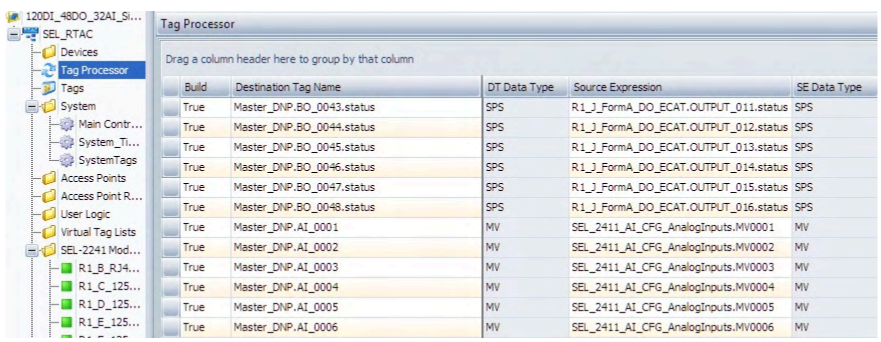


Figure 2.162 DNP Mapping With the Tag Processor

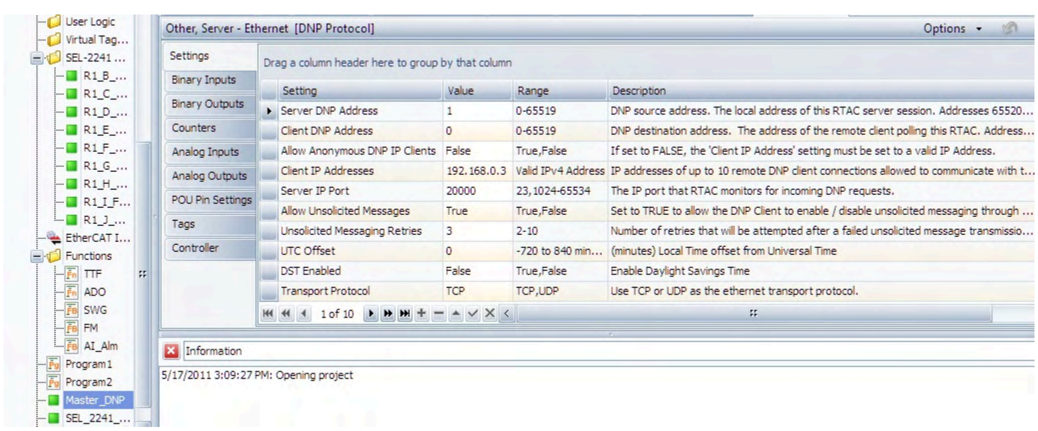


Figure 2.163 DNP Server Connection IP Settings

IEC 61131-3 Example Function Use

Each included example function includes comments describing functionality and variable descriptions. Function call examples for each Structured Text (ST) function are in Program 1. Function call examples for Continuous Function Chart (CFC) functions are in Program 2. Use these functions as necessary for I/O control and analog signal processing. Refer to *Section 9: Custom Logic* for detailed information about the IEC 61131-3 programming environment in ACSELERATOR RTAC.

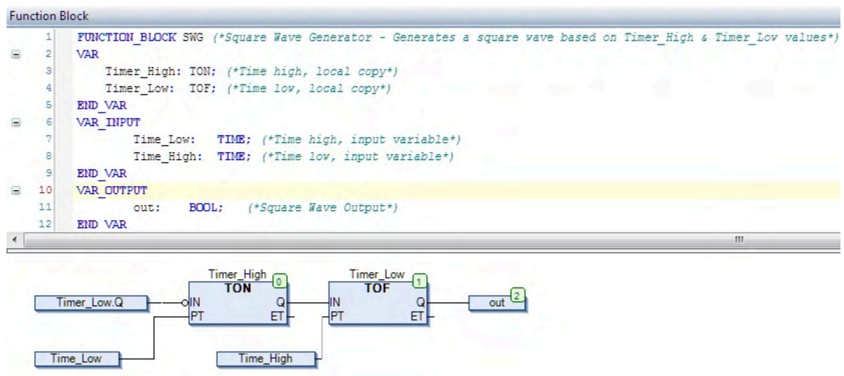


Figure 2.164 ST Function Example

IEC 61850 GOOSE Analog Settings

Four of the RTU projects include dc analog inputs from an SEL-2411 that uses GOOSE messages. The IEC 61850 configuration includes 32 dc analog inputs from an SEL-2411 with four installed 8 AI cards. To edit the network settings or the included points, use the export IEC 68150 configuration in the insert IEC 61850 dialog in ACSELERATOR RTAC. Open the export file with ACSELERATOR Architect to modify the IP or point settings for the SEL-2411 and SEL-2241. Once you have saved changes, use the insert IEC 61850 dialog in ACSELERATOR RTAC to insert the new configuration. Refer to *IEC 61850 on page 118* for more details.

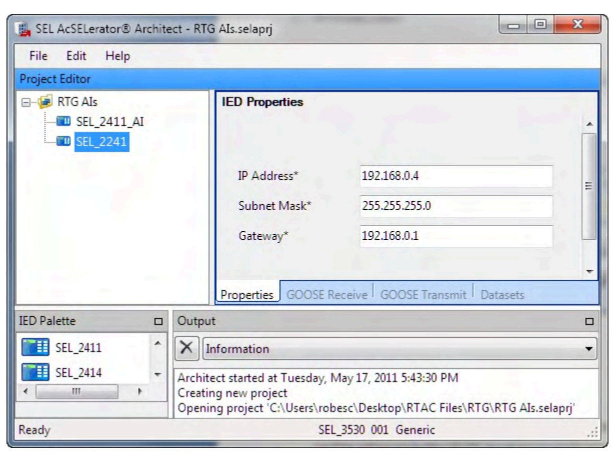


Figure 2.165 Architect GOOSE Configuration

Network Global Variable List (NGVL)

Use a Network Global Variable List (NGVL) to share a global variable list over an Ethernet network between a transmitting NGVL device and receiving NGVL devices, such as the SEL-1102 and the SEL-3530 RTAC. You can configure variables in an NGVL that are of standard data types (BOOL, REAL, etc.) and complex tag types (SPS, MV, CMV, etc.). The protocol supports transmitting variable lists from one device to another, to several, or to all devices on a given network. Data are transmitted on a configurable cyclical transmission interval, on change of tag value, or when triggered by a configurable Boolean event change of state. Configure other NGVL compatible devices on the network to receive the NGVL transmission by creating a matching tag list with matching tag list ID.

NGVL Configuration

You can configure multiple NGVL connections in the RTAC as either transmit (sends global list data) or receive (receives global list data). The global variables you create in an NGVL are available anywhere in the RTAC for data mapping or manipulation just as any other global variable. The NGVL protocol handles the transmission of the values contained in the global variables.

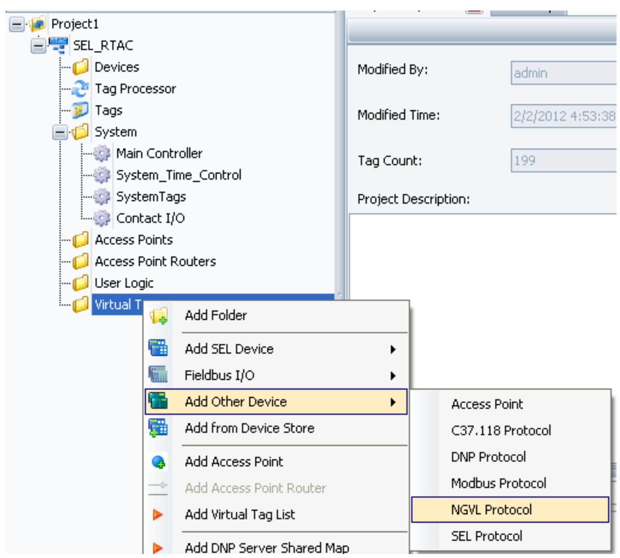


Figure 2.166 Insert NGVL

Insert an NGVL by selecting **Other** from the **Insert** ribbon or right-clicking anywhere in the device tree (see *Figure 2.166*). Give the device connection a unique name and note you cannot select the connection type because it is a peer-to-peer Ethernet protocol.

By selecting the **Settings** tab, you see all the configurable items for communications. Configure the NGVL to be transmitted or received. Check the **Description** column for details on each configuration item. Move the slider bar or hover your mouse over a truncated description to see the entire text of a description. Type any necessary comments in the blank column to the far right.

Select the **Variables** tab to add the global variables. The global variables can be of any type supported in the RTAC. You can use data structures or simple variable types. Configure one RTAC to transmit a list, including a unique list ID. Then, configure another RTAC to receive the list. The receive list configuration must specify the same unique list ID and must contain the same data list as the transmitted list. Although it may be confusing to do so, you can use different names in the transmit and receive lists, but the data types must be the same.

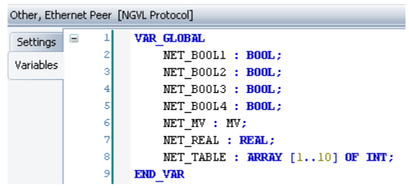


Figure 2.167 Add NGVL Tags

Example 2.5 Configure a Transmit and a Receive NGVL on Two RTACs

The following example demonstrates how to configure a transmit NGVL on one RTAC, then use that same list configuration on the receiving RTAC. Using the same NGVL configuration reduces the risk of user configuration error. In this example, RTAC 1 has an IP address of 192.168.1.2 on Eth 1 and RTAC 2 has an IP address of 192.168.1.5 on Eth 1.

Step 1. In a blank RTAC project, insert an NGVL into the device list by right-clicking on the Virtual Tag Lists folder (or anywhere in the device tree), then select **Add Other Device > NGVL Protocol**.

Step 2. Rename the **Device Name** to something meaningful and unique. In this example, we will call it **Transmit**.

Step 3. Change the Broadcast Address to 192.168.1.255. This designates all addresses on the network that start with 192.168.1 are to receive this message. Leave the other configurations under **Settings** at their default values.

Step 4. Click on Variables and configure several global variables as shown in the table below and *Figure 2.167*.

Variable Name	Variable Type
My_bool	BOOL
My_real	REAL
My_MV	MV

Step 5. Map values to the global variables from other tags in the RTAC system. *Figure 2.167* shows an example of mapping data from DNP client tags in the Tag Processor to these new global variables.

Step 6. Save the project.

Step 7. Right-click on the new NGVL, select **Save to Device Store**, and press **OK**.

Step 8. Load project into the first RTAC (referred to in this example as RTAC 1).

Step 9. Connect an Ethernet cable into Eth 1 from RTAC 1 and into Eth 1 of RTAC 2.

Step 10. Click on **Interfaces** on the RTAC 2 web interface to change the IP address of Eth 1 to 192.168.1.5.

Step 11. Create another RTAC project.

Step 12. Right-click on **Virtual Tag Lists** (or anywhere in the device tree), select **Add From Device Store**, and choose the same NGVL from *Step 7*.

Step 13. Change the GVL Type from **Transmit** to **Receive**.

Ensure the List ID is the same.

Step 14. Change the Broadcast Address to the IP address of RTAC 1, which is configured with the Transmit NGVL (192.168.1.2).

Step 15. Save the project and load it into the second RTAC (referred to as RTAC 2).

Data from the NGVL on RTAC 1 are broadcast on the network. RTAC 2 receives the NGVL data and places the data into the corresponding NGVL tags in RTAC 2.

Landis and Gyr LG 8979

In the mid 1980s, Landis and Gyr (LG) developed the Telegyr, or LG 8979 protocol, to provide byte-oriented half-duplex communications between the control center and remote terminal units. The RTAC supports a maximum of 10 LG 8979 client connections and 10 LG 8979 server connections simultaneously.

This section describes the configuration and use of LG 8979 protocol with ACSELERATOR RTAC. A list of supported function codes for client and server connections is included at the end of this section.

For a detailed description of client and server concepts and other general information on protocol configuration, see *Overview on page 25*.

LG 8979 Client

Configuration

Configure an LG 8979 client connection on any of the RTAC serial ports, or configure as a tunneled connection for use over Ethernet. The RTAC will poll for the configured data and store these data into global variables called tags. Use the Tag Processor to map these data to any other protocol, logs, user logic, etc.

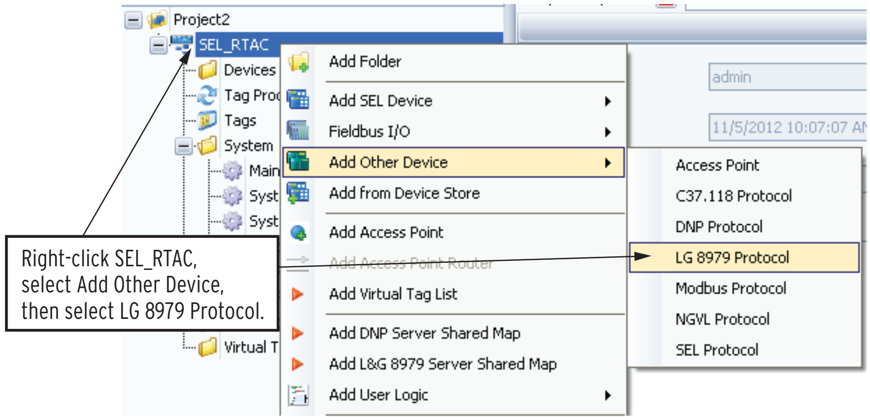


Figure 2.168 Insert LG 8979 Client

Give the device connection a unique name, and select the type of connection as shown in *Figure 2.1*.

Settings Tab

Upon your selecting the **Settings** tab, the RTAC displays all configurable items for communication. Check the Description column for details on each configuration item. Move the slider bar, or hover your mouse over a truncated description to see the entire text of a description. Type any necessary comments in the blank column to the far right.

Select the **Advanced Settings** check box to enable configuration of advanced settings.

Messages Tab

Define the type and frequency of messages the RTAC sends to the LG 8979 IED in the **Messages** tab. There are predefined messages by default, but you can adjust the default settings to alter the poll period, function, poll type, and range of points the messages use. See *Table 2.69*.

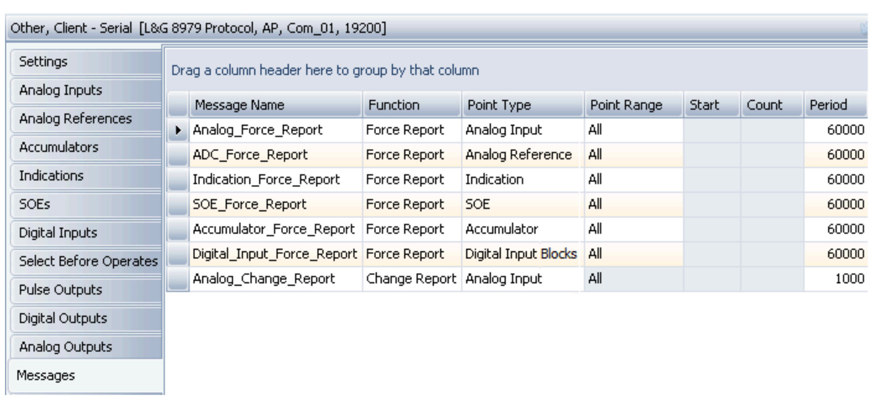


Figure 2.169 LG 8979 Messages

Table 2.69 LG 8979 Messages

Column Name	Description
Message Name	User-defined message name
Function	The function the message performs as either Force Report or Change Report
Point Type	The type of point requested for by the message
Point Range	The range of points polled for by the message
Start	The first point of a range of points
Count	The number of points within a range of points
Period	The period of time in milliseconds in which the RTAC will send the message

Add Client Device Data

Select device data settings tabs such as Analog Inputs, Analog References, Accumulators, etc., to add and configure LG 8979 tags.

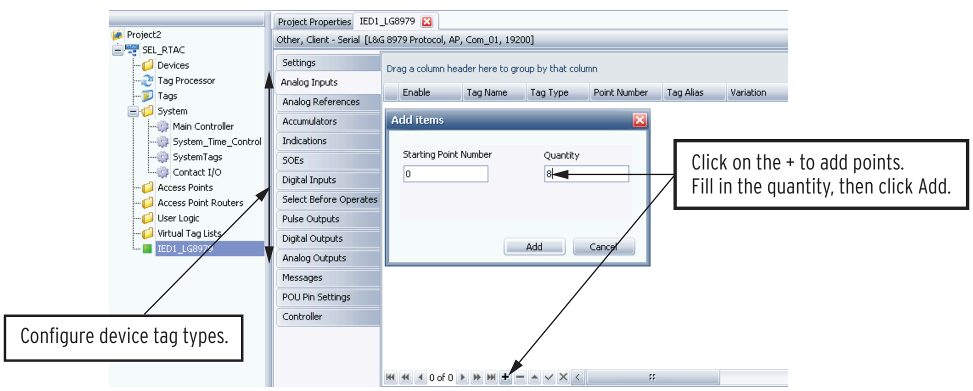


Figure 2.170 Add LG 8979 Client Tags

The following steps detail the addition of client device data.

Step 1. Click on a device tag type tab to add and configure tags.

Step 2. Click + to add tags (4096 tag limit per device tag type).

Creating only the number of necessary tags will help the system run at optimum performance.

Step 3. Change the names of the tags as necessary.

Step 4. Change other tag-related information as necessary.

Repeat these steps to configure all IED devices. When finished, configure one or more servers to deliver the data to remote client devices.

NOTE
Use Copy (**<Ctrl+C>**) and Paste (**<Ctrl+V>**) to populate columns of Tag names and to duplicate devices.

Device Tag Type Configuration Parameters

Data and poll types have parameters that you must configure correctly to ensure proper system operation. ACSELERATOR RTAC will gray out configuration fields that do not apply to a specific device.

Table 2.70 Common Device Tag Type Parameters

Parameter	Description	Default
Enable	Set this flag to True to enable processing of this tag. Set this flag to False to disable processing.	True
Tag Name	This parameter describes the tag. You can change this name as necessary.	The default tag name contains the device name and tag type and is numbered 0–*n* tags
Point Number	This is the point number for which the LG 8979 client will poll. If points are not contiguous in the IED, change these point numbers to match the LG 8979 database for the IED.	Contiguous from 0 to the point count
Tag Alias	Enter an optional descriptive tag name in this field, and then reference this tag alias anywhere in the RTAC system in place of the actual tag name.	
Comment	Optional user-entered comment field	

Table 2.71 Digital Input, Indications, and SOEs Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	SPS
Block Number	The digital input block number. Each block contains 16 binary inputs.	
Bit Number	The bit placement for this digital input within the block specified	Bit number 0 is the first digital input, bit 1 is the second, and so forth
Status Value	The initialized value at startup	False

Table 2.72 Digital and Pulse Output and Select Before Operate Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. In DNP, there are five binary output operations and one status.	Control operations is operSPC. Status is SPS
Block Number	The digital input block number. Each block contains 16 binary inputs.	
Bit Number	The bit placement for this digital input within the block specified	Bit number 0 is the first digital input, bit 1 is the second, and so forth
Status Value	The initialized state at startup	False
Control Model	Defines if the control is a single pulse or persistent (latch) type control	Defaults are set according to control type and cannot be changed

Parameter	Description	Default
Number of Pulses	Defines the number of pulses the control will issue each time this point is controlled	1
On Pulse Dur	Defines the duration for an active pulse	
Off Pulse Dur	Defines the duration for an inactive pulse	

Table 2.73 Accumulators Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	BCR
Actual Value	The initialized running value at startup	0
Frozen Value	The default frozen value the tag will have before the RTAC initializes the tag upon startup	

Table 2.74 Analog Input and Analog References Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	MV
Inst Magnitude	The initialized instantaneous value at startup	
Magnitude	The initialized dead-banded value at startup	
Dead Band	The number of units of change necessary to move the Inst Magnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag. An excursion of this dead band will generate a time-stamped DNP event.	
Zero Dead Band	The number of units at or below which Magnitude is forced to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	

Table 2.75 Analog Outputs

Parameter	Description	Default
Variation	The variation of the output type. Options are 12 bit signed or unsigned.	
Status Value	The initialized value at startup	

Parameter	Description	Default
Control Value	The initialized control value at startup	
Max Value	The maximum value allowed for this point. If InstMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	

LG 8979 Server Configuration

Configure an LG 8979 server connection to communicate via serial or Ethernet tunneled to LG 8979 polling clients. Create binary input, analog input and other tag types in an LG 8979 Server Shared Map as placeholders for data that the clients will poll. Configure the LG 8979 server to use the configured shared map. Each LG 8979 server can use a unique map or share the same map. You can create a maximum of 10 LG 8979 shared maps in a project. Map data to those shared map tags by using the Tag Processor from client IED or other source tags in the RTAC database.

Insert a unique LG 8979 Server connection for each LG 8979 client connected to the RTAC.

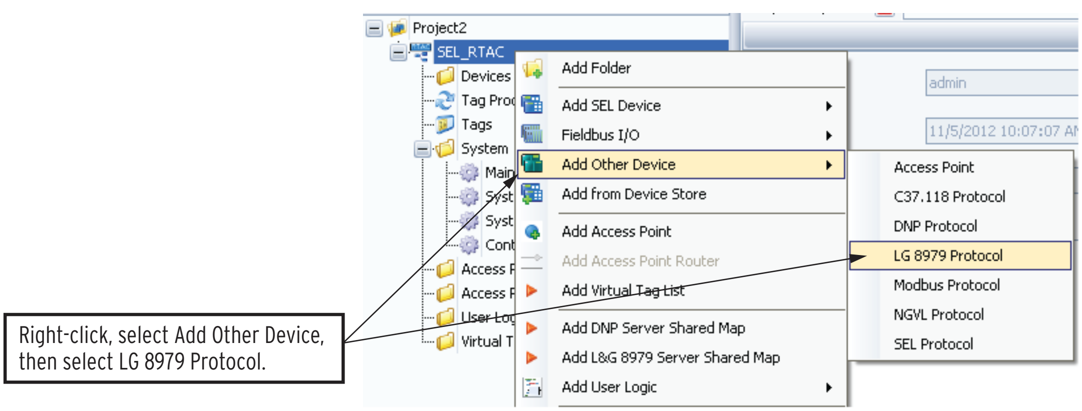


Figure 2.171 Insert LG 8979 Server Device

Give the device connection a unique name, and select the type of connection as shown in *Figure 2.3*. The **Settings** tab contains all the configurable items for communications. Note that you must enter an LG 8979 Server Shared Map name in the settings. This map contains all the tags that this LG 8979 server uses. Check the Description column for details on each configuration item. Move the slider or hover your cursor over the description to see the entire description text. Type any necessary comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Add LG 8979 Share Map

The following steps detail the addition of a server shared map.

Step 1. Under the **Insert** tab, select **Tag Lists**, then **LG 8979 Server Shared Map**.

This will insert a configurable tag list for an LG 8979 server. You can configure multiple LG 8979 servers to use this same shared tag list.

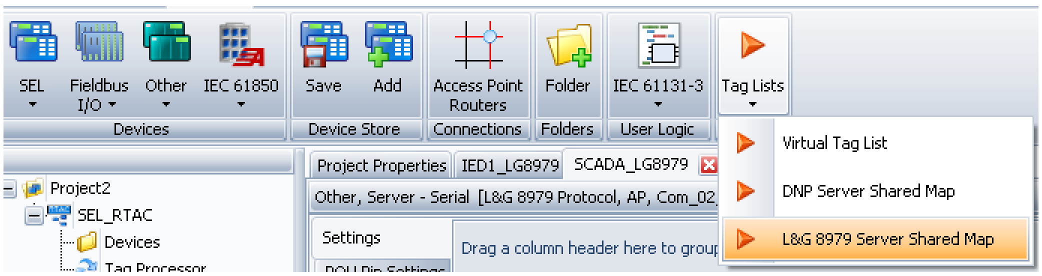


Figure 2.172 Add LG 8979 Server Map

Step 2. Click on a device tag type tab to add and configure tags.

Step 3. Click + to add tags (4,096 tag limit per device tag type).

Creating only the number of necessary tags optimizes system performance.

Step 4. Change the names of the tags as necessary.

Step 5. Change other tag-related information as necessary.

Repeat *Step 2* through *Step 5* to configure all server tags. When finished, configure the LG 8979 server to use this custom map by selecting the LG 8979 Map Name under settings. Although more than one LG 8979 server can use the same map, each LG 8979 server will manage separate event queues. Configure the Tag Processor to populate these server connection tags with actual values.

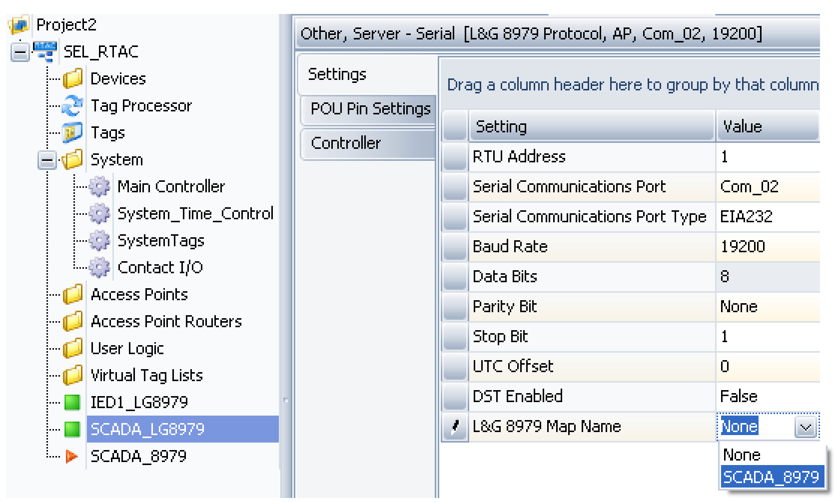


Figure 2.173 Select LG 8979 Shared Map

Device Tag Type Configuration Parameters

Each device tag type has parameters that must be configured correctly to ensure proper system operation. Configuration columns in the LG 8979 Shared Map have the same meaning as in LG 8979 clients (described previously) except for the addition of fields that have special meaning in LG 8979 servers. Use these fields to alter the default object variations returned to a polling LG 8979 client.

POU Pin Settings

See *POU Pin Settings (Advanced Usage) on page 37* for a description of the **POU Pin Settings** tab.

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

Table 2.76 LG 8979 Client Function Codes

Function Code	Description
0 Analog Change Report	Values that have exceeded an integrated delta value or values that have exceeded the configured dead band
1 Analog Force Report	The present value of a point or group of points
5 ADC Reference Force Report	The present value of ADC references
6 Indication Change Report	Indications that have changed state at least once since the last reported value
7 Indication Force Report	The present state and memory change of indications
8 SOE Change Report	SOE report containing the state and time of SOEs that have changed since the last reported value
9 SOE Force Report	All SOE entries
11 Digital Input Force Report	The present state of digital input points
12 Accumulator Change Report	Accumulator values that have been frozen and not reported in the last read
13 Accumulator Force Report	All frozen accumulator values
14 SOE Log Change Report	All SOE log entries that have changed since the last interrogation
20 Analog Output	Control an analog output value.
21 SBO Select	Select an SBO point to operate.
22 SBO Operate	Operate the previously select SBO point.
23 Digital Output	Immediate operate on a digital input point
24 Accumulator Freeze	Freeze one or more accumulator point values.

Function Code	Description
25 Pulse Output	Immediate control to raise or lower for a configured duration
26 Pulse Train Output	A single message that controls several points for a raise or lower for a configurable duration
30 Restart RTU	Perform a restart of the remote unit.
32 Time Synchronization	Time synchronize the remote unit.
33 Time Bias	Message sent from the polling master and returned by the client for the purpose of determining how long it takes for the message to travel on the wire. The time bias is added to the time synchronization to compensate for time lost in data transmission.
34 Analog Dead Bands	Set dead bands for each analog point in the remote unit.
37 Continuation Request	If the remote unit replies with the continuation bit set in the data message, the polling client sends a continuation request to the remote server. Upon receiving the continuation request, the remote server sends the next message.
38 Repeat Last Message	Upon receipt of the repeat last message function code, the server resends the last transmitted message.
39 Firmware Configuration	Upon receipt of the firmware configuration function code, the server replies with a number representing the firmware version.
63 Exception Report	If the number of exceptions (data changes) exceed the server's change buffer capacity, the server will generate an exception report.

In addition to the function codes listed in *Table 2.76*, the LG 8979 server supports the function codes shown in *Table 2.77*.

Table 2.77 LG 8979 Server Function Codes

Function Code	Description
2 Analog Group Change Report	Values that have exceeded an integrated delta value or values that have exceeded the configured dead band
3 Analog Group Force Report	The present value of a point or group of points
31 RTU Configuration	The server reports the point configuration

CP2179

CP2179 Client Configuration

Configure a CP2179 Client connection to communicate via serial or Ethernet tunneled serial to as many as 100 IEDs. The RTAC will poll data from these IEDs and store the data it receives in global variables. Use the Tag Processor to map these data to any protocol, logs, user logic, etc.

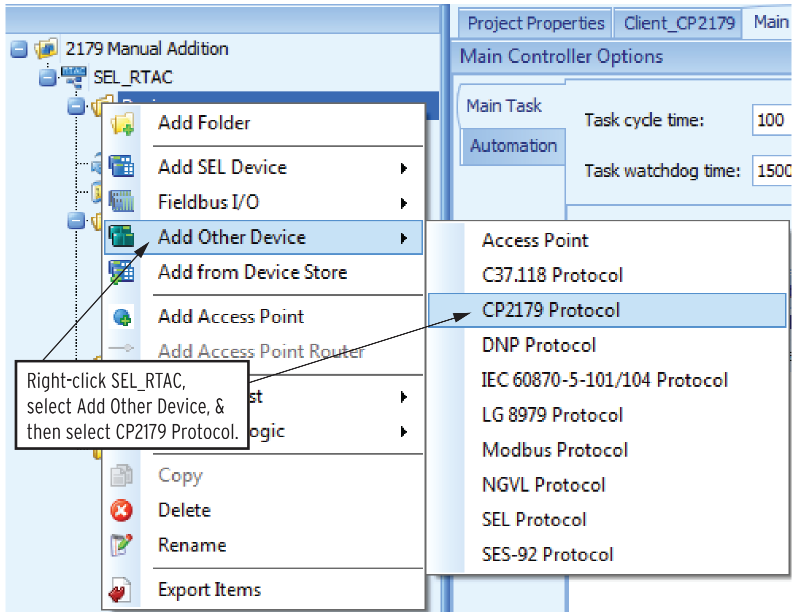


Figure 2.174 Insert CP2179 Client

Settings Tab

Upon your selecting the **Settings** tab, the RTAC displays all configurable items for communication. Check the **Description** column for details on each configuration item. Move the slider bar, or hover your mouse over a truncated description to see the entire text of a description. Type any necessary comments in the blank column to the far right.

Select the **Advanced Settings** check box to enable configuration of advanced settings.

Select device data settings tabs such as **2-bit Status**, **Analog Inputs**, etc., to add and configure CP2179 tags.

Messages

Set up the number and type of messages you need. Each message can apply to all sequence ID numbers or a configurable range.

Message Name:

Unique identifier for that message.

Period:

Frequency in millisecond a message will be triggered.

Function:

Function Code	Function Code Description	Supported Features
0x00	Basic Scan	2-bit Status Simple Status Analog Input (signed 16-bit format) Pulse Accumulator
0x01	Scan Inclusive	Signed 16-bit format
0x03	Special Calculations	32-bit IEEE Single-Precision Floating Point
0x20	Reset Accumulator	

Type:

Message type selection applying only to Basic Scan. Types are:

- 2-bit Status
- Simple Status
- Analog Input (signed 16-bit format)
- Pulse Accumulator

Range:

Sequence ID address selection. Option are: All or a range specified by a Starting Sequence ID and a count of Sequence IDs following referred to as Start/Count.

Start:

Sequence ID of the first point addressed by the message.

Count:

Number of consecutive sequence IDs requested by the message.

Add Client Device Data

The following steps detail adding client device data.

Step 1. Click a device tag type tab to add and configure tags.

Step 2. Click + to add tags (10,000 tag limit or 5000 for SEL-3505, per device tag type).

Creating only the number of necessary tags will help the system run at optimum performance.

Step 3. Change the names of the tags, if necessary.

Step 4. Change other tag-related information as necessary.

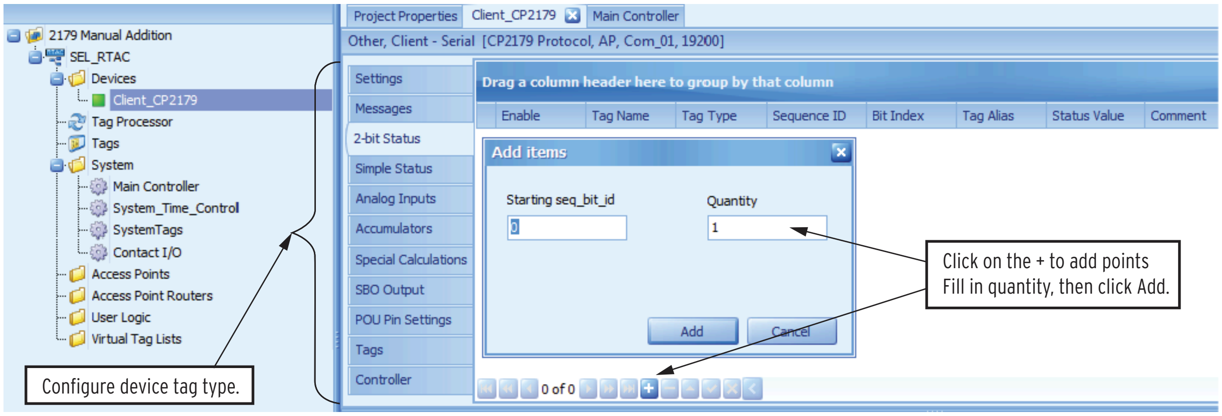


Figure 2.175 Add CP2179 Client Tags

Device Tag Type Configuration Parameters

Each device tag type has parameters that must be configured correctly to ensure proper system operation.

Table 2.78 Common Device Tag Type Parameters

Parameter	Description	Default
Enable	Set this flag to True to enable processing of this tag. Set this flag to False to disable processing.	True
Tag Name	This parameter describes the tag. You can change this name as necessary.	The default tag name contains the device name and tag type and is numbered 0–*n* tags.
Sequence ID	Point address on the server data map	Contiguous from 0 to the number of IDs
Tag Alias	Enter an optional descriptive tag name in this field, and then reference this tag alias anywhere in the RTAC system in place of the actual tag name.	
Comment	Optional user-entered comment field	

Table 2.79 2-Bit Status

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	SPS
Bit Index	Bit position within the Status Word, identified by the Sequence ID	Contiguous from 0 to 7
Status Value		

Table 2.80 Simple Status

Parameters	Description	Default
Tag Type	See *Data Types on page 747* for more details.	SPS
Bit Index	Bit position within the Status Word, identified by the Sequence ID	Contiguous from 0 to 7
Status Value		

Table 2.81 Analog Input

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	INS
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	

Table 2.82 Pulse Accumulator

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	BCR
Actual Value	The initialized running value at startup	0
Frozen Value	The default frozen value the tag will have before the RTAC initializes the tag upon startup	0

Table 2.83 Special Calculations

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	INS
Inst Magnitude	The initialized instantaneous value at startup	
Magnitude	The initialized dead-banded value at startup	
Dead Band	The number of units of change necessary to move the Inst Magnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag. An excursion of this dead band will generate a time-stamped DNP event.	
Zero Dead Band	The number of units at or below which Magnitude is forced to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	

Parameter	Description	Default
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	

Table 2.84 SBO Output

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details.	SPC
Status	SPS Status of the point identified by Sequence ID	
Control Model	Nonconfigurable field	Persist
On/Off Pulse Duration	Nonconfigurable field	

POU Pin Settings

See *POU Pin Settings (Advanced Usage) on page 37* for a description of the **POU Pin Settings** tab.

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

SES-92

SES-92 is a byte-oriented, asynchronous serial client/server-based (master/slave) protocol derived from the Landis & Gyr TELEGYR 8979 protocol. Designed to provide supervisory control and data acquisition (SCADA) between master stations and remote terminal units (RTUs), the SES-92 protocol supports features such as remote download of analog dead bands, report by exception, time synchronization, and counter freeze capability. The RTAC supports as many as 10 SES-92 server and 10 SES-92 client connections simultaneously.

This section describes the configuration and use of the SES-92 protocol with ACSELERATOR RTAC. The RTAC supports function codes listed in *Table 2.95*, as defined in the SES-92 Protocol Specification version 2.0, unless otherwise noted within this manual.

For a detailed description of client and server concepts and other general information on protocol configuration, see *Overview on page 25*.

SES-92 Client

Configuration

Configure an SES-92 client connection on any RTAC serial ports, or configure it as a tunneled connection for use over Ethernet. The RTAC will poll for the configured data and store these data into global variables called tags. Use the Tag Processor to map these data to any other protocol, logs, user logic, etc.

Give the device connection a unique name, and select the type of connection as shown in *Figure 2.176*.

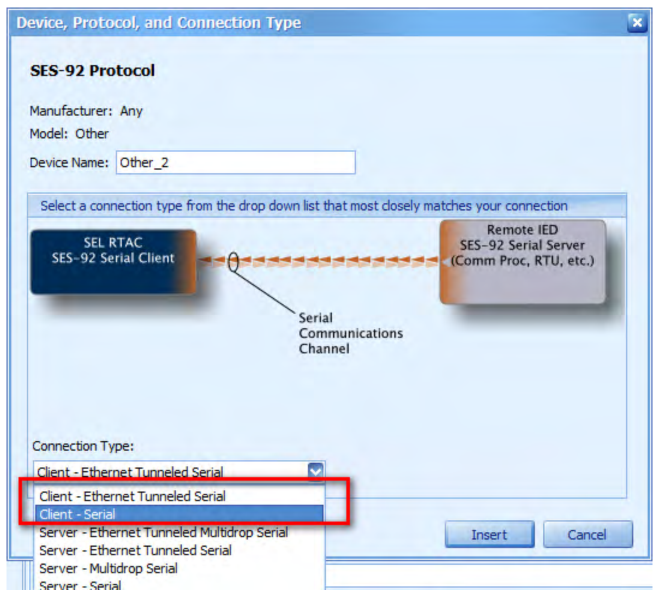


Figure 2.176 Adding SES-92 Client

Settings Tab

Upon selecting the **Settings** tab, the RTAC displays all configurable items for communication. Check the Description column for details on each configuration item. Move the slider bar or hover your mouse over a truncated description to see the entire text of a description. Type any necessary comments in the blank column to the far right.

Click the **Advanced Settings** check box to enable configuration of advanced settings.

Messages Tab

Define the type and frequency of messages the RTAC sends to the SES-92 IED in the **Messages** tab. There are predefined messages by default, but you can adjust the default settings to alter the poll period, function, poll type, and range of points the messages use.

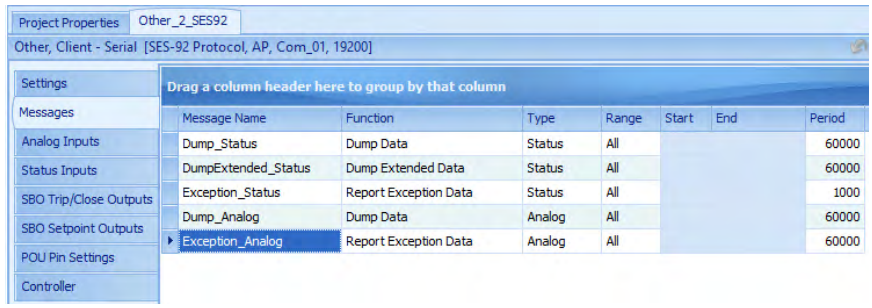
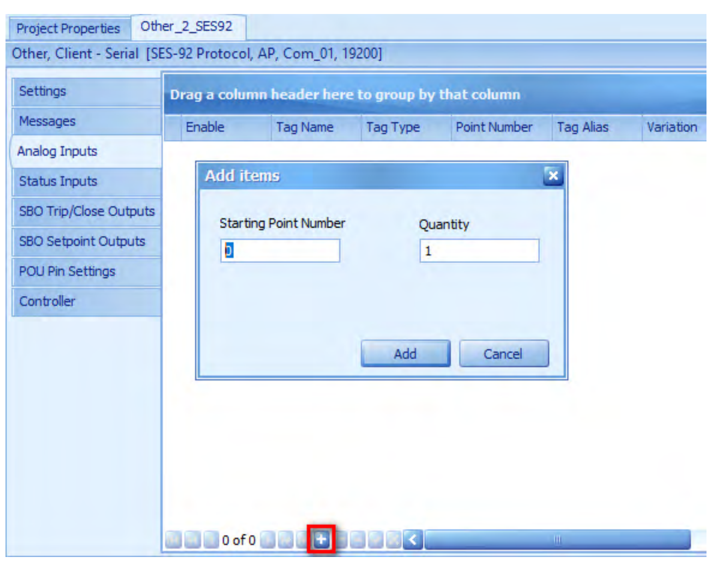


Table 2.85 SES-92 Messages

Column Name	Description
Message Name	User-defined message name
Function	Dump Data, Dump Extended Data, or Report Exception Data
Type	The type of data that will be polled (Status or Analog)
Range	The range of points that will be polled (All or Start/End)
Start	The first point that will be polled (0–4095)
End	The last point that will be polled (0–4095)
Period	The period of time (in milliseconds) in which the RTAC will send the message (0–100000000)

Add Client Device Data

Select device data settings tabs such as **Analog Inputs**, **Status Inputs**, **SBO Trip/Close Outputs**, or **SBO Setpoint Outputs** to add and configure SES-92 tags.



The following steps detail the addition of client device data.

Step 1. Click a device tag type tab to add and configure tags.

Step 2. Click the + button to add tags (as many as 4096 per device tag type). Creating only the number of necessary tags will help the system run at optimum performance.

Step 3. Change the names of the tags as necessary.

Step 4. Change other tag-related information as necessary.

Device Tag Type Configuration Parameters

Data and poll types have parameters that you must configure correctly to ensure proper system operation. ACSELERATOR RTAC will grayed out configuration fields that do not apply to a specific device.

Table 2.86 Device Tag Type Configuration Parameters

Parameter	Description	Default
Enable	Set this flag to True to enable processing of this tag. Set this flag to False to disable processing.	True
Tag Name	This parameter describes the tag. You can change this name as necessary.	The default tag name contains the device name and tag type and is numbered 0–*n* tags.
Point Number	This is the point number for which the SES-92 client will poll. If points are not contiguous in the IED, change these point numbers to match the SES-92 database for the IED.	Contiguous from 0 to the point count.
Tag Alias	Enter an optional descriptive tag name in this field, and then reference this tag alias anywhere in the RTAC system in place of the actual tag name.	
Status Value	This is the default value of the tag.	0 or False
Comment	Optional user-entered comment field	

POU Pin Settings and Controller

Table 2.87 POU Pin Settings and Controller

Pin Name	Pin Type	Description	Default
Disable_Controls	Input BOOL	While True, processing of any controllable tags is blocked.	FALSE
Disable_Tag_Updates	Input BOOL	The POU will not update or process changes to tags while this input is True.	FALSE

Pin Name	Pin Type	Description	Default
EN	Input BOOL	The EN input enables or disables this specific function block instance. Other inputs have no effect while EN is False.	TRUE
Issue_Dump_Diagnostics	Input BOOL	On the rising edge, issue a Dump Diagnostics message to the RTU.	FALSE
Issue_Dump_Diagnostics_Period	Input TIME (0, 100–10000000 ms)	The period at which to issue a Dump Diagnostics message to the RTU.	T#0MS
Issue_Poll_Exception_Data	Input BOOL	On the rising-edge, issue a Poll Exception Data message to the RTU.	FALSE
Issue_Poll_Exception_Data_Period	Input TIME	The period at which to issue a Poll Exception Data message to the RTU.	T#0MS
Issue_RTU_Configuration_-Complete	Input BOOL	On the rising-edge, issue a RTU Configuration Complete message to the RTU.	FALSE
Issue_RTU_Configuration_-Complete_Period	Input TIME (0, 100–100000000 ms)	The period at which to issue a RTU Configuration Complete message to the RTU.	T#0MS
Issue_Sync_Protocol_Version	Input BOOL	On the rising-edge, issue a Synchronize Protocol Version message to the RTU.	FALSE
Issue_Sync_Protocol_Version_-Period	Input TIME	The period at which to issue a Synchronize Protocol Version message to the RTU.	T#0MS
Issue_Time_Sync	Input BOOL	Requests time-synchronization message(s) to be sent on the rising edge. Time-synchronization processing is defined by the Propagation Delay Type setting.	FALSE
Issue_Time_Sync_Period	Input TIME (0, 100–100000000 ms)	Defines the interval used by the client to issue time-synchronization messages. Setting to zero causes the related function to be non-periodic.	T#600000MS
Reset_Statistics	Input BOOL	On the rising-edge trigger, all counter POU outputs are reset.	FALSE
Issue_<*Name*>[a]	Input BOOL	Request message to be sent on the rising edge.	
Issue_<*Name*>_Period[a]	Input TIME (0, 100–100000000 ms)	Defines the interval used by the client to issue the message. Setting to zero causes the related function to be non-periodic.	
ADC_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte ADC flag from the last Diagnostic Dump response.	FALSE
ALG_Flag_Status	Output BOOL	Contains the status of the Frame Header's Analog Exceptions flag from the last response received.	FALSE
Control_Overflow	Output BOOL	Asserted for a single processing interval when a Control operation could not be queued.	FALSE
Control_Overflow_Count	Output UDINT	Running sum of dropped Control operations.	0
Controls_Disabled	Output BOOL	Asserted when the Disable_Controls input is asserted. This indicates that the client will not issue control commands.	FALSE
Direct_Transparent_Connection	Output BOOL	Indicates that the communications have been interrupted by a direct transparent connection through an access point router.	FALSE

Pin Name	Pin Type	Description	Default
ENO	Output BOOL	Indicates that this specific function block instance is active if True. If False, the inputs have no effect and the outputs are not updated.	FALSE
ERR_Flag_Status	Output BOOL	Contains the status of the Frame Header's Error flag from the last response received.	FALSE
FW_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte F/W flag from the last Diagnostic Dump response.	FALSE
HW_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte H/W flag from the last Diagnostic Dump response.	FALSE
Invalid_Function_Block_Input	Output BOOL	True when a function block input currently has an invalid or out-of-range value.	FALSE
Issue_<*Name*>_DN[a]	Output BOOL	Asserts for a single processing interval once the message has completed.	
Issue_<*Name*>_Delay[a]	Output BOOL	Asserts when the message is triggered and the previous triggering of the message has not completed. Deasserts when on the rising edge of Issue_<Name>_DN.	
Issue_Dump_Diagnostics_Delay	Output BOOL	Asserts when the Dump Diagnostics message is triggered and the previous triggering of the Dump Diagnostics message has not completed. Deasserts when on the rising edge of Issue_Dump_Diagnostics_DN.	FALSE
Issue_Dump_Diagnostics_DN	Output BOOL	Asserts for a single processing interval once the message has completed.	FALSE
Issue_Poll_Exception_Data_Delay	Output BOOL	Asserts when the Poll Exception message is triggered and the previous triggering of the Poll Exception message has not completed. Deasserts when on the rising edge of Issue_Poll_Exception_Data_DN.	FALSE
Issue_Poll_Exception_Data_DN	Output BOOL	Asserts for a single processing interval when the message has completed.	FALSE
Issue_RTU_Configuration_-Complete_Delay	Output BOOL	Asserts when the RTU Configuration Complete message is triggered and the previous triggering of the RTU Configuration Complete message has not completed. Deasserts when on the rising edge of Issue_RTU_Configuration_Complete_DN.	FALSE
Issue_RTU_Configuration_-Complete_DN	Output BOOL	Asserts for a single processing interval when the message has completed.	FALSE
Issue_Sync_Protocol_Version_-Delay	Output BOOL	Asserts when the Sync Protocol Version message is triggered and the previous triggering of the Sync Protocol Version message has not completed. Deasserts when on the rising edge of Issue_Sync_Protocol_Version_DN.	FALSE
Issue_Sync_Protocol_Version_DN	Output BOOL	Asserts for a single processing interval when the message has completed.	FALSE

Pin Name	Pin Type	Description	Default
Issue_Time_Sync_Delay	Output BOOL	Asserts when the time-synchronization message is triggered and the previous triggering of the time-synchronization message has not completed. Deasserts when on the rising edge of Issue_Time_Sync_DN.	FALSE
Issue_Time_Sync_DN	Output BOOL	Asserts for a single processing interval when all time-synchronization processing has successfully completed.	FALSE
LCD_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte LCD flag from the last Diagnostic Dump response.	FALSE
LOP_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte LOP flag from the last Diagnostic Dump response.	FALSE
Message_Failure	Output BOOL	Asserts for a single processor cycle when a message is processed that does not conform to the expected protocol format. This may be because of an incomplete response or corrupted data in the message response.	FALSE
Message_Failure_Count	Output UDINT	A running sum indicating the number of messages that have successfully been sent or received.	0
Message_Received_Count	Output UDINT	A running sum indicating the number of messages received from the remote device.	0
Message_Sent_Count	Output UDINT	A running sum indicating the number of messages transmitted to the remote device.	0
Message_Success_Count	Output UDINT		0
Offline	Output BOOL	This output is False when protocol communications are in process. **Note:** Protocol communications may be in process, but no successful data are exchanged be- cause of settings configuration issues.	TRUE
PWR_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte PWR flag from the last Diagnostic Dump response.	FALSE
RST_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte RST flag from the last Diagnostic Dump response.	FALSE
STS_Flag_Status	Output BOOL	Contains the status of the Frame Header's Status Exceptions flag from the last response received.	FALSE
SYN_Flag_Status	Output BOOL	Contains the status of the Diagnostic Flag's Status Byte SYN flag from the last Diagnostic Dump response.	FALSE

[a] <Name> = Each configured message.

Supported Function Codes

The SES-92 client shall support the following function codes:

- Special Functions
  - Poll Exception Data (x00)
  - Repeat Last Message (x01)
  - Continue Message (x02)
  - Synchronize Protocol Version (x03)
- Digital Input Functions
  - Dump Status Data (x10)
  - Report Status Exception Data (x11)
  - Dump Extended Status Data (x12)
- Analog Input Functions
  - Dump Analog Data (x20)
  - Report Analog Exception Data (x21)
- Time-Synchronization Functions
  - Synchronize Time (x52)
- Control Output Functions
  - SBO Trip/Close Select (x60)
  - SBO Trip/Close Execute (x61)
  - SBO Setpoint Select (x62)
  - SBO Setpoint Execute (x63)
  - DO Setpoint Execute (x64)
- RTU Functions
  - Dump Diagnostics Request (x73)
  - RTU Configuration Complete (x79)

SES-92 Server

Configuration

Configure an SES-92 server connection to communicate via serial or Ethernet tunneled to SES-92 polling clients. Create binary input, analog input, and other tag types in an SES-92 server shared map as placeholders for data that the clients will poll. Configure the SES-92 server to use the configured shared map. Each SES-92 server can use a unique map or share the same map with other configured SES-92 server connections. You can create a maximum of 10 SES-92 shared maps in a project. Map data to those shared map tags by using the Tag Processor from client IED or other source tags in the RTAC database.

Insert a unique SES-92 server connection for each SES-92 client connected to the RTAC.

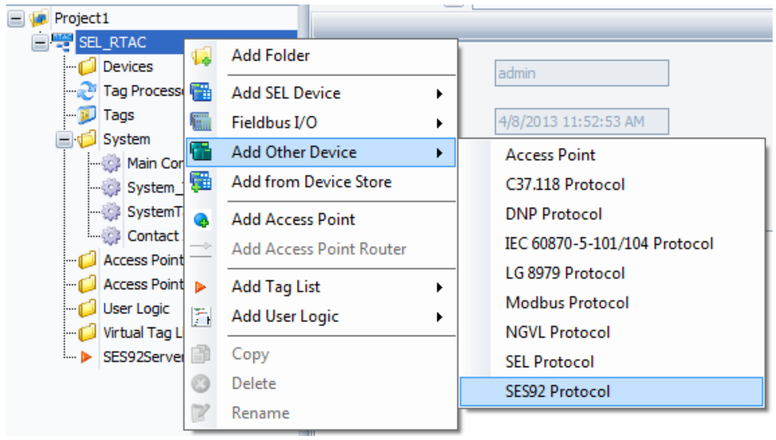


Figure 2.177 Insert SES-92 Server

Give the device connection a unique name, and select the type of connection as shown in *Figure 2.3*. The **Settings** tab contains all the configurable items for communications. Note that you must enter an SES-92 server shared map name in the settings. This map contains all the tags that this SES-92 server uses. Check the **Description** column for details on each configuration item. Move the slider or hover your cursor over the description to see the entire description text. Type any necessary comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Add SES-92 Share Map

Step 1. Under the **Insert** tab, select **Tag Lists > SES-92 Server Shared Map**.

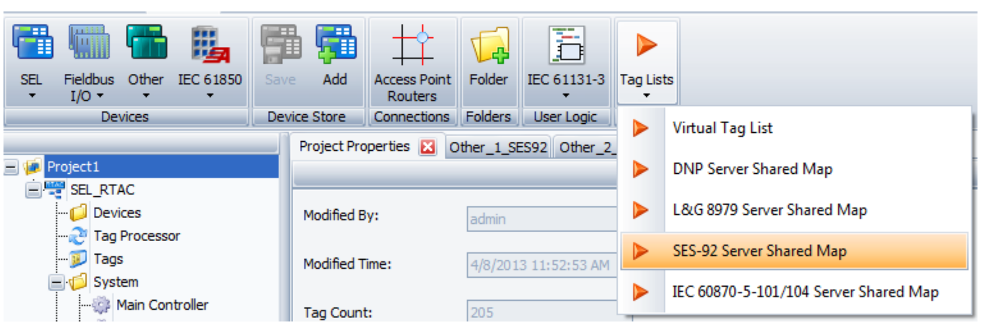


Figure 2.178 Add SES-92 Server Map

Step 2. Click on a device tag type tab to add and configure tags.

Step 3. Click + to add tags (4,096 tag limit per device tag type). Creating only the number of necessary tags optimizes system performance.

Step 4. Change the names of the tags as necessary.

Step 5. Change other tag-related information as necessary.

Repeat *Step 2* through *Step 5* to configure all server tags. When you are finished, configure the SES-92 server to use this custom map by selecting SES-92 Map Name in the **Setting** column. Although more than one SES-92 server can use the same map, each SES-92 server will manage separate event queues. Configure the Tag Processor to populate these server connection tags with actual values.

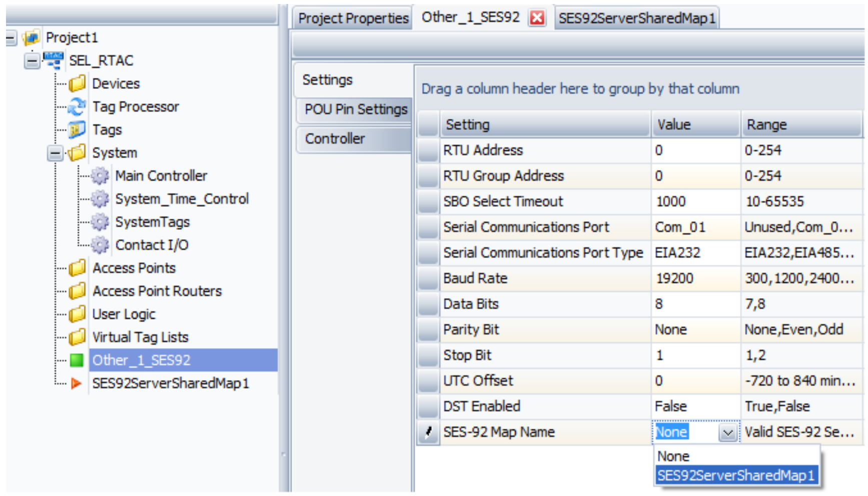


Figure 2.179 Select SES-92 Shared Map

Device Tag Type Configuration Parameters

Each device tag type has parameters that must be configured correctly to ensure proper system operation. *Table 2.88–Table 2.93* describe the columns used to configure the SES-92 Server tags. ACSELERATOR RTAC will gray out configuration fields that do not apply to a specific device.

Table 2.88 Common Device Tag Type Parameters

Parameter	Description	Default
Enable	Set this flag to TRUE to enable processing of this tag. Set this flag to FALSE to disable processing.	True
Tag Name	This parameter describes the tag. You can change this name as necessary.	
Point Number	This is the point number for which the SES-92 client will poll. If points are not contiguous in the IED, change these point numbers to match the SES-92 database for the IED.	Contiguous from 0 to the point count
Tag Alias	Enter an optional descriptive tag name in this field, and then reference this tag alias anywhere in the RTAC system in place of the actual tag name.	
Comment	Optional user-entered comment field	

Table 2.89 Status Input Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 4096 status input tags.	SPS
Status Value	The initialized value at startup	False
Comment	Optional user-entered comment field	

Table 2.90 SBO Trip/Close Output Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 256 SBO tags.	SPC
Control Model	Nonconfigurable field	Persist
On/Off Pulse Duration	Nonconfigurable field	

Table 2.91 Accumulator and Solid State Meter Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 256 accumulator and 256 solid state meter tags.	BCR
Actual Value	The initialized running value at startup	0
Frozen Value	The initialized value at startup	0

Table 2.92 Analog Input Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 4096 analog input tags.	INS
Inst Magnitude	The initialized instantaneous value at startup	
Magnitude	The initialized dead-banded value at startup	
Dead Band Count	The number of units of change necessary to move the Inst Magnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag.	
Zero Dead Band	The number of units at or below which Magnitude is forced to zero. If |mag| < zeroDB then mag := 0.	
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	

Table 2.93 SBO Setpoint Output Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 256 setpoint output tags.	INC
Variation	If the oper control word is 12-bit signed or 12-bit unsigned	
Set Magnitude	The initialized value at startup	0

POU Pins

Use POU pin settings to view the present state of the SES-92 server operation and to modify some of the behavior of the SES-92 protocol. Setting the **Visible** field to True will cause the POU pin to appear in the **Controller** tab. See *Table 2.94* for the settings descriptions.

Table 2.94 SES-92 POU Pin Settings

Pin Name	Pin Type	Description	Default
Communications_Offline_Timer	Input TIME	The time to wait for a poll from the client before determining the communications are offline. Configure in the **Settings** tab.	T#10000MS
Disable_Controls	Input BOOL	Disables controls through the protocol.	False
Disable_Tag_Updates	Input BOOL	Disables updating of tag values.	False
EN	Input BOOL	Enables this instance of SES-92.	True
PWR_Flag	Input BOOL	If this input pin is set to TRUE, the PWR flag is set to 1 in RTU Diagnostic data transmitted to the client.	False
Reset_Statistics	Input BOOL	Resets all communications statistics.	False
SYN_Flag	Input BOOL	If this input pin is set to TRUE, the SYN flag is set to 1 in RTU Diagnostic data transmitted to the client.	False
ACC	Output BOOL	Set to TRUE when unacknowledged frozen Accumulator and/or Solid State Meter data exists.	False
ALG	Output BOOL	Set to TRUE when unacknowledged Analog Input exception data exists.	False
Controls_Disabled	Output BOOL	Indicates controls have been disabled.	False
Direct_Transparent_Connection	Output BOOL	Indicates an access point router has disabled protocol communications and is using this port for direct transparent communications.	False
ENO	Output BOOL	Indicates this instance of SES-92 is running.	False
ERR	Output BOOL	Set to TRUE whenever the SES-92 Server ERR flag is transmitted in the message frame headers.	False
Freeze	Output BOOL	Asserts for one processing interval when a Freeze command is received from the client.	False
Message_Failure	Output BOOL	Indicates the last message failed.	False
Message_Failure_Count	Output UDINT	The number of times Message_Failure has been True.	0
Message_Received_Count	Output UDINT	The number of messages received.	0
Message_Sent_Count	Output UDINT	The number of messages sent.	0
Message_Success_Count	Output UDINT	The number of messages successfully sent to the client.	0
Offline	Output BOOL	Set to TRUE when Message_Received_Count has not been incremented for the number of ms defined in the Communication_Offline_Timeout setting.	True
SOE	Output BOOL	Set to TRUE when unacknowledged digital status change events exist in the SES-92 Server SOE event collection.	False
SOE_Overflow	Output BOOL	A Boolean flag that is asserted for a single processing interval if one or more digital status change events are discarded as a result of the SES-92 Server event collection being full.	False

Pin Name	Pin Type	Description	Default
SOE_Overflow_Count	Output UDINT	The number of times that the SOE_Overflow pin has been asserted.	0
STS	Output BOOL	Set to TRUE when unacknowledged Status Input exception data exists.	False

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

Table 2.95 SES-92 Function Codes

Function Code	Description	Notes
0h	Poll Exception	Response to FC 0h is the same as if the following sequence of commands has been received: 1. Report Status Exception Data 2. Report Analog Exception Data 3. Dump Accumulator Data 4. Dump Solid-State Meter Data
1h	Repeat Last Message	
2h	Continue Message	
3h	Synchronize Protocol Version	
10h	Dump Status Data	MCD reset if point quality changes.
11h	Report Status Exception Data	MCD reset if point quality changes. Events are maintained in a 10,000-entry circular buffer.
12h	Dump Extended Status Data	FC 12h is always the response to FC 12h request and is also the response to a Dump Status Data if any points have validity not = good. Dump and Extended Status requested stop point ID is increased in the response to provide 8-bit atomicity in the response message.
20h	Dump Analog Data	All requested points are always returned. Invalid or nonexistent values reported as –2048.
21h	Report Analog Exception Data	
24h	Assign Dead Band and Group	Configured dead band values are used until updated through the protocol by using FC 24h.
30h	Freeze Accumulator	
31h	Dump Accumulator Data	Reports value of 0 for tags with validity not = good.
33h	Dump Solid State Meter Data	Reports value of 0 for tags with validity not = good.
50h	Dump Digital Status Change Data	
52h	Synchronize Time	Response to FC 52h indicates a success, but the command is ignored by the RTAC.
53h	Read Time	

Function Code	Description	Notes
54h	Set Time Bias	
60h	SBO Trip/Close Select	Pulse on duration is the product of 10 ms and the control length from the **SELECT** command.
61h	SBO Trip/Close Execute	Execution time of control is one processing interval.
62h	SBO Setpoint Select	
63h	SBO Setpoint Execute	Set point tag trigger is asserted for one processing interval.
64h	DO (Direct Operate) Setpoint Execute	
70h	Reset RTU	EN POU pin is toggled False then back to True. Any OPB settings and integrity scan count, if included, are ignored.
73h	Dump Diagnostic Request	Subfunction diagnostic information is not returned.
77h	Suppress Data	
78h	Dump RTU Operating Parameters	The PRI, FRZ, COV, and RAE flags are always clear. Max message length and Message Integrity Count are set at zero. This establishes max message length to 261 bytes and disables integrity scan functionality.
79h	RTU Configuration Complete	

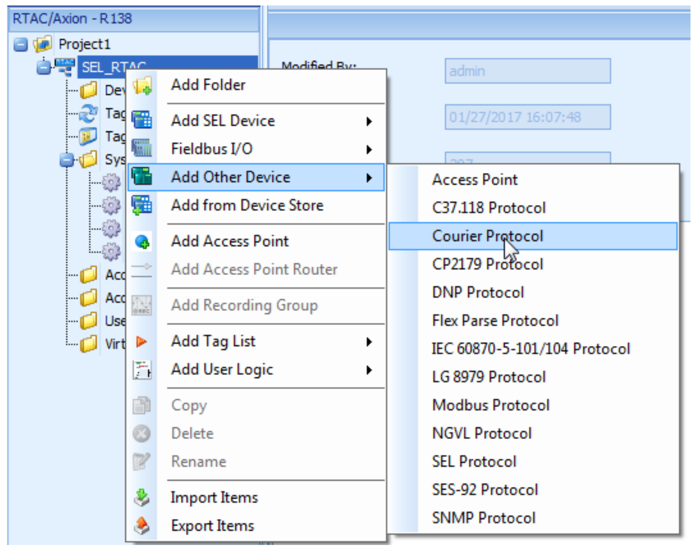
Alstom Event Collection (Courier Protocol)

Courier Client

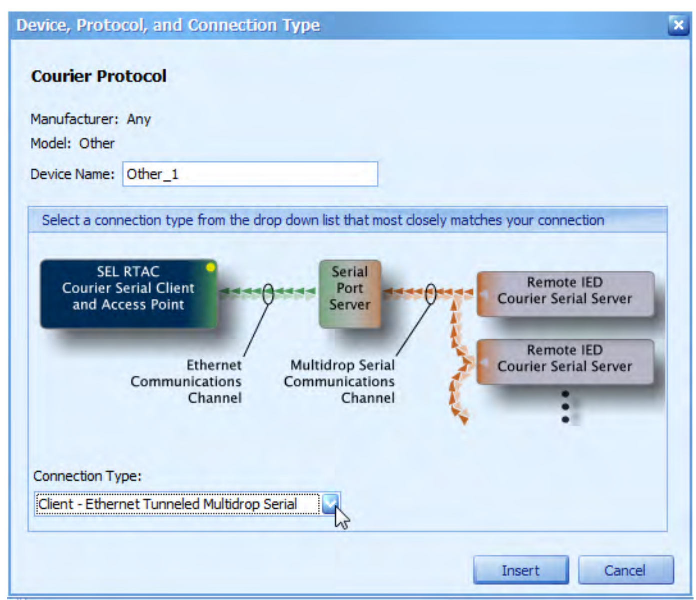
The Courier client is used to retrieve uncompressed Disturbance Records from Alstom relays and store them as compressed COMTRADE files. As many as 100 Courier clients are allowed. The client supports single-level addressing, and address allocation is not automatic. The client does not support collecting SCADA information.

Configuration

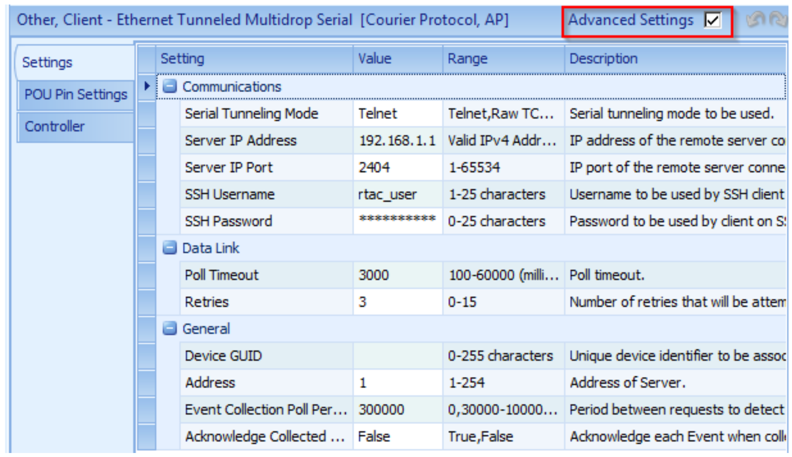
To use the Courier client, right-click in the project tree and select **Courier Protocol**.



Select the connection type (Ethernet Tunneled Multidrop Serial, Ethernet Tunneled Serial, Multidrop Serial, or Serial).



Then you can begin to configure the client. In the **Settings** tab, you can configure your connection, as well as event polling parameters, and whether or not you wish to acknowledge collected events. In the **Advanced Settings**, you can also specify a **Device GUID** to integrate with TEAM event collection, additional serial parameters, and SSH parameters.



Once you have your client configured and go online with your project, the RTAC will begin to poll the server for Disturbance Records.

POU Pins

Use POU pin settings to view the present state of the Courier client and to modify some of the behavior of the Courier protocol. Setting the **Visible** field to **True** will cause the POU pin to appear in the **Controller** tab. See *Table 2.96* for the settings descriptions.

Table 2.96 Courier POU Pin Settings

Pin Name	Pin Type	Description	Default
Direct_Transparent_Connection	Output BOOL	Indicates that the communications have been interrupted by a direct transparent connection through an access point router.	FALSE
EN	Input BOOL	Enables this instance of Courier client.	TRUE
ENO	Output BOOL	Indicates this instance of Courier client is running.	FALSE
Event_Collection_Count	Output UDINT	Number of Events collected since the POU was enabled.	0
Event_Collection_Delay	Output BOOL	Asserts when an event collection is triggered and the previous triggering of an event collection has not completed. Deasserts on the falling edge of Event_Collection_Pending.	FALSE
Event_Collection_Pending	Output BOOL	Asserted while there are uncollected Events. Deasserts once all Events have been collected.	FALSE
Event_Collection_Poll	Input BOOL	Triggers polls for new Events on the rising edge.	FALSE
Event_Collection_Poll_Period	Input TIME	Period between requests to detect Events.	T#300000MS
Event_Collection_Stored	Output BOOL	Asserts for a processing interval when an Event has been stored in nonvolatile memory.	FALSE
Invalid_Function_Block_Input	Output BOOL	True when a function block input currently has an invalid or out-of-range value.	FALSE

Pin Name	Pin Type	Description	Default
Message_Failure	Output BOOL	Asserts for a single processing interval following a message failure.	FALSE
Message_Failure_Count	Output UDINT	Increments value by one each time Message_Failure is asserted.	0
Message_Received_Count	Output UDINT	A counter of the number of messages received.	0
Message_Sent_Count	Output UDINT	A counter of the number of messages sent.	0
Message_Success_Count	Output UDINT	A running sum of the messages successfully sent or received.	0
Offline	Output BOOL	Set to True if data link activity time-out expires or a frame has not been received for a period of time, defined by Communications_Offline_Timer.	TRUE
Reset_Statistics	Input BOOL	Resets all communications statistics.	FALSE

IEC 60870-5-101 and -104

IEC 60870-5 is an IEC standard that defines systems used for supervisory control and data acquisition (SCADA), including details related to communications between devices. As part of the standard, IEC 60870-5-101 defines a byte-oriented, asynchronous serial-based protocol for communications associated with electrical power systems. The design is based on a balanced (where either client or server can transmit asynchronously) or unbalanced (server can only transmit when solicited by client) topologies and is suitable for point-to-point or point-to-multipoint configurations. IEC 60870-5-104 protocol extends the IEC 60870-5-101 protocol to fully integrate communications over Ethernet networks. At the application layer, the protocols are essentially the same, so selecting between the IEC 60870-5-101 or IEC 60870-5-104 protocol in ACSELERATOR RTAC is done by selecting serial or Ethernet connection methods. Although not used to a great extent in the U.S.A., the IEC 60870-5-101/104 protocols have historically been the *de facto* standard for communicating to devices connected with electrical power generation, transmission, and distribution worldwide.

This section describes the configuration and use of the IEC 60870-5-101/104 server with ACSELERATOR RTAC. You can configure as many as 50 concurrent IEC 60870-5-101/104 servers or IEC 60870-5-101/104 clients in a single RTAC project. The IEC 60870-5-101/104 server supports receiving commands (in the control direction) and transmitting responses (in the monitoring direction) for Application Service Data Unit (ASDU) types and Cause of Transmission (COT) as shown in the interoperability statements for IEC 60870-5-101 and -104, which list COTs for expected behavior only. Additional COTs are recognized for errors.

IEC 60870-5-101/104 Client Configuration

Configure a 101/104 client connection to communicate via serial or Ethernet to IEDs. The RTAC will poll data from these IEDs and store the data it receives in tags. Use the tag processor to map these data to any protocol, logs, user logic, etc.

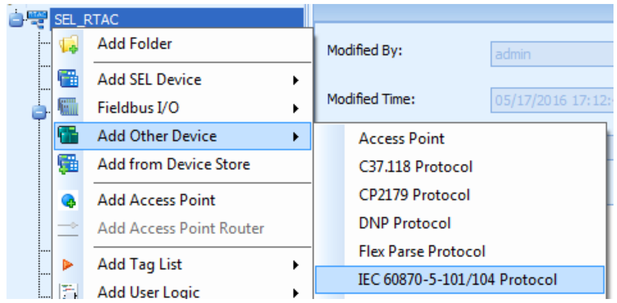


Figure 2.180 Adding IEC 60870-5-101/104 Client

Give the device connection a unique name and select the connection type, as shown in *Figure 2.180*. Refer to *Client Connection Types on page 31* for a description of each connection type.

Settings Tab

The **Settings** tab contains all configurable items for communication. Check the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description. Type any applicable comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Custom Messages Tab

The **Custom Messages** tab contains the poll requests the IEC 608070-5-101/104 client will use to collect data from the server. To add additional messages, use the + button. As many as 50 messages are supported. The supported message types and message qualifiers correspond with the interoperability document for the 101/104 client.

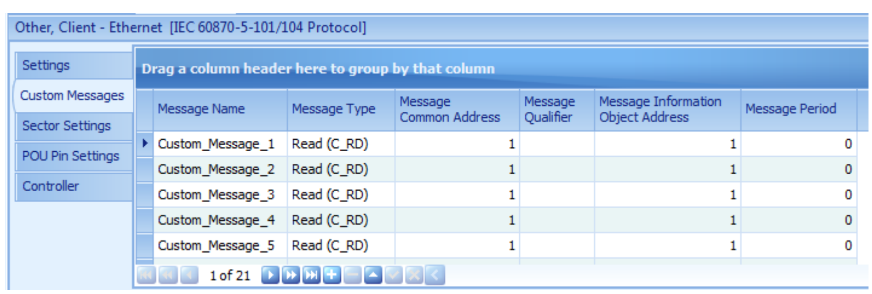


Figure 2.181 IEC 60870-5-101/104 Client Custom Messages

Table 2.97 Explanation of Column Headers

Column Name	Description
Message Name	User-defined name for the message
Message Type	The type of message the client will send
Message Common Address	The common address included in the message
Message Qualifier	A group selected for interrogation. Not all message types will require a qualifier.

Column Name	Description
Message Information Object Addresses	Address of the point the client will request
Message Period	The period during which the client will issue this message. This quantity is in milliseconds.

Sector Settings Tab

The **Sector Settings** tab contains configuration parameters for a maximum of eight sectors in a server. Each sector has a configurable name and common address. By default, each client has one sector. To add additional sectors, use the + button.

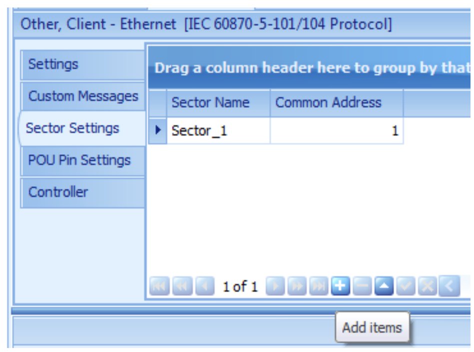


Figure 2.182 IEC 60870-5-101/104 Sector Settings

IEC 60870-5-101/104 Client Sector Map

To configure tags for the IEC 60870-5-101/104 client, select an existing **101/104 Client Sector Map**. Repeat *Step 1* through *Step 4* to configure each client tag.

Step 1. Click the tab for the appropriate tag type to add and configure tags.

Step 2. Click + to add tags. Create only the necessary number of tags to optimize system performance.

Step 3. Rename the tags as necessary.

Step 4. Configure other tag parameters as necessary.

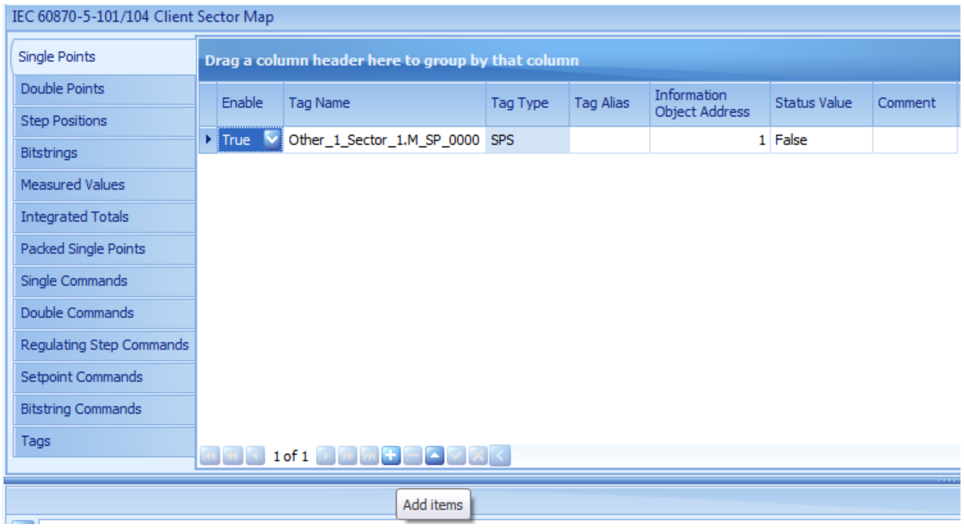


Figure 2.183 IEC 60870-5-101/104 Client Sector Map

POU Pins

Use POU pin settings to view the present state of the IEC 60870-5-101/104 client operation and to modify some of the behavior of the protocol. Setting the **Visible** field to True will cause the POU pin to appear in the **Controller** tab. See *Table 2.98* for the settings descriptions.

Table 2.98 IEC 60870-5-101/104 Client POU Pin Settings

Pin Name	Pin Type	Description	Default
EN	Input BOOL	The EN input enables or disables this specific function block instance. Other inputs have no effect while EN is False.	True
Disable_Tag_Updates	Input BOOL	The POU will not update or process changes to tags while this input is True.	False
Disable_Controls	Input BOOL	While True, processing of any controllable tags is blocked.	False
Reset_Statistics	Input BOOL	On the rising-edge trigger, all counter POU outputs are reset.	False
Issue_Time_Sync	Input BOOL	Requests time sync message(s) to be sent on the rising edge. Time-synchronization processing is defined by the Propagation Delay Type setting.	False
Issue_Time_Sync_Period	Input Time (0, 1–1440 minutes)	Defines the interval used by the client to issue time-synchronization messages. Setting it to zero causes the related function to be non-periodic.	Reference setting—Issue time-synch period
Issue_<Name>[a]	Input BOOL	Request custom message to be sent on the rising edge.	False
Issue_<Name>_Period[a]	Input Time (0, 250–100000000 milliseconds)	Defines the interval used by the client to issue the custom message. Setting to zero causes the related function to be non-periodic.	Reference custom message—Message_Period
Redundant_Change	Input BOOL	On a rising edge, this forces the client to change the active TCP connection if Aux Server IP Address is defined. Note: Only applicable on 104 connections/client (Ethernet)	False

Pin Name	Pin Type	Description	Default
ENO	Output BOOL	Indicates that this specific function block instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False
Offline	Output BOOL	This output is False when protocol communications are in process. Note: Protocol communications may be in process, but no successful data are exchanged because of settings configuration issues.	False
Slow_Poll_Mode_Enabled	Output BOOL	Asserts to indicate that the device has slowed polling by the slow poll mode multiplier.	False
Invalid_Function_Block_Input	Output BOOL	True when a function block input currently has an invalid or out-of-range value.	False
Controls_Disabled	Output BOOL	Asserted when the Disable_Controls input is asserted. This indicates that the client will not issue control commands.	False
Message_Sent_Count	Output UDINT	Running sum indicating the number of messages transmitted to the remote device.	False
Message_Received_Count	Output UDINT	Running sum indicating the number of messages received from the remote device.	0
Message_Failure	Output: BOOL	Asserts for a single processor cycle when a message is processed that does not conform to the expected protocol format. This may be because of an incomplete response or corrupted data in the message response.	False
Message_Failure_Count	Output UDINT	Running sum indicating the number of messages that have successfully been sent or received.	0
Direct_Transparent_Connection	Output BOOL	Indicates that the communications have been interrupted by a direct transparent connection through an access point router.	False
Data_Link_Timeout_Count	Output UDINT	Number of data link failures.	0
Response_Timeout_Count	Output UDINT	Number of application layer message response timeouts.	0
Reinit_In_Progress	Output: BOOL	While asserted, indicates that the client is performing initial communications queries prior to data processing	False
Issue_Time_Sync_DN	Output BOOL	Asserts for a single processing interval once all time-synchronization processing has successfully completed.	False
Issue_Time_Sync_Delay	Output BOOL	Asserts when the time-synchronization message is triggered and the previous triggering of the time-synchronization message has not completed. Deasserts when on the rising edge of Issue_Time_Sync_DN.	False
Issue_<Name>_DN[a]	Output BOOL	Asserts for a single processing interval once the message is successfully completed. Completion of the message is dependent on the configuration of the message.	False
Issue_<Name>_Delay[a]	Output BOOL	Asserts when the message is triggered and the previous triggering of the message has not completed. Deasserts when on the rising edge of Issue_<Name>_DN.	False
Test_Command_Failure	Output BOOL	Asserts for a single processing interval if an error is detected in the test command response.	False

Pin Name	Pin Type	Description	Default
Time_Sync_Event_Received	Output BOOL	Asserts for a single processing interval when the client receives a time-synchronization event (ASDU Type = 103, COT = Spontaneous(3)) from the server.	False
Redundant_Changed	Output BOOL	Asserts for a single processing interval when the client sends the start data transfer activation request if Aux Server IP Address is defined.	False
Redundant_Active_Connection	Output UDINT	The connection that is in use. If no connection exists, contains the last connection that was connected. 1 = channel associated with Server_IP_Address. 2 = channel associated with Aux_Server_IP_Address.	0
Redundant_Ready_Count	Output UDINT	The number of redundant channels that are connected.	0
Buffer_Overflow	Output BOOL	Asserts for a single processing interval if any message operations were discarded because of queuing limits. A message will be discarded if multiple occurrences of the same operation have been queued prior to the completion of that message type.	False
Buffer_Overflow_Count	Output UDINT	Count indicating the number of times that the Buffer_Overflow pin has been asserted.	0
Restart_Local	Output BOOL	Asserts for a single processing interval when the client receives an end of initialization message with a cause of initialization, indicating that a local command/power-cycle restarted the server.	False
Restart_Remote	Output BOOL	Asserts for a single processing interval when the Client receives an end of initialization message with a cause of initialization, indicating a remote start of the server.	False
Error_Detected	Output BOOL	Asserts for a single processing interval when a message is received where the negative bit of the cause of initialization is asserted.	False
Last_Error_Code	Output Enum	Contains the last received message's COT cause field where its COT p/n = negative.	0

[a] <Name> = Each configured custom message.

IEC 60870-5-101/104 Server Configuration

Configure an IEC 60870-5-101 server to communicate via serial (or serial tunneled) or an IEC 60870-5-104 server to communicate via Ethernet to respond to polling IEC 60870-5-101/104 clients. Create binary input, analog input, and other tag types in an IEC 60870-5 shared map. Each IEC 60870-5-101 or IEC 60870-5-104 server is configurable for as many as eight sectors or address spaces that effectively perform as logical devices. Assign one unique shared map to each sector you have configured. You can configure a shared map to be unique to a configured IEC 60870-5-101/104 server, or you can share the same shared map with other configured IEC 60870-5-101/104 servers. You can create a maximum of 50 shared maps. Map data to those shared map tags by using the Tag Processor from client IED or other source tags in the RTAC database.

Insert a unique IEC 60870-5-101 or IEC 60870-5-104 server connection for each IEC 60870-5-101 or IEC 60870-5-104 client connected to the RTAC.

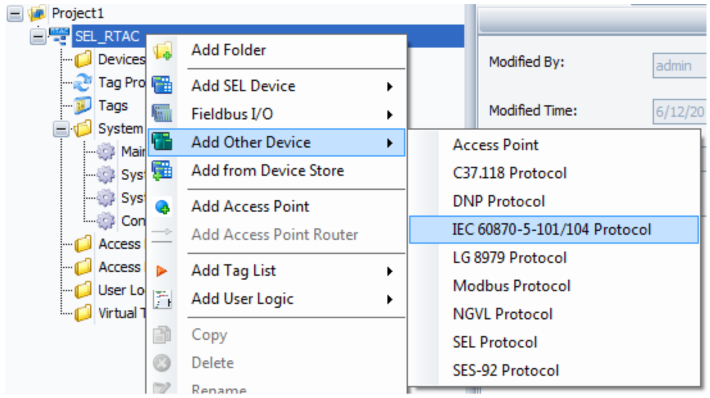


Figure 2.184 Insert IEC 60870-5-101/104 Server

Give the device connection a unique name and select the type of connection as shown in *Figure 2.3*. The **Settings** tab contains all the configurable items for communications. Check the **Description** column for settings details on each configuration item. Move the slider or hover your cursor over the description to see the entire description text. Type any necessary comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Shared Maps and Sector Settings

Each IEC 60870-5-101/104 server has a configurable number of sectors, each being analogous to a logical device within the IEC 60870-5-101/104 server. You must assign a IEC 60870-5-101/104 server shared map name to each sector to provide the data tags for that IEC 60870-5-101/104 server sector. You can configure as many as eight sectors, each with a unique Common Address, by selecting a shared sector map and adjusting the Cyclic Scan Period and Background Scan Period as needed. Configure double transmission of events by sector and by data type within the sector. Double transmission of events sends the event once without a time stamp and once with a time stamp.

Add IEC 60870-5-101/104 Shared Map

Step 1. Under the **Insert** tab, select **Tag Lists > IEC 6870-5-101/104 Server Shared Map**.

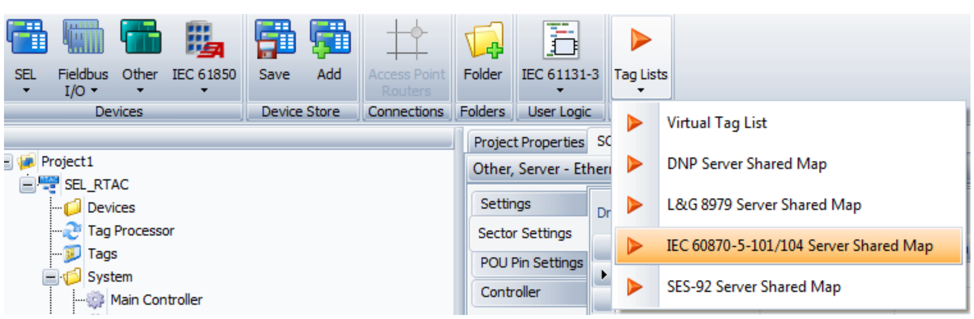


Figure 2.185 Add IEC 60870-5-101/104 Server Map

Step 2. Click on a device tag type tab to add and configure tags.

Step 3. Click + to add tags. Create only the necessary number of tags to optimize system performance.

Step 4. Change the names of the tags as necessary.

Step 5. Change other tag-related information as necessary.

Repeat *Step 2* through *Step 5* to configure all server tags. When you are finished, configure an IEC 60870-5-101/104 server to use this custom map by selecting **Shared Sector Map** in the **Sector Settings** tab. Although more than one IEC 60870-5-101/104 server can use the same map, each IEC 60870-5-101/104 server will manage separate event queues. Configure the Tag Processor to populate these server connection tags with actual values.

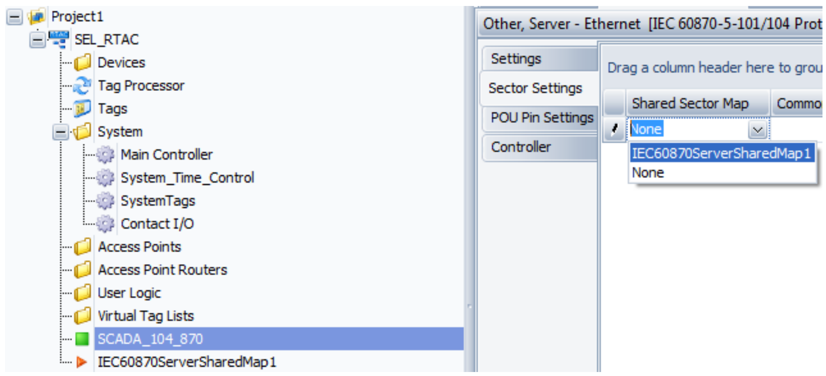


Figure 2.186 Select IEC 60870-5-101/104 Shared Map

IEC 60870-5-104 Server Failover

You can configure IEC 60870-5-104 servers to operate under the following failover scenarios.

Multiple Client IPs. Multiple clients can poll one IEC 60870-5-104 server if they are configured in a failover configuration so they poll the RTAC one at a time. In this configuration, the IEC 60870-5-104 server maintains one event queue per sector. Events reported to one client are not reported in the next reply, even if another poll comes from a different client.

If you set Allow Anonymous IP Clients to TRUE, the RTAC will allow any IEC 60870-5-104 client that can connect to the RTAC network to poll this IEC 60870-5-104 server, as long as the IEC 60870-5-104 sector addresses are correct. If you set Allow Anonymous IP Clients to FALSE, you can configure as many as 10 client IP addresses, separated by commas. In this configuration, the RTAC will only reply to any client that has one of the configured IP addresses.

Redundant. The IEC 60870-5-104 server supports redundant communications as defined in the IEC 60870-5-104 specification. Set Allow Anonymous IP Clients to FALSE, configure Client IP Address(es) as needed to authenticate valid polling clients, and set Enable Redundant = TRUE to enable additional TCP connections to the server.

IEC 60870-5-101 Server Dial-Out

You can configure an IEC 60870-5-101 serial server device to dial out automatically to report unsolicited data. Insert an IEC 60870-5-101 serial server connection and select **Advanced Settings**. *Figure 2.187* shows the settings that are enabled when Modem Connected = True. Notice the configuration supports an optional second phone number in case the RTAC does not get an answer when using the first phone number.

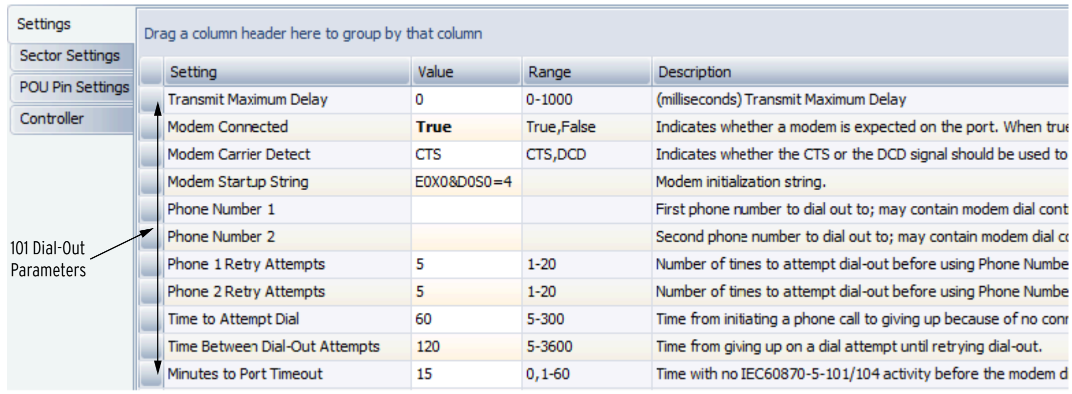


Figure 2.187 IEC 60870-5-101 Server Dial-Out

Device Tag Type Configuration Parameters

Each device tag type has parameters that must be configured correctly to ensure proper system operation. *Table 2.99–Table 2.110* describe the columns used to configure the IEC 60870-5-101/104 server tags. ACSELERATOR RTAC will gray out configuration fields that do not apply to a specific device.

Table 2.99 Common Device Tag Type Parameters

Parameter	Description	Default
Enable	Set this flag to True to enable processing of this tag. Set this flag to FALSE to disable processing.	True
Tag Name	This parameter describes the tag. You can change this name as necessary.	The default tag name contains the device name and tag type and is numbered 0–*n* tags.
Information Object Address (IOA)	Within a sector, the unique object address of a point. The total number of IOAs, plus Status IOAs, is 65535.	Contiguous from 0 to the point count.
Tag Alias	Enter an optional descriptive tag name in this field, and then reference this tag alias anywhere in the RTAC system in place of the actual tag name.	
Report By Exception	Set this flag to True to enable exception reporting for this tag.	Dependent on tag type
Report By Cyclic Scan	Set this flag to True to enable reporting this tag value by cyclic scan.	False
Report By Background Scan	Set this flag to True to enable reporting this tag value in a background scan.	False
Interrogation Group	Defines the General Interrogation Group for the point. If Global, the point is only reported when **General Interrogation All** command is received. If Group 1–16, the point is reported when the specific group or All is requested by a **General Interrogation** command. If None, the point is not reported by **General Interrogation** command.	Global
Comment	Optional user-entered comment field	

Table 2.100 Single-Point Status Input Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 65535 status input tags.	SPS
Status Value	The initialized value at startup	False

Table 2.101 Double-Point Status Input Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 65535 status input tags.	DPS
Status Value	The initialized value at startup	Intermediate

Table 2.102 Step Positions Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 65535 status input tags.	INS
Status Value	The initialized value at startup	0
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	63
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	56
H Limit	The high-alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	50
L Limit	The low-alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	–51
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	–57
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	–64

Table 2.103 Bitstrings Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 65535 status input tags.	INS
Status Value	The initialized value at startup	0
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	

Table 2.104 Measured Values Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 65535 status input tags.	MV
Measured Value Type	The 60870 measure value type of this tag	Normalized (ME_A)
Inst Magnitude	The initialized instantaneous value at startup	
Magnitude	The initialized dead-banded value at startup	
Dead Band Count	The number of units of change necessary to move the InstMagnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag.	
Zero Dead Band	The number of units at or below which Magnitude is forced to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.out OfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high-alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low-alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instmag < Min Value, ".q.detailQual.outOfRange" is set.	

Table 2.105 Integrated Totals Parameters

Parameters	Description	Default
Tag Type	See *Data Types on page 747* for more details.	LBCR
Actual Value	The initialized running value at startup	0
Frozen Value	The initialized value at startup	0

Table 2.106 Single Commands Parameters

Parameters	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 256 SBO tags.	SPC
Status Information Object Address (Status_IOA)	Defines the IOA used to access the status portion of the control. If Status_IOA = 0, the control status is not accessible.	Contiguous from 0 to the point count.
Status Value	The initialized value at startup	False
Control Model	Persist	

Parameters	Description	Default
Number of Pulses	The number of pulses operated for this control	
On Pulse Duration	The On duration of one pulse	
Off Pulse Duration	The Off duration of one pulse	Dependent on control operation
Short Pulse Duration	The duration of the pulse if the received command contains a short pulse qualifier. Short pulse duration must be < Long pulse duration.	1
Long Pulse Duration	The duration of the pulse if the received command contains a long pulse qualifier. Long pulse duration must be > Short pulse duration.	2000

Table 2.107 Double Commands Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 256 SBO tags.	DPC
Status Information Object Address (Status_IOA)	Defines the IOA used to access the status portion of the control. If Status_IOA = 0, the control status is not accessible.	Contiguous from 0 to the point count
Status Value	The initialized value at startup	Intermediate_state
Control Model	Persist	Dependent on control operation
Number of Pulses	The number of pulses operated for this control	1
On Pulse Duration	The On duration of one pulse	2000
Off Pulse Duration	The Off duration of one pulse	1
Short Pulse Duration	The duration of the pulse if the received command contains a short pulse qualifier. Short pulse duration must be < Long pulse duration.	100
Long Pulse Duration	The duration of the pulse if the received command contains a long pulse qualifier. Long pulse duration must be > Short pulse duration.	500

Table 2.108 Regulating Step Commands Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 256 SBO tags.	I870DC
Status Information Object Address (Status_IOA)	Defines the IOA used to access the status portion of the control. If Status_IOA = 0, the control status is not accessible.	Contiguous from 0 to the point count

Parameter	Description	Default
Status Value	The initialized value at startup	Intermediate_state
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	63
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high-alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low-alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < MinValue, ".q.detailQual.outOfRange" is set.	64
Control Value	The initialized value at startup	stop
Control Model	The method (pulse or persist) of the control operation	Pulse
Number of Pulses	The number of pulses operated for this control	1
On Pulse Duration	The On duration of one pulse	2000
Off Pulse Duration	The Off duration of one pulse	1
Step Size	Noneditable value indicating the amount of change when the control is operated	1
Short Pulse Duration	The duration of the pulse if the received command contains a short pulse qualifier. Short pulse duration must be < Long pulse duration.	100
Long Pulse Duration	The duration of the pulse if the received command contains a long pulse qualifier. Long pulse duration must be > Short pulse duration.	500

Table 2.109 Set Point Commands Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 4096 analog input tags.	APC
Setpoint Type	The type of set point control	Normalized (SE_A)
Inst Magnitude	The initialized instantaneous value at startup	
Magnitude	The initialized dead-banded value at startup	
Dead Band Count	The number of units of change necessary to move the InstMagnitude value into Magnitude. If |instMag – mag| > db, then mag := instMag.	

Parameter	Description	Default
Zero Dead Band	The number of units at or below which Magnitude is forced to zero. If |mag| < zeroDB, then mag := 0.	
Max Value	The maximum value allowed for this point. If instMag > Max Value, "q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high-alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low-alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < Min Value, ".q.detailQual.outOfRange" is set.	
Set Magnitude	The initialized set value at startup	

Table 2.110 Bitstring Commands Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Maximum 4096 analog input tags.	INC
Status Information Object Address (Status_IOA)	Defines the IOA used to access the status portion of the control. If Status_IOA = 0, the control status is not accessible.	Contiguous from 0 to the point count
Status Value	The initialized value at startup	0
Max Value	The maximum value allowed for this point. If instMag > Max Value, ".q.detailQual.outOfRange" is set.	
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	
H Limit	The high-alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	
L Limit	The low-alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	
Min Value	The minimum value allowed for this point. If instMag < MinValue, ".q.detailQual.outOfRange" is set.	

POU Pins

Use POU pin settings to view the present state of IEC 60870-5-101/104 server operation and to modify some of the behavior of the IEC 60870-5-101/104 protocol. Setting the **Visible** field to True will cause the POU pin to appear in the **Controller** tab. See *Table 2.111* for the settings descriptions.

Table 2.111 IEC 60870-5-101/104 POU Pin Settings

Pin Name	Pin Type	Description	Default
Accept_Time_Sync	Input BOOL	Set to True to allow accepting a clock synchronization command with COT equal to Activation (6).	True
Disable_Controls	Input BOOL	Disables controls through the protocol	False
Disable_Tag_Updates	Input BOOL	Disables updating of tag values	False
EN	Input BOOL	Enables this instance of IEC 60870-5-101/104	True
Reset_Statistics	Input BOOL	Resets all communications statistics.	False
CI_Freeze_IEC60870Server-SharedMap1	Output BOOL	Asserted for one processing cycle when a Counter Interrogation request is received, Qualifier of Counter Interrogation (QCC) option = freeze, freeze with reset, reset (1,2,3).	False
CI_Reset_IEC60870Server-SharedMap1	Output BOOL	Asserted for one processing cycle when a Counter Interrogation request is received, Qualifier of Counter Interrogation (QCC) option = freeze, freeze with reset, reset (1,2,3)	False
Controls_Disabled	Output BOOL	Indicates controls are disabled	False
ENO	Output BOOL	Indicates this instance of IEC 60870-5-101/104 is running	False
Error_Detected	Output BOOL	Pulsed for one processing cycle to indicate an error, indicated in Last_Error_Code, has been detected	False
Event_Buffer_Overflow	Output BOOL	BOOL	False
Event_Buffer_Overflow_Count	Output UDINT	UDINT	0
Last_Error_Code	Output DINT	An enumerated value of the last error detected, where COT is one of the following: Unknown Type ID (44) Unknown COT (45) Unknown Common Address (46) Unknown IOA (47)	44
Message_Failure	Output BOOL	Asserts for a single-processing interval following a message failure	False
Message_Failure_Count	Output UDINT	Increments value by one each time Message_Failure is asserted	0
Message_Received_Count	Output UDINT	A counter of the number of messages received	0
Message_Sent_Count	Output UDINT	A counter of the number of messages sent	0
Message_Success_Count	Output UDINT	A running sum of the messages successfully sent or received	0
Offline	Output BOOL	Set to True if data link activity timeout expires or a frame has not been received for a period of time, defined by Communications_Offline_Timer.	True

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

IEC 60870-5-101/104 Compatibility

The application of IEC 60870-5-101 and 60870-5-104 in the RTAC supports the following function codes:

Basic Application Functions Supported

- Remote initialization
- Cyclic data transmission
- Read procedure
- Spontaneous transmission

Command Transmission Supported

- Direct command transmission
- Direct set point command transmission
- Select and execute command
- Select and execute set point command
- C_SE ACCTTERM used
- No additional definition
- Short-pulse duration
- Long-pulse duration
- Persistent output

Transmission of Integrated Totals Supported

- Mode A: local freeze with spontaneous transmission
- Mode B: local freeze with counter interrogation
- Mode C: freeze and transmit by counter-interrogation commands
- Mode D: freeze by counter-interrogation command, frozen values reported spontaneously
- Counter read
- Counter freeze without reset
- Counter freeze with reset
- Counter reset
- General request counter
- Request counter Group 1
- Request counter Group 2
- Request counter Group 3
- Request counter Group 4

Clock Synchronization Details

- Clock synchronization supported
- Day of week used

- ➤ RES1, GEN (time tag substituted/not substituted) used
- ➤ SU-bit (summertime) used

Interoperability Statement for IEC 60870-5-101 Client for the SEL RTAC

Introduction

This document specifies the capabilities and features of the SEL-implemented IEC 60870-5-104 Server.

This document applies specifically to the following RTAC models and firmware versions.

Model	Firmware Version
SEL-3505 RTAC	SEL-3505-R136-V0-Z001001-D20160624
SEL-3505-3 RTAC	SEL-3505-3-R136-V0-Z001001-D20160624
SEL-3530 RTAC	SEL-3530-R136-V0-Z001001-D20160624
SEL-3530-4 RTAC	SEL-3530-4-R136-V0-Z001001-D20160624
SEL-2241 RTAC	SEL-2241-R136-V0-Z001001-20160624
SEL-3532 RTAC	SEL-3532-N-R136-V0-Z001001-D20160624
SEL-3555 RTAC	SEL-3555-R136-V0-Z001001-D20160624

Interoperability

System or Device

	System definition
X	Controlling station definition (master)
	Controlled station definition (slave)

Network Configuration

X	Point-to-point	X	Multipoint-partyline
X	Multiple point-to-point		Multipoint-start

Physical Layer

Transmission Speed (Control Direction)

Unbalanced Interchange Circuit V.24/V.28 Standard		Unbalanced Interchange Circuit V.24/V.28 Recommended if > 1200 bit/s		Balanced Interchange Circuit X.24/X.27			
	100 bit/s	X	2400 bit/s	X	2400 bit/s	X	56000 bit/s
X	200 bit/s	X	4800 bit/s	X	4800 bit/s		64000 bit/s
X	300 bit/s	X	9600 bit/s	X	9600 bit/s		
X	600 bit/s			X	19200 bit/s		
X	1200 bit/s			X	38400 bit/s		

Transmission Speed

Unbalanced Interchange Circuit V.24/V.28 Standard		Unbalanced Interchange Circuit V.24/V.28 Recommended if > 1200 bit/s		Balanced Interchange Circuit X.24/X.27			
	100 bit/s	X	2400 bit/s	X	2400 bit/s	X	56000 bit/s
X	200 bit/s	X	4800 bit/s	X	4800 bit/s		64000 bit/s
X	300 bit/s	X	9600 bit/s	X	9600 bit/s		
X	600 bit/s			X	19200 bit/s		
X	1200 bit/s			X	38400 bit/s		

Link Layer

Frame format FT 1.2, single character 1, and the fixed time-out interval are used exclusively in this companion standard.

Link Transmission		**Address Field of the Link**	
X	Balanced transmission	X	Not present (balanced transmission only)
X	Unbalanced transmission	X	One octet
Frame Length		X	Two octets
255	Maximum length L (control direction)		Structured
255	Maximum length L (monitor direction)	X	Unstructured
0–2250 s	Time during which repetitions are permitted (Trp) or number of repetitions.		

For an unbalanced link layer, the server returns the following ASDU types in Class 2 messages (low priority) with the indicated causes of transmission:

	The standard assignment of ASDUs to Class 2 messages is used as follows:

Type Identification	Cause of Transmission
9, 11, 13, 21	<1>
9, 11, 13, 21	<2>

	A special assignment of ASDUs to Class 2 messages is as follows:

NOTE
In response to a Class 2 poll, a controlled station can respond with Class 1 data when there are no Class 2 data available.

Type Identification	Cause of Transmission
1, 3, 5, 7	<1>
1, 3, 5, 7	<2>

Application Layer

Transmission Mode for Application Data

Mode 1 (least significant octet first), according to Section 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common Address of ASDU

X	One octet	X	Two octets

Information Object Addresses

X	One octet		Structured
X	Two octets	X	Unstructured
X	Three octets		

Cause of Transmission

X	One octet	X	Two octets (with originator address). Set the originator address to zero if it is not used.

Length of APDU

253	Maximum length of APDU per system in control direction
253	Maximum length of APDU per system in monitor direction

Selection of Standard ASDUs

Process Information in Monitor Direction

X	<1> := Single-point information	M_SP_NA_1
X	<2> := Single-point information with time tag	M_SP_TA_1
X	<3> := Double-point information	M_DP_NA_1
X	<4> := Double-point information with time tag	M_DP_TA_1
X	<5> := Step position information	M_ST_NA_1
X	<6> := Step position information with time tag	M_ST_TA_1

X	<7> := Bitstring of 32 bits	M_BO_NA_1
X	<8> := Bitstring of 32 bits with time tag	M_BO_TA_1
X	<9> := Measured value, normalized value	M_ME_NA_1
X	<10> := Measured value, normalized value with time tag	M_ME_TA_1
X	<11> := Measured value, scaled value	M_ME_NB_1
X	<12> := Measured value, scaled value with time tag	M_ME_TB_1
X	<13> := Measured value, short floating point value	M_ME_NC_1
X	<14> := Measured value, short floating point value with time tag	M_ME_TC_1
X	<15> := Integrated totals	M_IT_NA_1
X	<16> := Integrated totals with time tag	M_IT_TA_1
	<17> := Event of protection equipment with time tag	M_EP_TA_1
	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1
	<20> := Packed single-point information with status change detection	M_PS_NA_1
X	<21> := Measured value, normalized value without quality descriptor	M_ME_ND_1
X	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
X	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
X	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
X	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
X	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
X	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
X	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
X	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1
	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Process Information in Control Direction

X	<45> := Single command	C_SC_NA_1
X	<46> := Double command	C_DC_NA_1
X	<47> := Regulating step command	C_RC_NA_1
X	<48> := Set point command, normalized value	C_SE_NA_1
X	<49> := Set point command, scaled value	C_SE_NB_1
X	<50> := Set point command, short floating point value	C_SE_NC_1
X	<51> := Bitstring of 32 bits	C_BO_NA_1
	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
	<59> := Double command with time tag CP56Time2a	C_DC_TA_1
	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1

	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
	<64> := Bitstring of 32 bits with time tag CP56Time2a	C_BO_TA_1

System Information in Monitor Direction

X	<70> := End of initialization	M_EI_NA_1

System Information in Control Direction

X	<100> := Interrogation command	C_IC_NA_1
X	<101> := Counter-interrogation command	C_CI_NA_1
X	<102> := Read command	C_RD_NA_1
X	<103> := Clock synchronization command	C_CS_NA_1
X	<104> := Test command	C_TS_NA_1
X	<105> := Reset process command	C_RP_NA_1
X	<106> := Delay acquisition command	C_CD_NA_1
	<107> := Test command with time tag CP56Time2a	C_TS_TA_1

Parameter in Control Direction

	<110> := Parameter of measured value, normalized value	P_ME_NA_1
	<111> := Parameter of measured value, scaled value	P_ME_NB_1
	<112> := Parameter of measured value, short floating point value	P_ME_NC_1
	<113> := Parameter activation	P_AC_NA_1

File Transfer

	<120> := File ready	F_FR_NA_1
	<121> := Section ready	F_SR_NA_1
	<122> := Call directory, select file, call file, call section	F_SC_NA_1
	<123> := Last section, last segment	F_LS_NA_1
	<124> := Ack file, ack section	F_AF_NA_1
	<125> := Segment	F_SG_NA_1
	<126> := Directory {blank or X, only available in monitor (standard) direction}	F_DR_TA_1
	<127> := Query Log – Request archive file	F_SC_NB_1

Type Identification and Cause of Transmission Assignments

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1	X	X	X		X						X	X		X					
<2>	M_SP_TA_1	X	X	X		X						X	X		X					
<3>	M_DP_NA_1	X	X	X		X						X	X		X					
<4>	M_DP_TA_1	X	X	X		X						X	X		X					
<5>	M_ST_NA_1	X	X	X		X						X	X		X					

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<6>	M_ST_TA_1	X	X	X		X						X	X		X					
<7>	M_BO_NA_1	X	X	X		X						X	X		X					
<8>	M_BO_TA_1	X	X	X		X						X	X		X					
<9>	M_ME_NA_1	X	X	X		X						X	X		X					
<10>	M_ME_TA_1	X	X	X		X						X	X		X					
<11>	M_ME_NB_1	X	X	X		X						X	X		X					
<12>	M_ME_TB_1	X	X	X		X						X	X		X					
<13>	M_ME_NC_1	X	X	X		X						X	X		X					
<14>	M_ME_TC_1	X	X	X		X						X	X		X					
<15>	M_IT_NA_1			X		X										X				
<16>	M_IT_TA_1			X		X										X				
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			
<19>	M_EP_TC_1																			
<20>	M_PS_NA_1	X	X	X		X									X					
<21>	M_ME_ND_1	X	X	X		X									X					
<30>	M_SP_TB_1	X	X	X		X						X	X		X					
<31>	M_DP_TB_1	X	X	X		X						X	X		X					
<32>	M_ST_TB_1	X	X	X		X						X	X		X					
<33>	M_BO_TB_1	X	X	X		X						X	X		X					
<34>	M_ME_TD_1	X	X	X		X						X	X		X					
<35>	M_ME_TE_1	X	X	X		X						X	X		X					
<36>	M_ME_TF_1	X	X	X		X						X	X		X					
<37>	M_IT_TB_1			X		X										X				
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X			X							X	X	X
<46>	C_DC_NA_1						X	X			X							X	X	X
<47>	C_RC_NA_1						X	X			X							X	X	X
<48>	C_SE_NA_1						X	X			X							X	X	X
<49>	C_SE_NB_1						X	X			X							X	X	X
<50>	C_SE_NC_1						X	X			X							X	X	X
<51>	C_BO_NA_1						X	X			X							X	X	X

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<58>	C_SC_TA_1																			
<59>	C_DC_TA_1																			
<60>	C_RC_TA_1																			
<61>	C_SE_TA_1																			
<62>	C_SE_TB_1																			
<63>	C_SE_TC_1																			
<64>	C_BO_TA_1																			
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X			X							X	X	X
<101>	C_CI_NA_1						X	X			X							X	X	X
<102>	C_RD_NA_1					X												X	X	X
<103>	C_CS_NA_1			X			X	X										X	X	X
<104>	C_TS_NA_1						X	X										X	X	X
<105>	C_RP_NA_1						X	X										X	X	X
<106>	C_CD_NA_1			X			X	X										X	X	X
<107>	C_TS_TA_1																			
<110>	P_ME_NA_1																			
<111>	P_ME_NB_1																			
<112>	P_ME_NC_1																			
<113>	P_AC_NA_1																			
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1																			
<127>	F_SC_NB_1																			

Basic Application Functions

Station Initialization

X	Remote initialization

Cyclic Data Transmission

X	Cyclic data transmission

Read Procedure

X	Read procedure

Spontaneous Transmission

X	Spontaneous transmission

Double Transmission of Information Objects (Cause of Transmission Equals Spontaneous)

The following type identifications can be transmitted in succession as the result of a single status change for an information object. A project-specific list defines the particular information object addresses for which double transmission is enabled.

	Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1, and M_PS_NA_1
	Double-point information M_DP_NA_1, M_DP_TA_1, and M_DP_TB_1
	Step position information M_ST_NA_1, M_ST_TA_1, and M_ST_TB_1
	Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1, and M_BO_TB_1
	Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1, and M_ME_TD_1
	Measured value, scaled value M_ME_NB_1, M_ME_TB_1, and M_ME_TE_1
	Measured value, short floating point number M_ME_NC_1, M_ME_TC_1, and M_ME_TF_1

Station Interrogation

X	Global				
X	Group 1	X	Group 7	X	Group 13
X	Group 2	X	Group 8	X	Group 14
X	Group 3	X	Group 9	X	Group 15
X	Group 4	X	Group 10	X	Group 16
X	Group 5	X	Group 11		
X	Group 6	X	Group 12		

Clock Synchronization

X	Clock synchronization
X	Day of week used
X	RES1, GEN (time tag substituted/not substituted) used
X	SU-bit (summertime) used

Command Transmission

X	Direct command transmission
X	Direct set point command transmission
X	Select and execute command
X	Select and execute set point command
X	C_SE ACTTERM used
X	No additional definition
X	Short-pulse duration (duration determined by a system parameter in the outstation)
X	Long-pulse duration (duration determined by a system parameter in the outstation)
X	Persistent output
	Supervision of maximum delay in command direction of commands and set point commands
	Maximum allowable delay of commands and set point commands

Transmission of Integrated Totals

X	Mode A: Local freeze with spontaneous transmission
X	Mode B: Local freeze with counter interrogation
X	Mode C: Freeze and transmit by counter-interrogation commands
X	Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously
X	Counter read
X	Counter freeze without reset
X	Counter freeze with reset
X	Counter reset
X	General request
X	Request Counter Group 1
X	Request Counter Group 2
X	Request Counter Group 3
X	Request Counter Group 4

Parameter Loading

	Threshold value
	Smoothing factor
	Low limit for transmission of measured values
	High limit for transmission of measured values

Parameter Activation

	Act/deact of persistent cyclic or periodic transmission of the addressed object

Test Procedure

X	Test procedure

File Transfer

File transfer in monitor direction

	Transparent file
	Transmission of disturbance data of protection equipment
	Transmission of sequences of events
	Transmission of sequences of recorded analog values

File transfer in control direction

	Transparent file

Background Scan

X	Background scan

Acquisition of Transmission Delay

X	Acquisition of transmission delay

Interoperability Statement for IEC 60870-5-104 Client for the SEL RTAC

Introduction

This document specifies the capabilities and features of the SEL-implemented IEC 60870-5-104 Server.

This document applies specifically to the following RTAC models and firmware versions.

Model	Firmware Version
SEL-3505 RTAC	SEL-3505-R136-V0-Z001001-D20160624
SEL-3505-3 RTAC	SEL-3505-3-R136-V0-Z001001-D20160624
SEL-3530 RTAC	SEL-3530-R136-V0-Z001001-D20160624
SEL-3530-4 RTAC	SEL-3530-4-R136-V0-Z001001-D20160624

Model	Firmware Version
SEL-2241 RTAC	SEL-2241-R136-V0-Z001001-20160624
SEL-3532 RTAC	SEL-3532-N-R136-V0-Z001001-D20160624
SEL-3555 RTAC	SEL-3555-R136-V0-Z001001-D20160624

Interoperability

System or Device

	System definition
X	Controlling station definition (master)
	Controlled station definition (slave)

Network Configuration (Not Applicable)

	Point-to-point		Multipoint-partyline
	Multiple point-to-point		Multipoint-start

Physical Layer (Not Applicable)

Transmission Speed (Control Direction)

Unbalanced Interchange Circuit V.24/V.28 Standard		Unbalanced Interchange Circuit V.24/V.28 Recommended if > 1200 bit/s		Balanced Interchange Circuit X.24/X.27			
	100 bit/s		2400 bit/s		2400 bit/s		56000 bit/s
	200 bit/s		4800 bit/s		4800 bit/s		64000 bit/s
	300 bit/s		9600 bit/s		9600 bit/s		
	600 bit/s				19200 bit/s		
	1200 bit/s				38400 bit/s		

Transmission Speed (Monitor Direction)

Unbalanced Interchange Circuit V.24/V.28 Standard		Unbalanced Interchange Circuit V.24/V.28 Recommended if > 1200 bit/s		Balanced Interchange Circuit X.24/X.27			
	100 bit/s		2400 bit/s		2400 bit/s		56000 bit/s
	200 bit/s		4800 bit/s		4800 bit/s		64000 bit/s
	300 bit/s		9600 bit/s		9600 bit/s		
	600 bit/s				19200 bit/s		
	1200 bit/s				38400 bit/s		

Link Layer (Not Applicable)

Frame format FT 1.2, single character 1, and the fixed time-out interval are used exclusively in this companion standard.

Link Transmission	Address Field of the Link
Balanced transmission	Not present (balanced transmission only)
Unbalanced transmission	One octet
Frame Length	Two octets
Maximum length L (control direction)	Structured
Maximum length L (monitor direction)	Unstructured
Time during which repetitions are permitted (Trp) or number of repetitions.	

For an unbalanced link layer, the server returns the following ASDU types in Class 2 messages (low priority) with the indicated causes of transmission:

	The standard assignment of ASDUs to Class 2 messages is used as follows:

Type Identification	Cause of Transmission

	A special assignment of ASDUs to Class 2 messages is as follows:

NOTE
In response to a Class 2 poll, a controlled station can respond with Class 1 data when there are no Class 2 data available.

Type Identification	Cause of Transmission

Application Layer

Transmission Mode for Application Data

Mode 1 (least significant octet first), according to Section 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common Address of ASDU

X	One octet	X	Two octets

Information Object Addresses

X	One octet		Structured
X	Two octets	X	Unstructured
X	Three octets		

Cause of Transmission

X	One octet	X	Two octets (with originator address). Set the originator address to zero if it is not used.

Length of APDU

253	Maximum length of APDU per system in control direction
253	Maximum length of APDU per system in monitor direction

Selection of Standard ASDUs

Process Information in Monitor Direction

X	<1> := Single-point information	M_SP_NA_1
X	<2> := Single-point information with time tag	M_SP_TA_1
X	<3> := Double-point information	M_DP_NA_1
X	<4> := Double-point information with time tag	M_DP_TA_1
X	<5> := Step position information	M_ST_NA_1
X	<6> := Step position information with time tag	M_ST_TA_1
X	<7> := Bitstring of 32 bits	M_BO_NA_1
X	<8> := Bitstring of 32 bits with time tag	M_BO_TA_1
X	<9> := Measured value, normalized value	M_ME_NA_1
X	<10> := Measured value, normalized value with time tag	M_ME_TA_1
X	<11> := Measured value, scaled value	M_ME_NB_1
X	<12> := Measured value, scaled value with time tag	M_ME_TB_1
X	<13> := Measured value, short floating point value	M_ME_NC_1
X	<14> := Measured value, short floating point value with time tag	M_ME_TC_1
X	<15> := Integrated totals	M_IT_NA_1
X	<16> := Integrated totals with time tag	M_IT_TA_1
	<17> := Event of protection equipment with time tag	M_EP_TA_1
	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1

X	<20> := Packed single-point information with status change detection	M_PS_NA_1
X	<21> := Measured value, normalized value without quality descriptor	M_ME_ND_1
X	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
X	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
X	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
X	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
X	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
X	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
X	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
X	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1
	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Process Information in Control Direction

X	<45> := Single command	C_SC_NA_1
X	<46> := Double command	C_DC_NA_1
X	<47> := Regulating step command	C_RC_NA_1
X	<48> := Set point command, normalized value	C_SE_NA_1
X	<49> := Set point command, scaled value	C_SE_NB_1
X	<50> := Set point command, short floating point value	C_SE_NC_1
X	<51> := Bitstring of 32 bits	C_BO_NA_1
X	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
X	<59> := Double command with time tag CP56Time2a	C_DC_TA_1
X	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
X	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
X	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
X	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
X	<64> := Bitstring of 32 bits with time tag CP56Time2a	C_BO_TA_1

System Information in Monitor Direction

X	<70> := End of initialization	M_EI_NA_1

System Information in Control Direction

X	<100> := Interrogation command	C_IC_NA_1
X	<101> := Counter-interrogation command	C_CI_NA_1
X	<102> := Read command	C_RD_NA_1
X	<103> := Clock synchronization command	C_CS_NA_1
X	<104> := Test command	C_TS_NA_1
X	<105> := Reset process command	C_RP_NA_1
X	<106> := Delay acquisition command	C_CD_NA_1

X	<107> := Test command with time tag CP56Time2a	C_TS_TA_1

Parameter in Control Direction

	<110> := Parameter of measured value, normalized value	P_ME_NA_1
	<111> := Parameter of measured value, scaled value	P_ME_NB_1
	<112> := Parameter of measured value, short floating point value	P_ME_NC_1
	<113> := Parameter activation	P_AC_NA_1

File Transfer

	<120> := File ready	F_FR_NA_1
	<121> := Section ready	F_SR_NA_1
	<122> := Call directory, select file, call file, call section	F_SC_NA_1
	<123> := Last section, last segment	F_LS_NA_1
	<124> := Ack file, ack section	F_AF_NA_1
	<125> := Segment	F_SG_NA_1
	<126> := Directory {blank or X, only available in monitor (standard) direction}	F_DR_TA_1
	<127> := Query Log – Request archive file	F_SC_NB_1

Type Identification and Cause of Transmission Assignments

Type Identification		Cause of Transmission																
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1	X	X	X		X						X	X		X					
<2>	M_SP_TA_1	X	X	X		X						X	X		X					
<3>	M_DP_NA_1	X	X	X		X						X	X		X					
<4>	M_DP_TA_1	X	X	X		X						X	X		X					
<5>	M_ST_NA_1	X	X	X		X						X	X		X					
<6>	M_ST_TA_1	X	X	X		X						X	X		X					
<7>	M_BO_NA_1	X	X	X		X						X	X		X					
<8>	M_BO_TA_1	X	X	X		X						X	X		X					
<9>	M_ME_NA_1	X	X	X		X						X	X		X					
<10>	M_ME_TA_1	X	X	X		X						X	X		X					
<11>	M_ME_NB_1	X	X	X		X						X	X		X					
<12>	M_ME_TB_1	X	X	X		X						X	X		X					
<13>	M_ME_NC_1	X	X	X		X						X	X		X					
<14>	M_ME_TC_1	X	X	X		X						X	X		X					
<15>	M_IT_NA_1			X		X										X				
<16>	M_IT_TA_1			X		X										X				
<17>	M_EP_TA_1																			

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<18>	M_EP_TB_1																			
<19>	M_EP_TC_1																			
<20>	M_PS_NA_1	X	X	X		X									X					
<21>	M_ME_ND_1	X	X	X		X									X					
<30>	M_SP_TB_1	X	X	X		X						X	X		X					
<31>	M_DP_TB_1	X	X	X		X						X	X		X					
<32>	M_ST_TB_1	X	X	X		X						X	X		X					
<33>	M_BO_TB_1	X	X	X		X						X	X		X					
<34>	M_ME_TD_1	X	X	X		X						X	X		X					
<35>	M_ME_TE_1	X	X	X		X						X	X		X					
<36>	M_ME_TF_1	X	X	X		X						X	X		X					
<37>	M_IT_TB_1			X		X										X				
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X			X							X	X	X
<46>	C_DC_NA_1						X	X			X							X	X	X
<47>	C_RC_NA_1						X	X			X							X	X	X
<48>	C_SE_NA_1						X	X			X							X	X	X
<49>	C_SE_NB_1						X	X			X							X	X	X
<50>	C_SE_NC_1						X	X			X							X	X	X
<51>	C_BO_NA_1						X	X			X							X	X	X
<58>	C_SC_TA_1						X	X			X							X	X	X
<59>	C_DC_TA_1						X	X			X							X	X	X
<60>	C_RC_TA_1						X	X			X							X	X	X
<61>	C_SE_TA_1						X	X			X							X	X	X
<62>	C_SE_TB_1						X	X			X							X	X	X
<63>	C_SE_TC_1						X	X			X							X	X	X
<64>	C_BO_TA_1						X	X			X							X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X			X							X	X	X
<101>	C_CI_NA_1						X	X			X							X	X	X
<102>	C_RD_NA_1					X												X	X	X
<103>	C_CS_NA_1			X			X	X										X	X	X

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<104>	C_TS_NA_1						X	X										X	X	X
<105>	C_RP_NA_1						X	X										X	X	X
<106>	C_CD_NA_1																			
<107>	C_TS_TA_1						X	X										X	X	X
<110>	P_ME_NA_1																			
<111>	P_ME_NB_1																			
<112>	P_ME_NC_1																			
<113>	P_AC_NA_1																			
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1																			
<127>	F_SC_NB_1																			

Basic Application Functions

Station Initialization

X	Remote initialization

Cyclic Data Transmission

X	Cyclic data transmission

Read Procedure

X	Read procedure

Spontaneous Transmission

X	Spontaneous transmission

Double Transmission of Information Objects (Cause of Transmission Equals Spontaneous)

The following type identifications can be transmitted in succession as the result of a single status change for an information object. A project-specific list defines the particular information object addresses for which double transmission is enabled.

	Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1, and M_PS_NA_1
	Double-point information M_DP_NA_1, M_DP_TA_1, and M_DP_TB_1
	Step position information M_ST_NA_1, M_ST_TA_1, and M_ST_TB_1
	Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1, and M_BO_TB_1
	Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1, and M_ME_TD_1
	Measured value, scaled value M_ME_NB_1, M_ME_TB_1, and M_ME_TE_1
	Measured value, short floating point number M_ME_NC_1, M_ME_TC_1, and M_ME_TF_1

Station Interrogation

X	Global				
X	Group 1	X	Group 7	X	Group 13
X	Group 2	X	Group 8	X	Group 14
X	Group 3	X	Group 9	X	Group 15
X	Group 4	X	Group 10	X	Group 16
X	Group 5	X	Group 11		
X	Group 6	X	Group 12		

Clock Synchronization

X	Clock synchronization
X	Day of week used
X	RES1, GEN (time tag substituted/not substituted) used
X	SU-bit (summertime) used

Command Transmission

X	Direct command transmission
X	Direct set point command transmission
X	Select and execute command
X	Select and execute set point command
X	C_SE ACTTERM used
X	No additional definition
X	Short-pulse duration (duration determined by a system parameter in the outstation)
X	Long-pulse duration (duration determined by a system parameter in the outstation)

X	Persistent output
X	Supervision of maximum delay in command direction of commands and set point commands
	Maximum allowable delay of commands and set point commands

Transmission of Integrated Totals

X	Mode A: Local freeze with spontaneous transmission
X	Mode B: Local freeze with counter interrogation
X	Mode C: Freeze and transmit by counter-interrogation commands
X	Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously
X	Counter read
X	Counter freeze without reset
X	Counter freeze with reset
X	Counter reset
X	General request
X	Request Counter Group 1
X	Request Counter Group 2
X	Request Counter Group 3
X	Request Counter Group 4

Parameter Loading

	Threshold value
	Smoothing factor
	Low limit for transmission of measured values
	High limit for transmission of measured values

Parameter Activation

	Act/deact of persistent cyclic or periodic transmission of the addressed object

Test Procedure

	Test procedure

File Transfer

File transfer in monitor direction

	Transparent file
	Transmission of disturbance data of protection equipment

	Transmission of sequences of events
	Transmission of sequences of recorded analog values

File transfer in control direction

	Transparent file

Background Scan

X	Background scan

Acquisition of Transmission Delay

	Acquisition of transmission delay

Definition of Time-Outs

Parameter	Default Value	Remarks	Selected Value
t_0	15 s	Time-out of connection establishment	1–60 s
t_1	15 s	Time-out of send or test APDUs	1–255 s
t_2	10 s	Time-out for acknowledgments in case of no data messages $t_2 < t_1$	1–255 s
t_3	20 s	Time-out for sending test frames in case of a long idle state	Disable, 1–172800 s

Maximum Number of Outstanding I Format APDUs k and Latest Acknowledge APDUs (w)

Parameter	Default Value	Remarks	Selected Value
k	12 APDUs	Maximum difference receive sequence number to send state variable	1–24
w	8 APDUs	Latest acknowledgment(s) after receiving *w* I format APDUs	1–24

Portnumber

Parameter	Value	Remarks
Portnumber	2404	1–65534

Redundant Connections

0	Number N of redundancy group connections used

RFC 2200 Suite

RFC 2200 is an official Internet standard that describes the state of Internet protocol standardization as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

X	Ethernet 802.3
	Serial X.21 interface
	Other selection from RFC 2200:
	List of valid documents from RFC 2200
	1. ....
	2. ....
	3. ....
	4. ....
	5. ....
	6. ....
	7. etc.

Interoperability Statement for IEC 60870-5-101 Server for the SEL RTAC

Introduction

This document specifies the capabilities and features of the SEL-implemented IEC 60870-5-101 Server.

This document applies specifically to the following RTAC models and firmware versions.

Model	Firmware Version
SEL-3505 RTAC	SEL-3505-R136-V0-Z001001-D20160624
SEL-3505-3 RTAC	SEL-3505-3-R136-V0-Z001001-D20160624
SEL-3530 RTAC	SEL-3530-R136-V0-Z001001-D20160624
SEL-3530-4 RTAC	SEL-3530-4-R136-V0-Z001001-D20160624
SEL-2241 RTAC	SEL-2241-R136-V0-Z001001-20160624
SEL-3532 RTAC	SEL-3532-N-R136-V0-Z001001-D20160624
SEL-3555 RTAC	SEL-3555-R136-V0-Z001001-D20160624

Interoperability

System or Device

	System definition
	Controlling station definition (master)
X	Controlled station definition (slave)

Network Configuration

X	Point-to-point	X	Multipoint-partyline
X	Multiple point-to-point		Multipoint-start

Physical Layer

Transmission Speed (Control Direction)

Unbalanced Interchange Circuit V.24/V.28 Standard		Unbalanced Interchange Circuit V.24/V.28 Recommended if > 1200 bit/s		Balanced Interchange Circuit X.24/X.27			
	100 bit/s	X	2400 bit/s	X	2400 bit/s	X	56000 bit/s
X	200 bit/s	X	4800 bit/s	X	4800 bit/s		64000 bit/s
X	300 bit/s	X	9600 bit/s	X	9600 bit/s	X	115000 bit/s
X	600 bit/s			X	19200 bit/s		
X	1200 bit/s			X	38400 bit/s		

Transmission Speed (Monitor Direction)

Unbalanced Interchange Circuit V.24/V.28 Standard		Unbalanced Interchange Circuit V.24/V.28 Recommended if > 1200 bit/s		Balanced Interchange Circuit X.24/X.27			
	100 bit/s	X	2400 bit/s	X	2400 bit/s	X	56000 bit/s
X	200 bit/s	X	4800 bit/s	X	4800 bit/s		64000 bit/s
X	300 bit/s	X	9600 bit/s	X	9600 bit/s	X	115000 bit/s
X	600 bit/s			X	19200 bit/s		
X	1200 bit/s			X	38400 bit/s		

Link Layer

Frame format FT 1.2, single character 1, and the fixed time-out interval are used exclusively in this companion standard.

Link Transmission		**Address Field of the Link**	
X	Balanced transmission	X	Not present (balanced transmission only)
X	Unbalanced transmission	X	One octet
Frame Length		X	Two octets
255	Maximum length L (control direction)		Structured
255	Maximum length L (monitor direction)	X	Unstructured
	Time during which repetitions are permitted (Trp) or number of repetitions.		

For an unbalanced link layer, the server returns the following ASDU types in Class 2 messages (low priority) with the indicated causes of transmission:

X	The standard assignment of ASDUs to Class 2 messages is used as follows:

Type Identification	Cause of Transmission
9, 11, 13, 21	<1>
9, 11, 13, 21	<2>

X	A special assignment of ASDUs to Class 2 messages is as follows:

NOTE

In response to a Class 2 poll, a controlled station can respond with Class 1 data when there are no Class 2 data available.

Type Identification	Cause of Transmission
1, 3, 5, 7	<1>
1, 3, 5, 7	<2>

Application Layer

Transmission Mode for Application Data

Mode 1 (least significant octet first), according to Section 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common Address of ASDU

X	One octet	X	Two octets

Information Object Addresses

X	One octet		Structured
X	Two octets	X	Unstructured
X	Three octets		

Cause of Transmission

X	One octet	X	Two octets (with originator address). Set the originator address to zero if it is not used.

Selection of Standard ASDUs

Process Information in Monitor Direction

X	<1> := Single-point information	M_SP_NA_1
X	<2> := Single-point information with time tag	M_SP_TA_1
X	<3> := Double-point information	M_DP_NA_1
X	<4> := Double-point information with time tag	M_DP_TA_1
X	<5> := Step position information	M_ST_NA_1
X	<6> := Step position information with time tag	M_ST_TA_1
X	<7> := Bitstring of 32 bits	M_BO_NA_1
X	<8> := Bitstring of 32 bits with time tag	M_BO_TA_1
X	<9> := Measured value, normalized value	M_ME_NA_1
X	<10> := Measured value, normalized value with time tag	M_ME_TA_1
X	<11> := Measured value, scaled value	M_ME_NB_1
X	<12> := Measured value, scaled value with time tag	M_ME_TB_1
X	<13> := Measured value, short floating point value	M_ME_NC_1
X	<14> := Measured value, short floating point value with time tag	M_ME_TC_1
X	<15> := Integrated totals	M_IT_NA_1
X	<16> := Integrated totals with time tag	M_IT_TA_1
	<17> := Event of protection equipment with time tag	M_EP_TA_1
	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1
	<20> := Packed single-point information with status change detection	M_PS_NA_1
X	<21> := Measured value, normalized value without quality descriptor	M_ME_ND_1
X	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
X	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1

X	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
X	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
X	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
X	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
X	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
X	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1
	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Process Information in Control Direction

X	<45> := Single command	C_SC_NA_1
X	<46> := Double command	C_DC_NA_1
X	<47> := Regulating step command	C_RC_NA_1
X	<48> := Set point command, normalized value	C_SE_NA_1
X	<49> := Set point command, scaled value	C_SE_NB_1
X	<50> := Set point command, short floating point value	C_SE_NC_1
X	<51> := Bitstring of 32 bits	C_BO_NA_1

System Information in Monitor Direction

X	<70> := End of initialization	M_EI_NA_1

System Information in Control Direction

X	<100> := Interrogation command	C_IC_NA_1
X	<101> := Counter-interrogation command	C_CI_NA_1
X	<102> := Read command	C_RD_NA_1
X	<103> := Clock synchronization command	C_CS_NA_1
X	<104> := Test command	C_TS_NA_1
X	<105> := Reset process command	C_RP_NA_1
X	<106> := Delay acquisition command	C_CD_NA_1

Parameter in Control Direction

	<110> := Parameter of measured value, normalized value	P_ME_NA_1
	<111> := Parameter of measured value, scaled value	P_ME_NB_1
	<112> := Parameter of measured value, short floating point value	P_ME_NC_1
	<113> := Parameter activation	P_AC_NA_1

File Transfer

	<120> := File ready	F_FR_NA_1
	<121> := Section ready	F_SR_NA_1
	<122> := Call directory, select file, call file, call section	F_SC_NA_1
	<123> := Last section, last segment	F_LS_NA_1
	<124> := Ack file, ack section	F_AF_NA_1

	<125> := Segment	F_SG_NA_1
	<126> := Directory {blank or X, only available in monitor (standard) direction}	F_DR_TA_1

Type Identification and Cause of Transmission Assignments

Type Identification		Cause of Transmission																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1	X	X	X		X						X	X		X					
<2>	M_SP_TA_1			X		X						X	X		X					
<3>	M_DP_NA_1	X	X	X		X						X	X		X					
<4>	M_DP_TA_1			X		X						X	X		X					
<5>	M_ST_NA_1	X	X	X		X						X	X		X					
<6>	M_ST_TA_1			X		X						X	X		X					
<7>	M_BO_NA_1	X	X	X		X						X	X		X					
<8>	M_BO_TA_1			X		X						X	X		X					
<9>	M_ME_NA_1	X	X	X		X						X	X		X					
<10>	M_ME_TA_1			X		X						X	X		X					
<11>	M_ME_NB_1	X	X	X		X						X	X		X					
<12>	M_ME_TB_1			X		X						X	X		X					
<13>	M_ME_NC_1	X	X	X		X						X	X		X					
<14>	M_ME_TC_1			X		X						X	X		X					
<15>	M_IT_NA_1			X		X										X				
<16>	M_IT_TA_1			X		X										X				
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			
<19>	M_EP_TC_1																			
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1	X	X	X		X									X					
<30>	M_SP_TB_1			X		X						X	X		X					
<31>	M_DP_TB_1			X		X						X	X		X					
<32>	M_ST_TB_1			X		X						X	X		X					
<33>	M_BO_TB_1			X		X						X	X		X					
<34>	M_ME_TD_1			X		X						X	X		X					
<35>	M_ME_TE_1			X		X						X	X		X					
<36>	M_ME_TF_1			X		X						X	X		X					
<37>	M_IT_TB_1			X		X										X				
<38>	M_EP_TD_1																			

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X	X	X							X	X	X
<46>	C_DC_NA_1						X	X	X	X	X							X	X	X
<47>	C_RC_NA_1						X	X	X	X	X							X	X	X
<48>	C_SE_NA_1						X	X	X	X	X							X	X	X
<49>	C_SE_NB_1						X	X	X	X	X							X	X	X
<50>	C_SE_NC_1						X	X	X	X	X							X	X	X
<51>	C_BO_NA_1						X	X	X	X	X							X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X			X							X	X	X
<101>	C_CI_NA_1						X	X			X							X	X	X
<102>	C_RD_NA_1					X												X	X	X
<103>	C_CS_NA_1			X			X	X										X	X	X
<104>	C_TS_NA_1						X	X										X	X	X
<105>	C_RP_NA_1						X	X										X	X	X
<106>	C_CD_NA_1			X			X	X										X	X	X
<110>	P_ME_NA_1						X	X										X	X	X
<111>	P_ME_NB_1						X	X										X	X	X
<112>	P_ME_NC_1						X	X										X	X	X
<113>	P_AC_NA_1						X	X										X	X	X
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1																			

Basic Application Functions

Station Initialization

X	Remote initialization

Cyclic Data Transmission

X	Cyclic data transmission

Read Procedure

X	Read procedure

Spontaneous Transmission

X	Spontaneous transmission

Double Transmission of Information Objects (Cause of Transmission Equals Spontaneous)

The following type identifications can be transmitted in succession as the result of a single status change for an information object. A project-specific list defines the particular information object addresses for which double transmission is enabled.

	Double-point information M_DP_NA_1, M_DP_TA_1, and M_DP_TB_1
	Step position information M_ST_NA_1, M_ST_TA_1, and M_ST_TB_1
	Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1, and M_BO_TB_1
	Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1, and M_ME_TD_1
	Measured value, scaled value M_ME_NB_1, M_ME_TB_1, and M_ME_TE_1
	Measured value, short floating point number M_ME_NC_1, M_ME_TC_1, and M_ME_TF_1

Station Interrogation

X	Global				
X	Group 1	X	Group 7	X	Group 13
X	Group 2	X	Group 8	X	Group 14
X	Group 3	X	Group 9	X	Group 15
X	Group 4	X	Group 10	X	Group 16
X	Group 5	X	Group 11		
X	Group 6	X	Group 12		

Clock Synchronization

X	Clock synchronization
X	Day of week used
X	RES1, GEN (time tag substituted/not substituted) used
X	SU-bit (summertime) used

Command Transmission

X	Direct command transmission
X	Direct set point command transmission
X	Select and execute command
X	Select and execute set point command
X	C_SE ACTTERM used
X	No additional definition
X	Short-pulse duration (duration determined by a system parameter in the outstation)
X	Long-pulse duration (duration determined by a system parameter in the outstation)
X	Persistent output

Transmission of Integrated Totals

X	Mode A: Local freeze with spontaneous transmission
X	Mode B: Local freeze with counter interrogation
X	Mode C: Freeze and transmit by counter-interrogation commands
X	Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously
X	Counter read
X	Counter freeze without reset
X	Counter freeze with reset
X	Counter reset
X	General request
X	Request Counter Group 1
X	Request Counter Group 2
X	Request Counter Group 3
X	Request Counter Group 4

Parameter Loading

X	Threshold value
	Smoothing factor
	Low limit for transmission of measured values
	High limit for transmission of measured values

Parameter Activation

X	Act/deact of persistent cyclic or periodic transmission of the addressed object

Test Procedure

X	Test procedure

File Transfer

File transfer in monitor direction

	Transparent file
	Transmission of disturbance data of protection equipment
	Transmission of sequences of events
	Transmission of sequences of recorded analog values

File transfer in control direction

	Transparent file

Background Scan

X	Background scan

Acquisition of Transmission Delay

X	Acquisition of transmission delay

Interoperability Statement for IEC 60870-5-104 Server for the SEL RTAC

Introduction

This document specifies the capabilities and features of the SEL-implemented IEC 60870-5-104 Server.

This document applies specifically to the following RTAC models and firmware versions.

Model	Firmware Version
SEL-3505 RTAC	SEL-3505-R136-V0-Z001001-D20160624
SEL-3505-3 RTAC	SEL-3505-3-R136-V0-Z001001-D20160624
SEL-3530 RTAC	SEL-3530-R136-V0-Z001001-D20160624
SEL-3530-4 RTAC	SEL-3530-4-R136-V0-Z001001-D20160624
SEL-2241 RTAC	SEL-2241-R136-V0-Z001001-20160624
SEL-3532 RTAC	SEL-3532-N-R136-V0-Z001001-D20160624
SEL-3555 RTAC	SEL-3555-R136-V0-Z001001-D20160624

Interoperability

System or Device

	System definition
	Controlling station definition (master)
X	Controlled station definition (slave)

Network Configuration (Not Applicable)

	Point-to-point		Multipoint-partyline
	Multiple point-to-point		Multipoint-start

Physical Layer (Not Applicable)

Link Layer (Not Applicable)

Application Layer

Transmission Mode for Application Data

Mode 1 (least significant octet first), according to Section 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common Address of ASDU

X	One octet	X	Two octets

Information Object Addresses

X	One octet		Structured
X	Two octets	X	Unstructured
X	Three octets		

Cause of Transmission

X	One octet	X	Two octets (with originator address). Set the originator address to zero if it is not used.

Length of APDU

253	Maximum length of APDU per system in control direction
253	Maximum length of APDU per system in monitor direction

Selection of Standard ASDUs

Process Information in Monitor Direction

X	<1> := Single-point information	M_SP_NA_1
X	<2> := Single-point information with time tag	M_SP_TA_1
X	<3> := Double-point information	M_DP_NA_1
X	<4> := Double-point information with time tag	M_DP_TA_1

X	<5> := Step position information	M_ST_NA_1
X	<6> := Step position information with time tag	M_ST_TA_1
X	<7> := Bitstring of 32 bits	M_BO_NA_1
X	<8> := Bitstring of 32 bits with time tag	M_BO_TA_1
X	<9> := Measured value, normalized value	M_ME_NA_1
X	<10> := Measured value, normalized value with time tag	M_ME_TA_1
X	<11> := Measured value, scaled value	M_ME_NB_1
X	<12> := Measured value, scaled value with time tag	M_ME_TB_1
X	<13> := Measured value, short floating point value	M_ME_NC_1
X	<14> := Measured value, short floating point value with time tag	M_ME_TC_1
X	<15> := Integrated totals	M_IT_NA_1
X	<16> := Integrated totals with time tag	M_IT_TA_1
	<17> := Event of protection equipment with time tag	M_EP_TA_1
	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1
	<20> := Packed single-point information with status change detection	M_PS_NA_1
X	<21> := Measured value, normalized value without quality descriptor	M_ME_ND_1
X	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
X	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
X	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
X	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
X	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
X	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
X	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
X	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1
	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Process Information in Control Direction

X	<45> := Single command	C_SC_NA_1
X	<46> := Double command	C_DC_NA_1
X	<47> := Regulating step command	C_RC_NA_1
X	<48> := Set point command, normalized value	C_SE_NA_1
X	<49> := Set point command, scaled value	C_SE_NB_1
X	<50> := Set point command, short floating point value	C_SE_NC_1
X	<51> := Bitstring of 32 bits	C_BO_NA_1
X	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
X	<59> := Double command with time tag CP56Time2a	C_DC_TA_1

X	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
X	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
X	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
X	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
X	<64> := Bitstring of 32 bits with time tag CP56Time2a	C_BO_TA_1

System Information in Monitor Direction

X	<70> := End of initialization	M_EI_NA_1

System Information in Control Direction

X	<100> := Interrogation command	C_IC_NA_1
X	<101> := Counter-interrogation command	C_CI_NA_1
X	<102> := Read command	C_RD_NA_1
X	<103> := Clock synchronization command	C_CS_NA_1
X	<104> := Test command	C_TS_NA_1
X	<105> := Reset process command	C_RP_NA_1
X	<106> := Delay acquisition command	C_CD_NA_1
X	<107> := Test command with time tag CP56Time2a	C_TS_TA_1

Parameter in Control Direction

X	<110> := Parameter of measured value, normalized value	P_ME_NA_1
X	<111> := Parameter of measured value, scaled value	P_ME_NB_1
X	<112> := Parameter of measured value, short floating point value	P_ME_NC_1
X	<113> := Parameter activation	P_AC_NA_1

File Transfer

	<120> := File ready	F_FR_NA_1
	<121> := Section ready	F_SR_NA_1
	<122> := Call directory, select file, call file, call section	F_SC_NA_1
	<123> := Last section, last segment	F_LS_NA_1
	<124> := Ack file, ack section	F_AF_NA_1
	<125> := Segment	F_SG_NA_1
	<126> := Directory {blank or X, only available in monitor (standard) direction}	F_DR_TA_1
	<127> := Query Log – Request archive file	F_SC_NB_1

Type Identification and Cause of Transmission Assignments

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1	X	X	X		X						X	X		X					
<2>	M_SP_TA_1			X		X						X	X		X					
<3>	M_DP_NA_1	X	X	X		X						X	X		X					

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<4>	M_DP_TA_1			X		X						X	X		X					
<5>	M_ST_NA_1	X	X	X		X						X	X		X					
<6>	M_ST_TA_1			X		X						X	X		X					
<7>	M_BO_NA_1	X	X	X		X						X	X		X					
<8>	M_BO_TA_1			X		X						X	X		X					
<9>	M_ME_NA_1	X	X	X		X						X	X		X					
<10>	M_ME_TA_1			X		X						X	X		X					
<11>	M_ME_NB_1	X	X	X		X						X	X		X					
<12>	M_ME_TB_1			X		X						X	X		X					
<13>	M_ME_NC_1	X	X	X		X						X	X		X					
<14>	M_ME_TC_1			X		X						X	X							
<15>	M_IT_NA_1			X		X										X				
<16>	M_IT_TA_1			X		X										X				
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			
<19>	M_EP_TC_1																			
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1	X	X	X		X									X					
<30>	M_SP_TB_1			X		X						X	X		X					
<31>	M_DP_TB_1			X		X						X	X		X					
<32>	M_ST_TB_1			X		X						X	X		X					
<33>	M_BO_TB_1			X		X						X	X		X					
<34>	M_ME_TD_1			X		X						X	X		X					
<35>	M_ME_TE_1			X		X						X	X		X					
<36>	M_ME_TF_1			X		X						X	X		X					
<37>	M_IT_TB_1			X		X										X				
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X	X	X							X	X	X
<46>	C_DC_NA_1						X	X	X	X	X							X	X	X
<47>	C_RC_NA_1						X	X	X	X	X							X	X	X
<48>	C_SE_NA_1						X	X	X	X	X							X	X	X
<49>	C_SE_NB_1						X	X	X	X	X							X	X	X

Type Identification		Cause of Transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<50>	C_SE_NC_1						X	X	X	X	X							X	X	X
<51>	C_BO_NA_1						X	X	X	X	X							X	X	X
<58>	C_SC_TA_1						X	X	X	X	X							X	X	X
<59>	C_DC_TA_1						X	X	X	X	X							X	X	X
<60>	C_RC_TA_1						X	X	X	X	X							X	X	X
<61>	C_SE_TA_1						X	X	X	X	X							X	X	X
<62>	C_SE_TB_1						X	X	X	X	X							X	X	X
<63>	C_SE_TC_1						X	X	X	X	X							X	X	X
<64>	C_BO_TA_1						X	X	X	X	X							X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X			X							X	X	X
<101>	C_CI_NA_1						X	X			X							X	X	X
<102>	C_RD_NA_1					X												X	X	X
<103>	C_CS_NA_1			X			X	X										X	X	X
<104>	C_TS_NA_1						X	X										X	X	X
<105>	C_RP_NA_1						X	X										X	X	X
<106>	C_CD_NA_1																			
<107>	C_TS_TA_1						X	X										X	X	X
<110>	P_ME_NA_1						X	X										X	X	X
<111>	P_ME_NB_1						X	X										X	X	X
<112>	P_ME_NC_1						X	X										X	X	X
<113>	P_AC_NA_1						X	X										X	X	X
<120>	F_FR_NA_1																X			
<121>	F_SR_NA_1																X			
<122>	F_SC_NA_1																X			
<123>	F_LS_NA_1																X			
<124>	F_AF_NA_1																X			
<125>	F_SG_NA_1																X			
<126>	F_DR_TA_1																X			
<127>	F_SC_NB_1																X			

Basic Application Functions

Station Initialization

X	Remote initialization

Cyclic Data Transmission

X	Cyclic data transmission

Read Procedure

X	Read procedure

Spontaneous Transmission

X	Spontaneous transmission

Double Transmission of Information Objects (Cause of Transmission Equals Spontaneous)

The following type identifications can be transmitted in succession as the result of a single status change for an information object. A project-specific list defines the particular information object addresses for which double transmission is enabled.

	Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1, and M_PS_NA_1
	Double-point information M_DP_NA_1, M_DP_TA_1, and M_DP_TB_1
	Step position information M_ST_NA_1, M_ST_TA_1, and M_ST_TB_1
	Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1, and M_BO_TB_1
	Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1, and M_ME_TD_1
	Measured value, scaled value M_ME_NB_1, M_ME_TB_1, and M_ME_TE_1
	Measured value, short floating point number M_ME_NC_1, M_ME_TC_1, and M_ME_TF_1

Station Interrogation

X	Global				
X	Group 1	X	Group 7	X	Group 13
X	Group 2	X	Group 8	X	Group 14
X	Group 3	X	Group 9	X	Group 15
X	Group 4	X	Group 10	X	Group 16
X	Group 5	X	Group 11		
X	Group 6	X	Group 12		

Clock Synchronization

X	Clock synchronization
X	Day of week used
X	RES1, GEN (time tag substituted/not substituted) used
X	SU-bit (summertime) used

Command Transmission

X	Direct command transmission
X	Direct set point command transmission
X	Select and execute command
X	Select and execute set point command
X	C_SE ACTTERM used
X	No additional definition
X	Short-pulse duration (duration determined by a system parameter in the outstation)
X	Long-pulse duration (duration determined by a system parameter in the outstation)
X	Persistent output
X	Supervision of maximum delay in command direction of commands and set point commands
65535 ms	Maximum allowable delay of commands and set point commands

Transmission of Integrated Totals

X	Mode A: Local freeze with spontaneous transmission
X	Mode B: Local freeze with counter interrogation
X	Mode C: Freeze and transmit by counter-interrogation commands
X	Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously
X	Counter read
X	Counter freeze without reset
X	Counter freeze with reset
X	Counter reset
X	General request
X	Request Counter Group 1
X	Request Counter Group 2
X	Request Counter Group 3
X	Request Counter Group 4

Parameter Loading

X	Threshold value
	Smoothing factor
	Low limit for transmission of measured values
	High limit for transmission of measured values

Parameter Activation

X	Act/deact of persistent cyclic or periodic transmission of the addressed object

Test Procedure

X	Test procedure

File Transfer

File transfer in monitor direction

	Transparent file
	Transmission of disturbance data of protection equipment
	Transmission of sequences of events
	Transmission of sequences of recorded analog values

File transfer in control direction

	Transparent file

Background Scan

X	Background scan

Acquisition of Transmission Delay

	Acquisition of transmission delay

Definition of Time-Outs

Parameter	Default Value	Remarks	Selected Value
t_0	30 s	Time-out of connection establishment	N/A
t_1	15 s	Time-out of send or test APDUs	1–255 s
t_2	10 s	Time-out for acknowledgments in case of no data messages $t_2 < t_1$	1–255 s
t_3	20 s	Time-out for sending test frames in case of a long idle state	Disable, 1–172800 s

Maximum Number of Outstanding I Format APDUs k and Latest Acknowledge APDUs (w)

Parameter	Default Value	Remarks	Selected Value
k	12 APDUs	Maximum difference receive sequence number to send state variable	1–32767
w	8 APDUs	Latest acknowledgment(s) after receiving *w* I format APDUs	1–32767

Portnumber

Parameter	Value	Remarks
Portnumber	2404	23, 1024–65534

Redundant Connections

2	Number N of redundancy group connections used

RFC 2200 Suite

RFC 2200 is an official Internet standard that describes the state of Internet protocol standardization as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

X	Ethernet 802.3
	Serial X.21 interface
	Other selection from RFC 2200:
	List of valid documents from RFC 2200
	1. ..
	2. ..
	3. ..
	4. ..
	5. ..
	6. ..
	7. etc.

IEC 60870-5-103

IEC 60870-5 is an IEC standard that defines systems used for supervisory control and data acquisition (SCADA), including details related to communications between devices. As part of the standard, IEC 60870-5-103 defines a byte-oriented, asynchronous serial-based protocol for communications associated with electrical power systems. The design is based on either balanced (where either client or server can transmit asynchronously) or unbalanced (server can only transmit when solicited by client) topologies and is suitable for point-to-point or point-to-multipoint configurations.

This section describes the configuration and use of the IEC 60870-5-103 server with ACSELERATOR RTAC. You can configure as many as 50 concurrent IEC 60870-5-103 clients in a single RTAC project. The IEC 60870-5-103 client supports commands (in the control direction) and transmitting of requests for Application Service DataUnit (ASDU) types and Cause of Transmission (COT) as shown in the interoperability statements for IEC 60870-5-103, which list COTs for expected behavior only. Additional COTs are recognized for errors.

IEC 60870-5-103 Client Configuration

Configure an IEC 60870-5-103 client connection to communicate via serial or Ethernet to IEDs. The RTAC will poll data from these IEDs and store the data it receives in tags. Use the tag processor to map these data to any protocol, logs, user logic, etc.

Give the device connection a unique name and select the connection type, as shown in *Figure 2.188*. Refer to *Client Connection Types on page 31* for a description of each connection type.

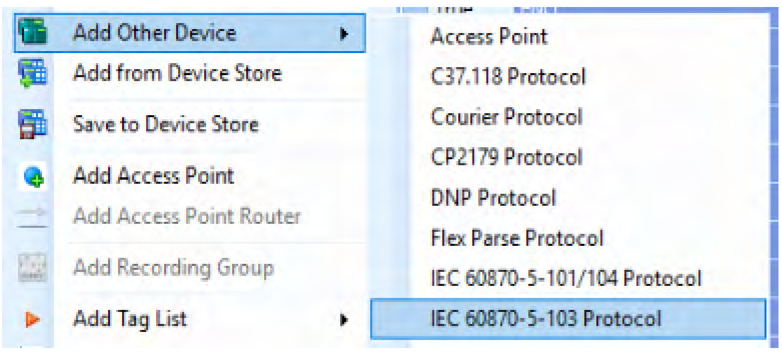


Figure 2.188 Adding IEC 60870-5-10 Client

Settings Tab

The **Settings** tab contains all configurable items for communication. Check the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description. Type any applicable comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Sector Settings Tab

The **Sector Settings** tab contains configuration parameters for a maximum of eight sectors in a server. Each sector has a configurable name and common address. By default, each client has one sector. To add additional sectors, click the + button.

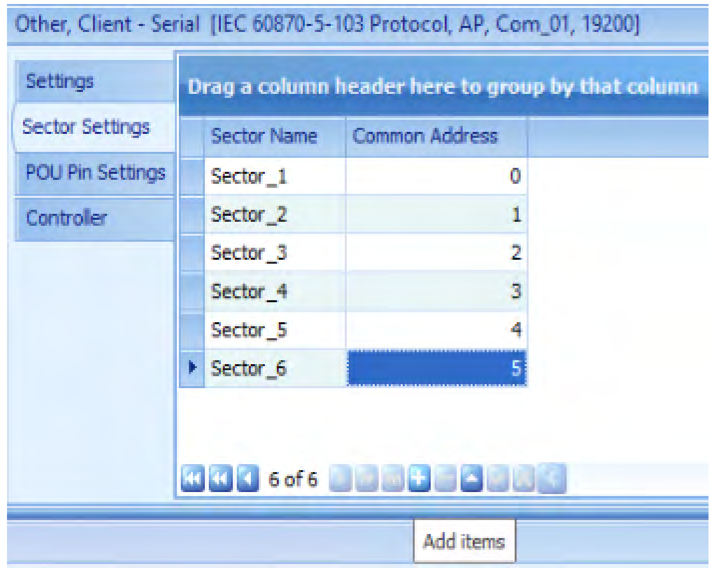


Figure 2.189 IEC 60870-5-103 Sector Settings

IEC 60870-5-103 Client Sector Map

To configure tags for the IEC 60870-5-103 client, select an existing **103 Client Sector Map**. Repeat *Step 1* through *Step 4* to configure each client tag.

Step 1. Click the tab for the appropriate tag type to add and configure tags.

Step 2. Click + to add tags. To optimize system performance, only create as many tags as you need.

Step 3. Rename the tags as necessary.

Step 4. Configure other tag parameters as necessary.

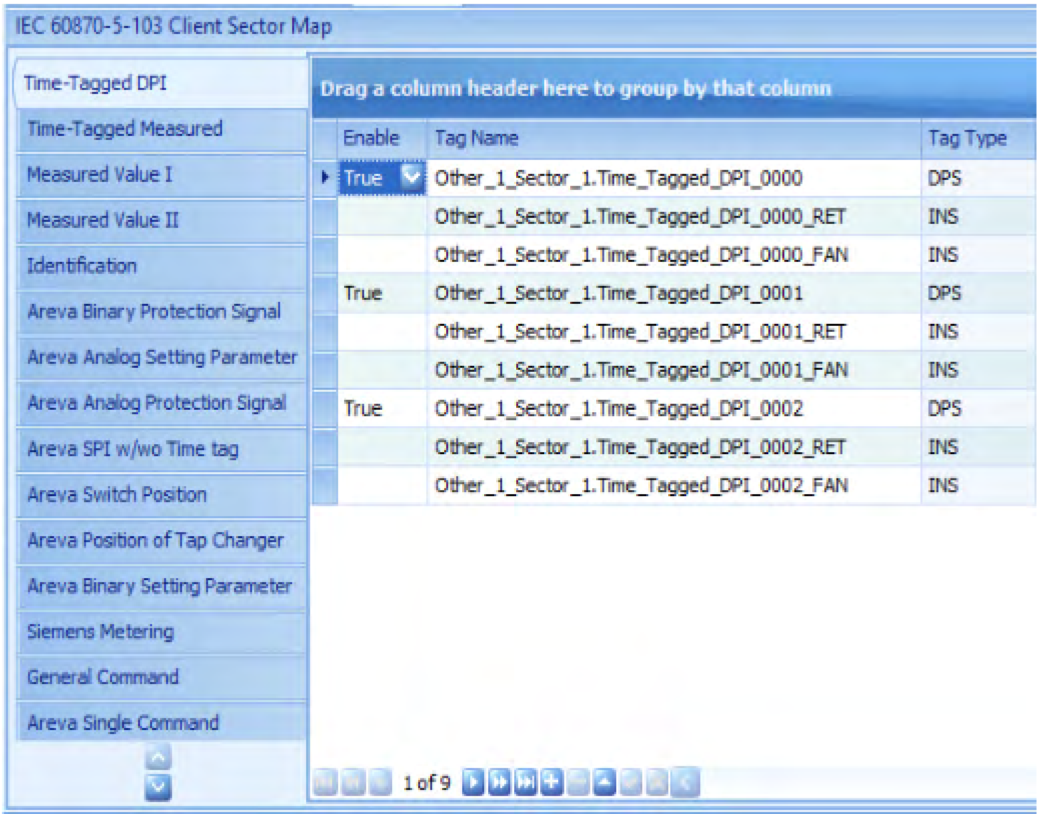


Figure 2.190 IEC 60870-5-103 Client Sector Map

POU Pins

Use POU pin settings to view the present state of the IEC 60870-5-103 client operation and to modify some of the behavior of the protocol. Setting the Visible field to True will cause the POU pin to appear in the **Controller** tab. See *Table 2.112* for the settings descriptions.

Table 2.112 IEC 60870-5-103 Client POU Pin Settings

Pin Name	Pin Type	Description	Default
EN	Input BOOL	The EN input enables or disables this specific function block instance. Other inputs have no effect while EN is False.	True
Disable_Tag_Updates	Input BOOL	The POU will not update or process changes to tags while this input is True.	False
Disable_Controls	Input BOOL	While True, processing of any controllable tags is blocked.	False
Reset_Statistics	Input BOOL	On the rising-edge trigger, all counter POU outputs are reset.	False
Set_ClockFail	Input BOOL	When this pin is set to true this will set the ClockFail bit in the timestamp_t structure of IEC 60870-5-103 tags.	False
ENO	Output BOOL	Indicates that this specific function block instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False

Pin Name	Pin Type	Description	Default
Offline	Output BOOL	This output is False when protocol communications are in process. **Note**: Protocol communications may be in process, but no successful data are exchanged because of settings configuration issues.	False
Invalid_Function_Block_Input	Output BOOL	True when a function block input currently has an invalid or out-of-range value.	False
Controls_Disabled	Output BOOL	Asserted when the Disable_Controls input is asserted. This indicates that the client will not issue control commands.	False
Message_Sent_Count	Output UDINT	Running sum indicating the number of messages transmitted to the remote device.	False
Message_Received_Count	Output UDINT	Running sum indicating the number of messages received from the remote device.	0
Message_Failure	Output: BOOL	Asserts for a single processor cycle when a message is processed that does not conform to the expected protocol format. This may be because of an incomplete response or corrupted data in the message response.	False
Message_Failure_Count	Output UDINT	Running sum indicating the number of messages that have failed.	0
Message_Success_Count	Output UDINT	Running sum indicating the number of messages successively sent and received communicating with the remote device.	0
Direct_Transparent_Connection	Output BOOL	Indicates that the communications have been interrupted by a direct transparent connection through an access point router.	False
Data_Link_Timeout_Count	Output UDINT	Number of data link failures.	0
Response_Timeout_Count	Output UDINT	Number of application layer message response time-outs.	0
Reinit_In_Progress	Output: BOOL	While asserted, indicates that the client is performing initial communications queries prior to data processing	False
Buffer_Overflow	Output BOOL	Asserts for a single processing interval if any message operations were discarded because of queuing limits. A message will be discarded if multiple occurrences of the same operation have been queued prior to the completion of that message type.	False
Buffer_Overflow_Count	Output UDINT	Count indicating the number of times that the Buffer_Overflow pin has been asserted.	0
Restart	Output BOOL	Asserts for a single processing interval when the client receives an end of initialization message with a cause of initialization, indicating a remote start of the server.	False
Event_Collection_Count	Output UDINT	Number of events collected since the POU was enabled.	0
Event_Collection_Enabled	Output BOOL	When True, the RTAC is able and ready to collect events from server.	False

Pin Name	Pin Type	Description	Default
Event_Collection_Pending	Output BOOL	Asserted while there are uncollected events. Deasserts once all events have been collected.	False
Event_Collection_Stored	Output BOOL	Asserts for a processing interval when an event has been stored in nonvolatile memory.	False

Interoperability Statement for IEC 60870-5-103 Client for the SEL RTAC

Introduction

This document specifies the capabilities and features of the SEL-implemented IEC 60870-5-103 Client.

This document applies specifically to the following RTAC models and firmware versions.

Model	Firmware Version
SEL-3505 RTAC	SEL-3505-R145-V0-Z000001-D20190830
SEL-3505-3 RTAC	SEL-3505-3-R145-V0-Z000001-D20190830
SEL-3530 RTAC	SEL-3530-R145-V0-Z000001-D20190830
SEL-3530-4 RTAC	SEL-3530-4-R145-V0-Z000001-D20190830
SEL-2241 RTAC	SEL-2241-R145-V0-Z000001-D20190830
SEL-3532 RTAC	SEL-3532-N-R145-V0-Z000001-D20190830
SEL-3555 RTAC	SEL-3555-R145-V0-Z000001-D20190830

Interoperability

Physical Layer

Electrical Interface

X	EIA-485
X	EIA-232
X	Number of loads is 32 for one piece of protection equipment

NOTE
The EIA-485 standard defines unit loads such that 32 can be operated on one line. For detailed information, refer to clause 3 of the EIA-485 standard.

Optical Interface

	Glass fiber
X	Plastic fiber
	F-SMA type connector
X	BFOC/2,5 type connector

Transmission Speed

X	200 bit/s
X	300 bit/s
X	600 bit/s
X	1200 bit/s
X	2400 bit/s
X	4800 bit/s
X	9600 bit/s
X	19200 bit/s
X	38400 bit/s
X	57600 bit/s
X	115200 bit/s

Link Layer

There are no choices for the link layer.

Application Layer

Transmission Mode for Application Data

Mode 1 (least significant octet first), as defined in 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common Address of ASDU

X	One COMMON ADDRESS of ASDU
	More than one COMMON ADDRESS of ASDU

Selection of Standard Information Numbers in Monitor Direction

	INF	Semantics
X	<0>	End of general interrogation
X	<0>	Time synchronization
X	<2>	Reset FCB
X	<3>	Reset CU
X	<4>	Start/restart
X	<5>	Power on

Status Indication in Monitor Direction

	INF	Semantics
X	<16>	Auto-recloser
X	<17>	Teleprotection active
X	<18>	Protection active
X	<19>	LED reset
X	<20>	Monitor direction blocked
X	<21>	Test mode
X	<22>	Local parameter setting
X	<23>	Characteristic 1
X	<24>	Characteristic 2
X	<25>	Characteristic 3
X	<26>	Characteristic 4
X	<27>	Auxiliary input 1
X	<28>	Auxiliary input 2
X	<29>	Auxiliary input 3
X	<30>	Auxiliary input 4

Supervision Indication in Monitor Direction

	INF	Semantics
X	<32>	Measurand supervision I
X	<33>	Measurand supervision V
X	<35>	Phase sequence supervision
X	<36>	Trip circuit supervision
X	<37>	I»backup operation
X	<38>	VT fuse failure
X	<39>	Teleprotection disturbed
X	<46>	Group warning
X	<47>	Group alarm

Earth Fault Indication in Monitor Direction

	INF	Semantics
X	<48>	Earth fault L1
X	<49>	Earth fault L2
X	<50>	Earth fault L3

	INF	Semantics
X	<51>	Earth fault forward, i.e., line
X	<52>	Earth fault reverse, i.e., busbar

Fault Indication in Monitor Direction

	INF	Semantics
X	<64>	Start/pick-up L1
X	<65>	Start/pick-up L2
X	<66>	Start/pick-up L3
X	<67>	Start/pick-up N
X	<68>	General trip
X	<69>	Trip L1
X	<70>	Trip L2
X	<71>	Trip L3
X	<72>	Trip I» (backup operation)
X	<73>	Fault location X in ohms
X	<74>	Fault forward/line
X	<75>	Fault reverse/busbar
X	<76>	Teleprotection signal transmitted
X	<77>	Teleprotection signal received
X	<78>	Zone 1
X	<79>	Zone 2
X	<80>	Zone 3
X	<81>	Zone 4
X	<82>	Zone 5
X	<83>	Zone 6
X	<84>	General start/pick-up
X	<85>	Breaker failure
X	<86>	Trip measuring system L1
X	<87>	Trip measuring system L2
X	<88>	Trip measuring system L3
X	<89>	Trip measuring system E
X	<90>	Trip I>
X	<91>	Trip I»
X	<92>	Trip IN>
X	<93>	Trip IN»

Auto-Reclosure Indication in Monitor Direction

	INF	Semantics
X	<128>	CB on by AR
X	<129>	CB on by long time AR
X	<130>	AR blocked

Measurands in Monitor Direction

	INF	Semantics
X	<144>	Measurands I
X	<145>	Measurands I, V
X	<146>	Measurands I, V, P, Q
X	<147>	Measurands IN, VEN
X	<148>	Measurands IL1,2,3, VL1,2,3, P, Q, f

Generic Functions in Monitor Direction

	INF	Semantics
X	<240>	Read headings of all defined groups
X	<241>	Read values or attributes of all entries of one group
X	<243>	Read directory of a single entry
X	<244>	Read value or attribute of a single entry
X	<245>	End of general interrogation of generic data
X	<249>	Write entry with confirmation
X	<250>	Write entry with execution
X	<251>	Write entry aborted

Selection of Standard Information Numbers in Control Direction

System Functions in Control Direction

	INF	Semantics
X	<0>	Initiation of general interrogation
X	<0>	Time synchronization

General Commands in Control Direction

	INF	Semantics
X	<16>	Auto-recloser on/off
X	<17>	Teleprotection on/off
X	<18>	Protection on/off

	INF	Semantics
X	<19>	LED reset
X	<23>	Activate characteristic 1
X	<24>	Activate characteristic 2
X	<25>	Activate characteristic 3
X	<26>	Activate characteristic 4

Generic Functions in Control Direction

	INF	Semantics
X	<240>	Read headings of all defined groups
X	<241>	Read values or attributes of all entries of one group
X	<243>	Read directory of a single entry
X	<244>	Read value or attribute of a single entry
X	<245>	General interrogation of generic data
X	<248>	Write entry
X	<249>	Write entry with confirmation
X	<250>	Write entry with execution
X	<251>	Write entry aborted

Basic Application Functions

	Test mode
	Blocking of monitor direction
X	Disturbance data
	Generic services
X	Private data

Miscellaneous

Measurands are transmitted with ASDU 3 as well as with ASDU 9. As defined in 7.2.6.8, the maximum MVAL can either be 1,2 or 2,4 times the rated value. No different rating shall be used in ASDU 3 and ASDU 9, i.e., for each measurand, there is only one choice.

measurand	1,2	2,4
Current L1		
Current L2		
Current L3		
Voltage L1-E		
Voltage L2-E		

measurand	1,2	2,4
Voltage L3-E		
Active power P		
Reactive power Q		
Frequency f		
Voltage L1–L2		

Flex Parse Protocol

Flex Parse Messages

Use Flex Parse Messages in an SEL client or in a Flex Parse Protocol client to create custom commands as long as 255 bytes and optionally parse the response messages. You can configure the RTAC to send a Command String to the SEL device, or any other connected device, at a configurable Poll Period interval. You can also configure the Start of Response Parse Expression and other parse expressions to enable the RTAC to determine when a valid response is received and to interpret the returned message and populate user-created tags. You can define parse expressions in SEL type class expressions or Perl 5 regular expressions (regex), both of which are explained later in this section. See *Table 2.113* for a description of Flex Parse Messages settings.

Add a flex parse command by performing the following steps:

Step 1. From the **Insert** ribbon, select **Other** > **Flex Parse Protocol**.

Step 2. On the **Settings** tab, configure necessary communications settings parameters.

NOTE
The process for adding and configuring a flex parse message is the same on an SEL client or in a Flex Parse Protocol client that you have added to the project.

Step 3. Click on **Flex Parse Messages**.

Step 4. Click on **+** to add a message, then click on **Add**.

Step 5. Configure the Command String, which is the message that will be sent to the SEL device.

Step 6. Configure the timing parameters to dictate how often the command should be sent and the time allotted for the command to complete.

Step 7. Configure the Start of Response Expression that the RTAC will compare with the incoming message to ensure it is parsing the correct message.

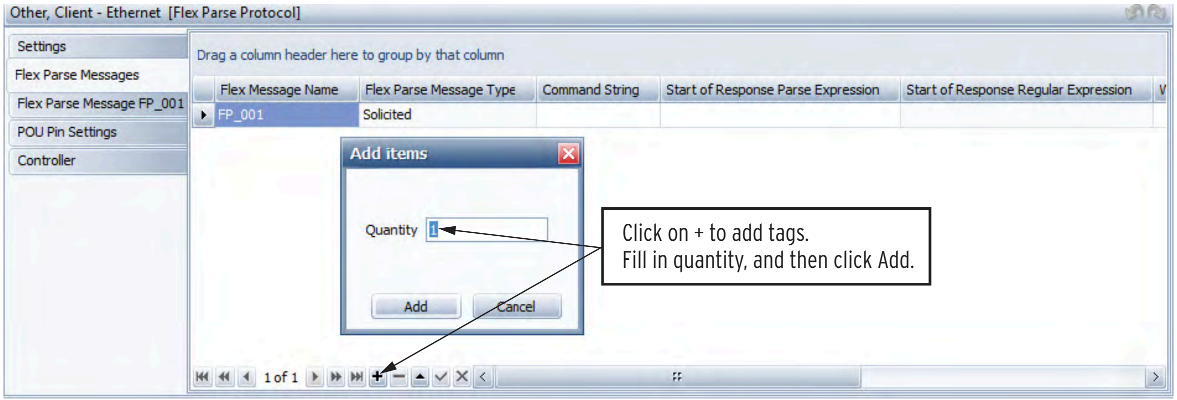


Figure 2.191 Add Flex Parse Message

Table 2.113 Flex Parse Message Settings

Parameter	Description	Default
Flex Message Name	The default message name. You can change this as necessary.	FP–*n*, where *n* is the last flex parse message number + 1
Flex Parse Message Type	The type of message processing.	Solicited
Command String	User-configured message the RTAC will send to the device.	
Start of Response Parse Expression	Optional user-configured expression to locate the start of the response. Enter as Perl 5 regex and/or SEL type class expression.	
Start of Response Regular Expression	The view-only derived Regular Expression from Start of Response Parse Expression.	
Window Size	The maximum number of characters used to satisfy Start_of_Response_Parse_Expression.	512
Poll Period	The period in milliseconds after which the RTAC will send Command String to the configured device.	0
Timeout	The time in milliseconds that the RTAC will wait for a valid reply before aborting message processing (and retrying). If no tags are defined for this message, processing will continue as if all parse messages are satisfied.	5000
Access Level	The access level on the device that the RTAC must attain before sending Command String.	Level_1
Max number of Retries	The maximum number of retries before the IED is marked Offline.	0
Terminate By	Terminate parse search either on completion of the parse or on Response Timeout. Note that if your parse expression has an error, the parse will not complete so the expression will never terminate by parse complete.	On_Timeout
Response Termination Timeout	The time in milliseconds the RTAC will wait after all parse expressions have been evaluated successfully and the line is idle. The timer is reset each time an ASCII character is received.	1000

After you configure a Flex Parse Message, a new tab will appear called **Flex Parse Message FP_001**, where FP_001 is the configured Flex Message Name. In this tab, you can add and configure tags, at your option, to store data from the incoming message. Add tags and configure parse expressions that will search for the data points you want to extract from the incoming message. The RTAC will execute these parse expressions once the Start of Response Expression has been satisfied, which indicates that the RTAC has received a valid message. See *Table 2.114* for a description of Flex Parse tag settings.

To configure the flex parse tags, perform the following steps:

Step 1. Click on the tab **Flex Parse message FP_001** (where FP_001 is the name of the message) in the SEL client or Flex Parse Protocol client.

Step 2. Click on +, type the quantity of tags you need, and press **Add**.

Step 3. Change the tag type and treat as fields, as necessary.

Step 4. Use SEL type class expressions and regex to parse necessary data from the incoming message.

Step 5. Optionally, you can test the flex parse commands you have configured by clicking on the Flex Parse Helper Form.

Step 6. In this blank form, either copy/paste or type in text for which you are building the parse commands. The flex parse helper form highlights the message to show the results of the parsing.

Step 7. Hover your cursor over each data element in the flex parse helper form to see the tag assignment for that data element. See *Figure 2.195*.

Step 8. Adjust the parse commands, if needed, if the data are not assigned to the tags correctly.

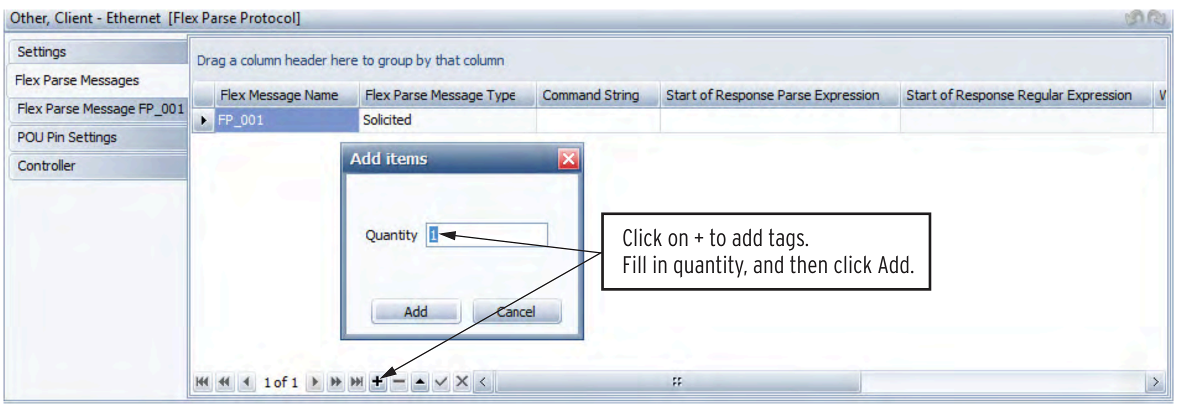


Figure 2.192 Add Flex Parse Tags

Table 2.114 Flex Parse Tags Settings

Parameter	Description	Default
Enable	The processing enable flag. Set this flag to True to enable processing of this tag. Set this flag to False to disable processing.	True
Tag Name	The default tag name contains the device name and tag description. You can change this name as necessary.	
Tag Type	The data type of this tag. See *Data Types on page 747* for more details.	INS
Tag Alias	An optional descriptive tag name. Use this tag alias anywhere in the RTAC system in place of the actual tag name.	
Order ID	The order in which the parse expressions will be executed.	
Parse Expression	The configurable search expression used to identify the piece of data for this tag. Enter as Perl 5 regex and/or SEL type class expression.	
Regular Expression	The view-only derived regular expression from Parse Expression.	
Window Size	The maximum number of characters used to satisfy Parse Expression.	
Treat As	The expected representation or format of integer data.	Decimal

Parameter	Description	Default
Size	The size of data in bytes, once it has been processed. This affects sign extension and range checking.	4
Default Value	The string of characters used if the parsing fails on the captured data. (quality=invalid).	
Comment	Optional user-entered comment field.	

During the operation of sending/receiving and parsing messages, the flex parse message task exhibits the following behavior:

- ➤ If a Command String has been sent and this Command String neither times out nor completes prior to the triggering of another Command String, the issuing of the new message will be delayed and the POU pin Send_Flex_Parse_Message_FP_001_Delay will turn True for one processing interval.
- ➤ After a triggered Command String is sent and the received message parsed successfully, the POU pin Send_Flex_Parse_Message_FP_001_DN will toggle True for one processing cycle.
- ➤ After a parse match is found for one piece of data, the next search group will start after the last character of the previous match group. For example, a search for "little" in the string "Mary had a little lamb" will start with Mary and end when "little" is found. The next search begins immediately after the word "little".
- ➤ If the result of the parse is out of range of the tag definition, the tag value is assigned infinity for MV, zero for all other data types.
- ➤ If the parse fails, the tag value is assigned the default value as configured in the tag setup, and quality is set to invalid.
- ➤ If a parse fails within the window size or the message timeout expires, subsequent parse commands are not executed.

SEL Type Class Exceptions

You can use SEL type class expressions along with regex expressions in your parse expressions. SEL type class expressions are shortcuts that allow you to match on data without writing complex regex expressions. ACSELERATOR RTAC will expand SEL type class expressions to regex expressions each time you save the project.

Each type class expression comes in either a capturing or non-capturing form. Use a capturing type class to extract data into a tag. Use a non-capturing expression to search for something you do not want to put into a tag. For example, searching for the start of the message enables you to know you have the correct message but you do not want to store that string.

Capturing type class expressions have the following format:

```
(?#<TYPE[:param1[,paramN]]>)
```

Non-capturing type class expressions have the following format:

```
(?#<?:TYPE[:param1[,paramN]]>)
```

[ ] represents optional text.

TYPE is the type class name (e.g., INS, REAL, CHAR, etc.).

The value param1 is an optional parameter. For example, (?#<INS: 2>) will limit the number of digits extracted to two. *Table 2.115* describes individual type class expressions, what optional parameters each has, and the default value for the optional parameter if you do not specify one.

Table 2.115 SEL Type Class Expressions

Type Class	Description	Optional Parameters With Defaults	Example
INS	Extracts a group of digits that represent an integer.	Number of digits (1–10), default = 10	0, 4, –500
HINS	Extracts a group of hexadecimal digits that represent an integer.	Number of digits (1–8), default = 8	C1, A1C
HEX_ASCII	Extracts a group of hexadecimal digits written in ASCII that represent an integer, where four hexadecimal digits are a single byte.	Number of digit quartets or bytes (1–4), default = 4	4632 (xF2), 31323334 (x1234)
REAL	Extracts a group of digits that represent a floating point. Note that making digits of precision = 0 may cause unexpected results. The first search of 1.23 will return 1. The second search will find 23, which may not be what you intended.	Number of digits of precision (0–9), default = 3	+5, 35., 7.1, –.87
CHAR	Extracts at most one character.	N/A	a, 7, j
DLIM	Extracts at most one delimiter character.	N/A	<SPC>, <TAB>, <CR>, <LF>, <:>, <;>, </>, <\>, <'>, <">

Example 2.6

In this basic example, the RTAC sends a meter (MET) command to an SEL-221 relay as shown in the following figure. In this example, we will configure the RTAC to parse the returned data into tags.

```
=>>MET <Enter>

Example 230 kV Line                Date: 1/1/1       Time: 03:15:14

             A       B       C      AB      BC      CA
I (A)       994     995     994    1723    1724    1724
V (kV)     134.4   134.3   134.2   233.1   232.8   232.9

P (MW)    401.12
Q (MVAR)    1.00
```

How it works: The configurations in *Figure 2.193* and *Figure 2.194* have the following effects on the MET message shown in *Example 2.6*:

1. The RTAC will log in to the SEL relay in Access Level 1.
2. The RTAC will send the Command String (**MET**, carriage return, line feed) every Poll Period (60,000 ms).
3. The RTAC will look for a string of characters, configured as the Start of Response Parse Expression. In this example, it is **MET**, then any characters until it sees CA. It will only search the number of characters configured as Window Size (512 bytes). A successful match means that the next search will begin directly after the CA in the column headings.

4. In *Figure 2.194*, (?#<INS:4>) tells the RTAC to search through the number of characters specified in Window Size (50) for an INS (integer) that is at least one character and a maximum of four characters long. This will return the value 994 as an integer into the tag SEL_221_16_1_SEL.I_A.stVal.
5. The next search, (?#<INS:4>), will begin to search directly after 994 and will return the value 995 into the second tag and so forth. Note that the REAL number searches include an option (:1) to indicate the number of characters to include past the decimal point.
6. After all searches complete successfully, POU pin Send_Flex_Parse_Message_ Meter_DN toggles True for one processing interval.

Perform the following steps to configure the RTAC to populate tags by parsing the data from the MET command on an SEL-221 Relay:

Step 1. From the **Insert** ribbon, add an SEL-221 relay by clicking on **SEL > 200 Series > 221 Versions > 221-16 > SEL**.

Step 2. Click on the **Flex Parse Messages** tab.

Step 3. Click on **+** and then on **Add** to create one flex parse message.

Step 4. Configure as shown in *Figure 2.193*. Leave fields not shown in *Figure 2.193* at default values.

Step 5. Press **<Ctrl+S>** to save.

Flex Message Name	Command String	Start of Response Parse Expression	Window Size	Poll Period	Timeout	Access Level
Meter	MET<CR><LF>	MET.*\bCA\b	512	60000	5000	Level_1

Figure 2.193 Example 2.6 Flex Parse Command

Step 6. Click on the newly created tab called **Flex Parse Message Meter**.

Step 7. Click on **+**, type in **14**, and then press **Add** to add 14 flex parse tags.

Step 8. Configure as shown in *Figure 2.194*, and press **<Ctrl+S>** to save.

Step 9. Open the flex parse helper form.

Step 10. Copy the **MET** command text from the figure in *Example 2.6*, and paste into the flex parse helper form.

Step 11. Notice the text is highlighted in yellow. Hover your cursor over each data element and notice how the flex parse helper form shows you the tag names and which data are assigned to them. See *Figure 2.195*.

Tag Name	Tag Type	Parse Expression	Window Size	Treat As	Default Value
SEL_221_16_1_SEL.I_A	INS	(?#<INS:4>)	50	Decimal	999
SEL_221_16_1_SEL.I_B	INS	(?#<INS:4>)	25	Decimal	999
SEL_221_16_1_SEL.I_C	INS	(?#<INS:4>)	25	Decimal	999
SEL_221_16_1_SEL.I_AB	INS	(?#<INS:4>)	25	Decimal	9999
SEL_221_16_1_SEL.I_BC	INS	(?#<INS:4>)	25	Decimal	9999
SEL_221_16_1_SEL.I_CA	INS	(?#<INS:4>)	25	Decimal	9999
SEL_221_16_1_SEL.V_A	MV	(?#<REAL:1>)	50	Real	999.9
SEL_221_16_1_SEL.V_B	MV	(?#<REAL:1>)	25	Real	999.9
SEL_221_16_1_SEL.V_C	MV	(?#<REAL:1>)	25	Real	999.9
SEL_221_16_1_SEL.V_AB	MV	(?#<REAL:1>)	25	Real	999.9
SEL_221_16_1_SEL.V_BC	MV	(?#<REAL:1>)	25	Real	999.9
SEL_221_16_1_SEL.V_CA	MV	(?#<REAL:1>)	25	Real	999.9
SEL_221_16_1_SEL.P	MV	(?#<REAL:2>)	50	Real	999.99
SEL_221_16_1_SEL.Q	MV	(?#<REAL:2>)	50	Real	999.99

Figure 2.194 Example 2.6 Flex Parse Tags

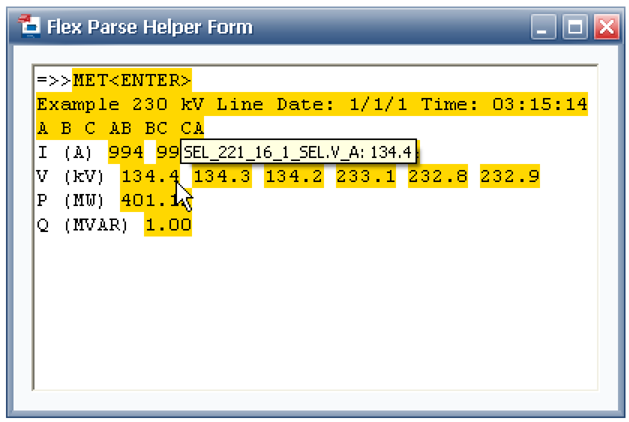


Figure 2.195 Flex Parse Helper Form

Example 2.7

In this example, the RTAC sends a meter m (MET M) command to an SEL-351R relay as shown in the following figure and parses the data into tags. This example is more advanced, because it shows how you can skip fields that may or may not have valid data by using the look-ahead feature. Notice in the following figure that some data fields have the word RESET, which means that there are no data. The associated date fields are blank. In this example, we will add conditions to parse commands such that the parse command will search for a number OR a RESET and will search for valid date data OR look ahead to make sure there is another line to search if the date field is blank.

```
=>>MET M <Enter>
RECLOSER R1 Date: 03/28/01 Time: 10:39:02.389
FEEDER XYZ
        Max     Date            Time            Min     Date        Time
IA(A)   200.0   03/28/01        10:38:56.774    198.0   03/28/01    10:38:57.291
IB(A)   202.1   03/28/01        10:38:56.774    199.0   03/28/01    10:38:58.069
IC(A)   202.2   03/28/01        10:38:57.091    197.0   03/28/01    10:38:56.774
IN(A)   RESET                                   RESET
IG(A)   RESET                                   RESET
VA(kV)  RESET                                   RESET
VB(kV)  RESET                                   RESET
VC(kV)  RESET                                   RESET
VS(kV)  RESET                                   RESET
MW3P    RESET                                   RESET
MVAR3P  RESET                                   RESET
LAST RESET 03/28/01 10:38:56.773
```

How it works: The configurations in *Figure 2.196* and *Figure 2.197* have the following effects on the MET M message shown in *Example 2.7*:

1. The RTAC will log in to the SEL relay in Access Level 1.
2. The RTAC will send the Command String (**MET M**, carriage return, line feed) every Poll Period (60,000 ms).
3. The RTAC will look for a string of characters, configured as the Start of Response Parse Expression. In this example, it is **MET M**, then any characters until it sees IA (A). It will only search the number of characters configured as Window Size (512 bytes). A successful match means that the next search will begin directly after IA (A) in the first data row shown in *Example 2.7*.
4. In *Figure 2.197*, `(?#<REAL:1>)|RESET` tells the RTAC to search through the number of characters specified in Window Size (80) for either the word RESET or for a REAL number with one number to the right of the decimal point. In this example, the value 200.0 is returned as a REAL into the tag SEL_351R_1_SEL.IA_MAX.instMag.
5. The next search, `(?#<INS:2>)|(?=IB)`, will begin searching directly after the value 200.0 for an integer value with a maximum of two characters. If the match fails because the space that should contain a value is blank, it will look ahead for the string IB. If either the number or the string IB is found, the parsing will continue. The tag is populated with the value satisfied by this expression; or, if there is no value, and the look-ahead expression (? = IB) is satisfied, the tag is not populated and the next tag is used for the next configured parse. If the expression is not satisfied, the parse will fail and no more parsing will occur for this data group. In this example, the RTAC finds the month value 03 and places it in SEL_351R_1_SEL.IA_MAX_MONTH.
6. The searches continue to populate DAY, YEAR, HOUR, etc., tags. Each new line (IA, IB, IC, IN, etc.) is tested for the word RESET. If the test for RESET and the look ahead test for the next line were not included, the parsing would fail as soon as it could not find a number to match the parse expression.
7. After all searches are successfully completed, POU pin Send_Flex_Parse_Message_Meter_DN toggles TRUE for one processing interval.

Perform the following steps to configure the RTAC to populate tags by parsing the data from the MET M command on an SEL-351R Relay:

Step 1. From the **Insert** ribbon, add an SEL-351R relay by clicking on **SEL > 300 Series > 351R > SEL**.

Step 2. Click on the **Flex Parse Messages** tab.

Step 3. Click on **+** and then on **Add** to create one flex parse message.

Step 4. Configure as shown in *Figure 2.196*. Leave fields not shown in *Figure 2.196* at default values.

Step 5. Press **<Ctrl+S>** to save.

Flex Message Name	Command String	Start of Response Parse Expression	Window Size	Poll Period	Timeout	Access Level
MET_Min_Max_ALL	MET M<CR><LF>	MET M.*IA\(A\)	512	60000	5000	Level_1

Figure 2.196 Example 2.7 Flex Parse Command

Step 6. Click on the newly created tab called **Flex Parse Message MET_Min_Max_ALL**.

Step 7. Click on **+**, type in **154**, and then press **Add** to add 154 flex parse tags.

Step 8. Configure as shown in *Figure 2.197*, and press **<Ctrl+S>** to save.

Notice the SEL class type ORed with a look ahead for the characters IB `(?#<INS:2>)|(?=IB)` will collect an integer or move on to parsing the next tag if the field is blank by looking ahead to make sure the next line contains IB. In some situations you may not know what is on the next line or it may be an integer value so you cannot look ahead for it. To resolve this, you could combine the SEL class type ORed with a look ahead for an end of line `(?#<INS:2>)|(?=\x0A)`, which will collect an integer; or, if there is not an integer, it will look ahead to see if there is a line feed. As long as there is a line feed, the process will skip to parse the next tag.

Tag Name	Tag Type	Parse Expression	Window Size	Treat As	Size	Default Value
SEL_351R_1_SEL.IA_MAX	MV	(?#<REAL:1>)|RESET	80	Real	4	999.9
SEL_351R_1_SEL.IA_MAX_MONTH	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	99
SEL_351R_1_SEL.IA_MAX_DAY	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	99
SEL_351R_1_SEL.IA_MAX_YEAR	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	99
SEL_351R_1_SEL.IA_MAX_HOUR	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	99
SEL_351R_1_SEL.IA_MAX_MINS	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	99
SEL_351R_1_SEL.IA_MAX_SEC	MV	(?#<REAL>)|(?=IB)	80	Real	4	99.999
SEL_351R_1_SEL.IA_MIN	MV	(?#<REAL:1>)|(?=IB)	80	Real	4	888.8
SEL_351R_1_SEL.IA_MIN_MONTH	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	88
SEL_351R_1_SEL.IA_MIN_DAY	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	88
SEL_351R_1_SEL.IA_MIN_YEAR	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	88
SEL_351R_1_SEL.IA_MIN_HOUR	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	88
SEL_351R_1_SEL.IA_MIN_MINS	INS	(?#<INS:2>)|(?=IB)	80	Decimal	4	88
SEL_351R_1_SEL.IA_MIN_SEC	MV	(?#<REAL>)|IB	80	Real	4	88.888
SEL_351R_1_SEL.IB_MAX	MV	(?#<REAL:1>)|RESET	80	Real	4	999.9
SEL_351R_1_SEL.IB_MAX_MONTH	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	99
SEL_351R_1_SEL.IB_MAX_DAY	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	99
SEL_351R_1_SEL.IB_MAX_YEAR	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	99
SEL_351R_1_SEL.IB_MAX_HOUR	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	99
SEL_351R_1_SEL.IB_MAX_MINS	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	99
SEL_351R_1_SEL.IB_MAX_SEC	MV	(?#<REAL>)|(?=IC)	80	Real	4	99.999
SEL_351R_1_SEL.IB_MIN	MV	(?#<REAL:1>)|(?=IC)	80	Real	4	888.8
SEL_351R_1_SEL.IB_MIN_MONTH	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	88
SEL_351R_1_SEL.IB_MIN_DAY	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	88
SEL_351R_1_SEL.IB_MIN_YEAR	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	88
SEL_351R_1_SEL.IB_MIN_HOUR	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	88
SEL_351R_1_SEL.IB_MIN_MINS	INS	(?#<INS:2>)|(?=IC)	80	Decimal	4	88
SEL_351R_1_SEL.IB_MIN_SEC	MV	(?#<REAL>)|IC	80	Real	4	88.888
SEL_351R_1_SEL.IC_MAX	MV	(?#<REAL:1>)|RESET	80	Real	4	999.9
SEL_351R_1_SEL.IC_MAX_MONTH	INS	(?#<INS:2>)|(?=IN)	80	Decimal	4	99
SEL_351R_1_SEL.IC_MAX_DAY	INS	(?#<INS:2>)|(?=IN)	80	Decimal	4	99
SEL_351R_1_SEL.IC_MAX_YEAR	INS	(?#<INS:2>)|(?=IN)	80	Decimal	4	99
SEL_351R_1_SEL.IC_MAX_HOUR	INS	(?#<INS:2>)|(?=IN)	80	Decimal	4	99

Figure 2.197 Example 2.7 Flex Parse Tags

Example 2.8 Parsing Strings

Although there is not an SEL type class expression for parsing a string value, you can easily do this by using a flex parse expression wrapped in parentheses ( ). The ( ) indicate this is a capturing search so the value is stored in the tag. Anything configured outside of the ( ) is still used for the search but will not be stored in the tag. To configure a flex parse expression to store a string value, set the **Tag Type** to **STR** and the **Treat As** column to **String** for that tag. Use regular expression syntax, such as in *Table 2.116*, to indicate the string that is to be captured. If you wanted to capture each line label from the **MET M** command in *Example 2.7*, you could enter the following for the flex parse expression: **([A–Z, /(,/)]{1,6})**.

The ( ) wrap a capturing expression to indicate the value obtained should be stored in the tag.

The brackets [ ] wrap a list of values to search for. In this example, we are searching for the following:

- Any capitalized letter from A to Z
- Any spaces
- Any ( ) characters. Notice they must be escaped by placing a / before each of them.

The curly brackets { } wrap a min,max value.

The entire message indicates we want a group of characters containing any capitalized letters, spaces, ( ) characters, and we can have a minimum and maximum total length and repetition of any of the defined characters of 1 and 6, respectively.

Regex

A regular expression (regex) is a method for matching characters in a string of text. Regex is similar to, but far more powerful than, a search engine in a word processor, and can be used to match exact strings, ranges, parts of a string, beginning and ending of lines; to exclude certain strings; and much more. The RTAC uses Perl 5-compatible regex in flex parse messaging to extract data from returned message strings. You can use regex in the flex parse message, or you can use SEL type class expressions, and ACSELERATOR RTAC will convert the expression to regex for you.

Each regex attempts to find a single specified match. For example, if you applied the regex 43 to the string 19304-5043920-04399, it would begin searching at 193... and return the first instance of 43. If you applied the regex again, it would begin searching at 920 and return the second instance of 43. The regex search will always begin at the first character after the last successful matched string. The set of characters the regex is going to search is called a character set or, more commonly, a character class. The characters used in the search are referred to as tokens (alpha/numeric characters, spaces, special characters, etc.). Some searches look for places before, after or between characters instead of searching for a character. These are called anchors. For example, $ will match the end of a string before a line break.

Regex is very common in the computing industry, so there are many resources in books and on the Internet that contain detailed regex syntax and theory. *Table 2.116* contains some common regex expressions.

Table 2.116 Common Regular Expression Syntax

Metacharacter	Description	Example
[	Defines a character class. Match a single character out of a list.	[0-9a-fA-F] Matches a single hex character from a string, case insensitive.
\	Meaning one: Escape the next character. Several characters you may want to search for have special meaning in regex. Use the \ to tell regex to not interpret the meaning of the next character.	+ Searches for a "+" instead of interpreting the + as a regex command.
	Meaning two: Combine with certain characters to become shorthand character classes. There are many shorthand classes.	\d Matches a digit, as [0–9]. \b Matches a whole alphanumeric sequence, as \bmyword\b

<table>
<tr><th>Metacharacter</th><th>Description</th><th>Example</th></tr>
<tr><td>^</td><td>Meaning one: A logical NOT of the character class when used in brackets [^].</td><td>q[^x] Matches "q" in "query" but will not match the "q" in Iraq because there are no characters after the latter "q".</td></tr>
<tr><td></td><td>Meaning two: An anchor that matches the position before the first character in a line.</td><td>^J Matches before the "J" in "Jack" and the Beanstalk"</td></tr>
<tr><td>$</td><td>An anchor that matches right after the last character in a line.</td><td>k $ Matches after the "k" in "Jack and the Beanstalk"</td></tr>
<tr><td>.</td><td>Matches any single token.</td><td>ca. Matches cat, cap, car...</td></tr>
<tr><td>|</td><td>Logical OR alternation.</td><td>cat|dog Will match the word cat or the word dog.</td></tr>
<tr><td>?</td><td>Makes the preceding token optional.</td><td>colou?r Matches "color" or "colour".</td></tr>
<tr><td>*</td><td>Matches the preceding token zero or more times. Similar to ? except will match zero or more tokens of the preceding token or group.</td><td>[0–9].*\bdogs\b Matches the string "4 People were walking their dogs".</td></tr>
<tr><td>+</td><td>Matches the preceding token one or more times.</td><td>[1–9]+ Matches 9954 from the string "9954"</td></tr>
<tr><td>( )</td><td>Used for matching a group. All tokens within the ( ) are a group or single token.</td><td>go(fishing)? Matches "go" or "go fishing".</td></tr>
<tr><td>{ }</td><td>Specifies the number of repeats. Also specifies a min/max number of repeats as {min,max}. {2,} indicates there is a minimum of two repeats but no maximum.</td><td>[1–9][0–9]2 Matches 100 through 999 because the second character class repeats twice.</td></tr>
<tr><td>?=</td><td>Look ahead looks forward for a match but does not stop at the look ahead character. In the example, q(?=u) applied to the word "query", the parse returns true because it found a q and also looked ahead and found a u. The next search begins at the "u", not at the "e".</td><td>q(?=u) Matches the "q" in "query" but not in "Iraq" because it looked ahead for a "u".
q(?!u) Matches the "q" in "Iraq" but not in "query".</td></tr>
</table>

eDNA

Overview

Configure an eDNA client Ethernet connection to communicate with an eDNA Universal Service such as data historians or other data archival solutions. The RTAC supports one eDNA client connection. This protocol is supported by the SEL-3555 and SEL-3532-4.

This section describes the configuration and use of the eDNA client connection in ACSELERATOR RTAC.

eDNA Client Configuration

Configure the eDNA client to communicate via Ethernet to an eDNA Universal Service. Create binary and analog tags in the eDNA client **Digitals** and **Analogs** tabs. Although each RTAC project can support only one eDNA client connection, you can provide a primary and failover IP address from the **Settings** tab.

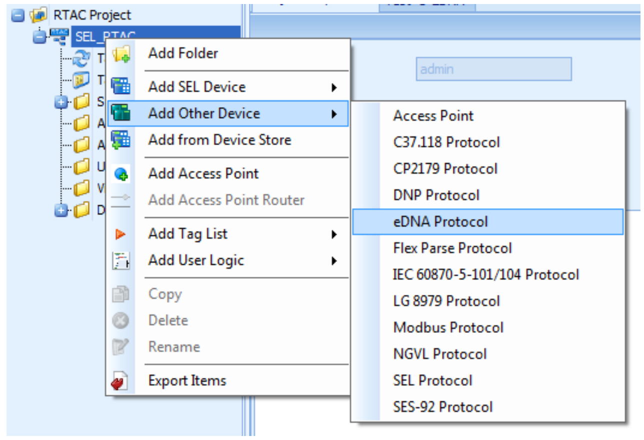


Figure 2.198 Insert eDNA Client Device

Give the device connection a unique name. Note that the only **Connection Type** available is **Client – Ethernet**.

Settings Tab

The **Settings** tab contains all configurable items necessary for communications. Refer to the **Description** column for information about each setting. Move the main window horizontal slider, resize the column width, or hover your mouse over the description cell to view the cell content in its entirety. Type any applicable comments in the blank column to the far right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

Add Client Device Data

Perform the following steps to add client device data.

Step 1. Click a device tag type tab (digitals or analogs) to add and configure tags.

Step 2. Click + to add tags (2000 tag limit each for digitals and analogs).

Creating only the number of necessary tags helps optimize system performance.

Step 3. Change the names of the tags as necessary by using the **Tag Name** column.

Step 4. Change the point ID of each tag (using the **Point ID** column) as necessary to match the settings of the eDNA Universal Service that will be collecting these data from the RTAC.

Step 5. Change other tag-related information as necessary.

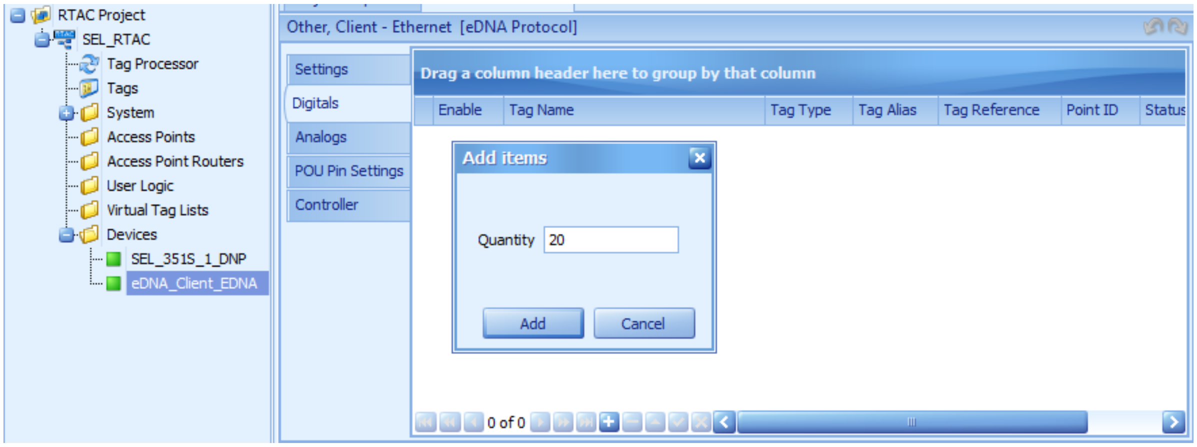


Figure 2.199 Add eDNA Client Tags

Device Tag Type Configuration Parameters

Each device tag type has parameters that must be configured correctly to ensure proper system operation.

Table 2.117 Common Device Tag Type Parameters

Parameter	Description	Default
Enable	Set to True to enable processing, and set to False to disable processing.	True
Tag Name	This parameter describes the tag. You can change this name as necessary.	The default tag name contains the device name, an abbreviation of the tag type, and is numbered 0–*n* tags.
Tag Alias	Enter an optional descriptive tag name into this field, and you can reference this tag alias anywhere in the RTAC project in place of the actual tag name.	
Tag Reference	See *Tag Mapping on page 353* for more details.	
Point ID	A unique name that is used to transmit the tag value to the eDNA Universal Service. The Point ID can have no more than eight characters.	The default Point ID contains an abbreviation of the tag type and is numbered 0–*n* tags.
Comment	User-entered comment field. The first 224 characters of this field will be transmitted to the eDNA Universal Service as the **Extended Description**.	

Table 2.118 Binary Output Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Supported tag types: SPS.	SPS
Status Value	The initialized value of the tag at startup.	False

Table 2.119 Analog Output Parameters

Parameter	Description	Default
Tag Type	See *Data Types on page 747* for more details. Supported tag types: MV, CMV, INS.	MV
Inst Magnitude	The initialized instantaneous value at startup.	0

Parameter	Description	Default
Magnitude	The initialized dead-band value at startup.	0
Dead Band	The number of units of change necessary for magnitude to update to the value of Inst Magnitude (IF |instMag – mag| > db, then mag := instMag).	100
Zero Dead Band	The number of units at or below which a unit is forced to zero (IF |mag| < zeroDB, then mag := 0).	2
Max Value	The maximum value allowed for this point. If instMag > Max Value, then ".q.detailQual.outOfRange" is set.	+9.9999999e+036
H H Limit	The high-high alarm limit. Excursions beyond this limit cause the RTAC to generate a high-high alarm.	+9.999999e+035
H Limit	The high alarm limit. Excursions beyond this limit cause the RTAC to generate a high alarm.	1e+035
L Limit	The low alarm limit. Excursions beyond this limit cause the RTAC to generate a low alarm.	–1e+035
L L Limit	The low-low alarm limit. Excursions beyond this limit cause the RTAC to generate a low-low alarm.	–9.9999999e+035
Min Value	The minimum value allowed for this point (IF instMag < Min Value, then ".q.detailQual.outOfRange" is set).	–9.9999999e+036

Tag Mapping

The eDNA client has a built-in direct-assignment mechanism, so there is no need for the tag processor to map analogs and digitals. Note that this method of direct tag assignment assigns the value of the source tag directly to the value of the destination tag without modification. If you need any conditioning or manipulation of the source tag (such as scaling, offsetting, and logical combination) as part of the assignment, use the tag processor or user logic instead. The following steps detail how to map eDNA tags.

Step 1. Type in the **Tag Reference** column the name of the tag to which you want to map. Note that when you use this tag-mapping method, you cannot use the eDNA tag alias. The **Tag Alias** column will be grayed out to indicate its lack of availability. For more information about the **Tag Reference** column, see *Tag Reference on page 355*.

Step 2. Repeat for each tag in the eDNA client.

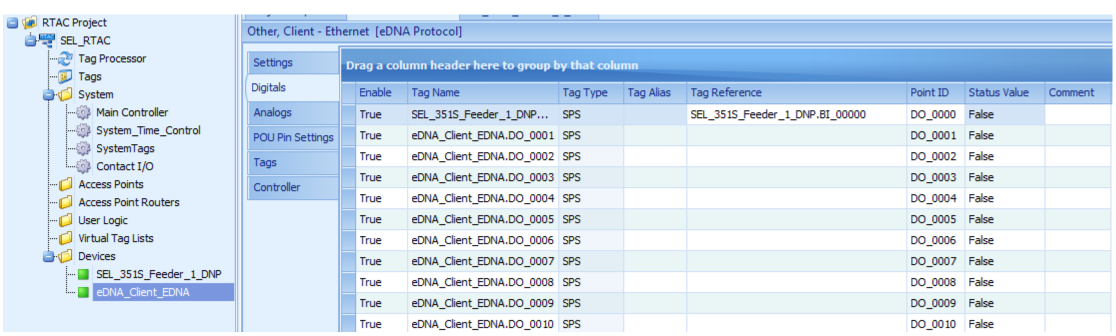


Figure 2.200 Mapping eDNA Client Tags

Table 2.120 POU Pin Settings

Parameter	Description	Default
Alternate_Connection_Status	When zero, the alternate connection is disconnected. When one, the alternate connection is disconnected. When two, the alternate connection is not in use.	0
Buffer_Overflow	Indicates that the RTAC was unable to process all eDNA tag event changes fast enough to avoid event loss. Asserts for one task cycle.	False
Buffer_Overflow_Count	Number of buffer overflows since last RTAC reset or power cycle.	0
Data_Cache_Queue_Count	Messages that have been queued to send to the eDNA Universal Service. Is zero if Data Cache Size is set to zero.	0
Data_Cache_Queue_File_Count	Number of files the data cache is currently using.	0
Data_Cache_Queue_Full	When True, the data cache is full.	False
Data_Cache_Queuing_Data	When True, the data cache is being used to store data.	False
Data_Cache_Send_Queue_Count	Number of messages that have been prepared to send to the eDNA Universal Service. Is zero if **Data Cache Size** is set to zero. Data_Cache_Send_Queue_Count – Data_Cache_Send_Queue_Count is equal to the number of messages that have been sent to the eDNA Universal Service.	0
EN	POU is enabled.	True
ENO	POU is enabled and initialized.	False
Issue_Data_Sync	When True a data synchronization message is being issued.	False
Issue_Data_Sync_Period	Time between data synchronization transmissions. This output is set by the **Issue Data Sync Period** in the **Advanced Settings** on the **Settings** tab.	T#0M
Message_Failure	When data cache size is set to zero, increases by one each time an event occurs while the RTAC is disconnected from the eDNA Universal Service.	False
Message_Failure_Count	Message failure count increases by one every time a message failure occurs.	0
Message_Received_Count	Message received count increases by one every time a message is received from the eDNA Universal Service.	0
Message_Sent_Count	Message sent count increases by one every time a message is sent to the eDNA Universal Service. Only increases if **Data Cache Size** is set to zero. This value will increase regardless of whether the eDNA Universal Service successfully accepts a message.	0
Message_Success_Count	Increases by one every time a message is successfully sent to the eDNA Universal Service. Only increases if **Data Cache Size** is set to zero.	0
NAK_Count	Increases by one when the eDNA Universal Service responds to a transmission with a negative acknowledgment.	0
Offline	Asserts when the RTAC is unable to communicate to the eDNA Universal Service.	True
Primary_Connection_Status	When zero, the primary connection is disconnected. When one, the primary connection is connected. When two, the primary connection is not in use.	0
Using_Primary_Connection	When True, the primary connection is in use. When False, the primary connection is not in use.	False

Tag Reference

Creation of a data assignment through use of the **Tag Reference** column differs from creation of a data assignment through the tag processor or user logic. Instead of copying the value of the tag from the source tag to the destination tag, the software creates a direct link to the source tag in the memory of the RTAC. Because of this, working with reference tags within RTAC settings differs from use of the tag processor or user logic in the following ways:

- ➤ No data manipulation (such as scaling, offsetting, or logical math) can be performed on the data prior to assignment.
- ➤ Once an eDNA tag has had data assigned to it through use of the **Tag Reference** column, it is not accessible from elsewhere in the RTAC settings. Only the tag placed into the **Tag Reference** column will be accessible.

Tags

See *Tags (Overview) on page 38* for a description of the **Tags** tab.

Controller

See *Controller (Advanced Usage) on page 37* for a description of the **Controller** tab.

eDNA Protocol Implementation Notes

This section contains additional details about implementation of the eDNA protocol in the RTAC.

- ➤ The eDNA extended ID will be composed of the first 128 characters of the tag name.
- ➤ The eDNA extended description will be composed of the first 224 characters of the comment field associated with each tag.
- ➤ The eDNA data cache clears for transmission of new settings to the RTAC, the eDNA data cache will clear. There is no clearing of the cache if the RTAC is turned off and then back on.

CDC Type 2

CDC Type 2 Client Configuration

Configure a CDC Type 2 client to communicate via serial to IEDs by clicking the Extensions icon, as shown in *Figure 2.201*. The CDC protocol is a bit-based protocol that is only supported via expansion serial cards for the SEL-3555 and SEL-3560 Real-Time Automation Controllers. The number of supported CDC Type 2 clients in a single RTAC project is restricted to the number of physical serial ports available. Use the Tag Processor to map collected data to server tags.

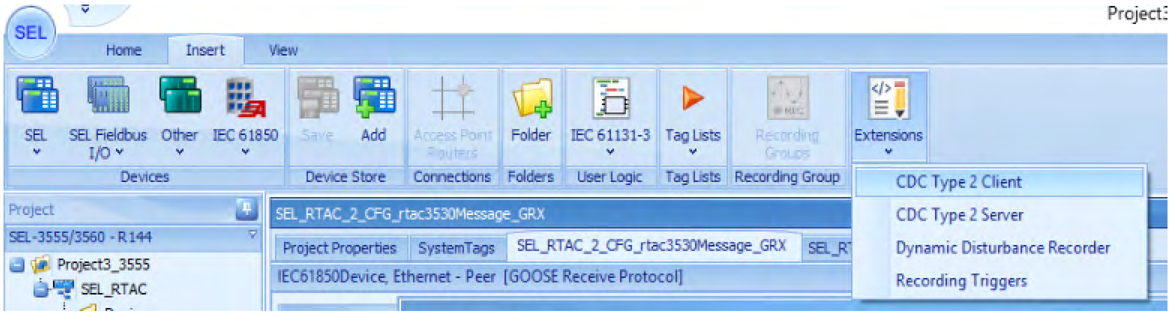


Figure 2.201 Add CDC Type 2 Client

Table 2.121 lists the function codes that the CDC Type 2 client will respond to. All other messages received by the server will not generate a response.

NOTE
Function Code Values are in hexadecimal notation.

Table 2.121 Supported Function Codes

Function Code	Description
00	Scan 1
03	Direct Set Point
05	Control Trip (select)
06	Control Close (select)
07	Set Point (select)
08	Operate
0B	Direct - Trip
0C	Direct - close

Settings Tab

The **Settings** tab contains all configurable items for communications. The **Description** column shows details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description.

Add Client Device Data

Step 1. Click a device tag type tab to add tags.

Step 2. Click + to add tags. Aliases cannot be assigned to tag names in the CDC client.

Step 3. Change other tag-related information as necessary.

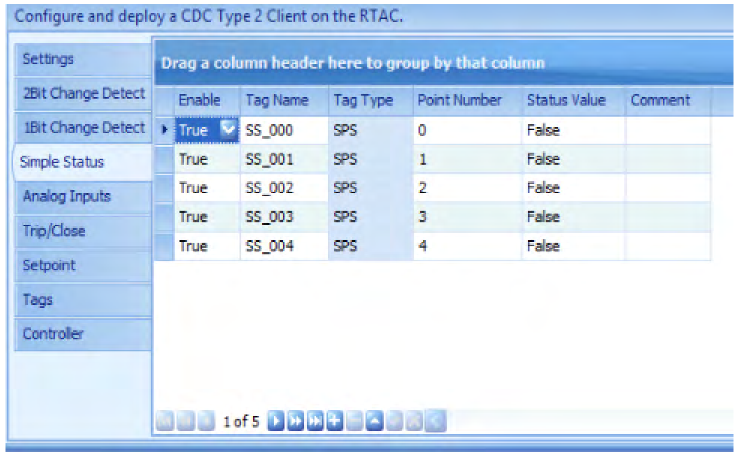


Figure 2.202 Add CDC Type 2 Tags

Repeat these steps to configure all CDC Type 2 client connections. When finished, configure the Tag Processor to populate server connection tags with collected CDC Type 2 client values.

Each tag type per the CDC Type 2 protocol supports the maximum number of tags, as shown in *Table 2.122*.

Table 2.122 CDC Type 2 Tag Types

Tag Type	Supported Tags Per CDC Client
2bit Change Detect	256
1bit Change Detect	256
Simple Status	256
Analog Inputs	128
Trip/Close	64

Controller Pins

Use POU pin settings to view the present state of the CDC Type 2 client operation and to modify behavior of the protocol. See *Table 2.123* for the settings descriptions.

Table 2.123 CDC Tag Type 2 Client Pin Descriptions

Pin Name	Pin Type	Description	Default
EN	Input: BOOL	The EN input enables or disables this specific Function Block instance. Other inputs have no effect while EN is False.	True
Disable_Tag_Updates	Input: BOOL	The POU will not update tags or process changes to tags while this input is True.	False
Disable_Controls	Input: BOOL	If True, the processing of any controllable tags is blocked.	False
Reset_Statistics	Input: BOOL	On rising-edge trigger, all counter POU output are reset.	False
ENO	Output: BOOL	Indicates this specific Function Block Instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False
Offline	Output: BOOL	This output is False when protocol communications are in process. Note: Protocol communications may be in process but no successful data exchange due to settings configuration issue.	False

Pin Name	Pin Type	Description	Default
Controls_Disabled	Output: BOOL	Asserted when Disable_Controls input is asserted. This indicates the client will not issue control commands.	False
Message_Sent_Count	Output: UDINT	Running sum indicating the number of messages transmitted to the remote device.	False
Message_Received_Count	Output: UDINT	Running sum indicating the number of messages received from the remote device.	0
Message_Failure	Output: BOOL	Asserts for a single processor cycle when a message is processed that does not conform to the expected protocol format. This may be because of an incomplete response or corrupted data in the message response.	False
Message_Failure_Count	Output: UDINT	Running sum indicating the number of messages that have successfully been sent or received.	0
Message_Success_Count	Output: UDINT	Running sum indicating the number of messages successively sent and received communicating with the remote device.	0

CDC Type 2 Server Configuration

Configure a CDC Type 2 server to communicate via serial to IEDs by clicking the Extensions icon. The CDC protocol is a bit-based protocol that is only supported on the SEL-3555 expansion serial cards. As many as 10 CDC Type 2 servers are supported in a single RTAC project. Use the Tag Processor to map collected data to CDC Type 2 server tags.

Supported Function Codes

Table 2.124 lists the function codes that the CDC Type 2 server will respond to. All other messages received by the server will not generate a response.

NOTE
Function code values are in hexadecimal notation.

Table 2.124 Supported Function Codes

Function Codes	Description
00	Scan 1
01	Scan 2
02	Scan 3
21–2F	Scan 3X
03	Direct Set Point
04	No Operation
05	Control Trip (select)
06	Control Close (select)
07	Set Point (select)
08	Operate
09	Reset

Function Codes	Description
0A	Accumulator Freeze
0B	Direct - Trip
0C	Direct - Close
0D	Accumulator Reset
13	Memory Read
14	Memory Write
18	SOE Size Setup
19	SOE Event Dump
1A	SOE Point Status Dump
1B	SOE Point Enable/Disable Setup
1C	SOE Point Enable/Disable Dump
1D	SOE Time Sync
1E	SOE Time Retrieval
1F	SOE Time Correction
80	Repeat Scan 1
81	Repeat Scan 2
82	Repeat Scan 3
A1	Repeat Scan 3X

To add a CDC Type 2 server connection, select **CDC Type 2 Server** from the **Custom Applications** menu. Give the device connection a unique name.

Settings Tab

The **Settings** tab contains all configurable items for communications. Check the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description.

Add Server Device Data

The following steps detail the process of adding server device data.

Step 1. Click on a device tag type to add tags.

Step 2. Click + to add tags. Aliases cannot be assigned to tag names in the CDC server.

Step 3. Change other tag-related information as necessary.

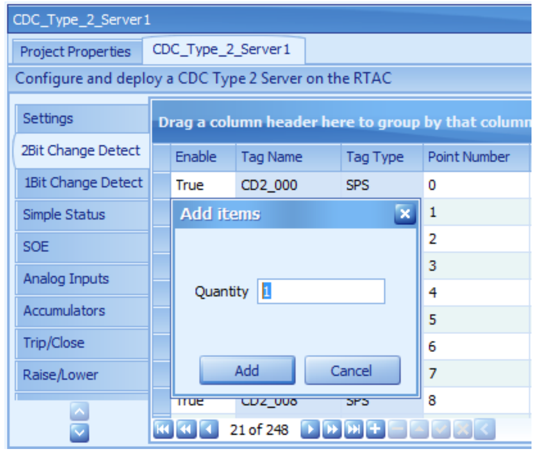


Figure 2.203 Add CDC Type 2 Tags

Repeat these steps to configure all CDC Type 2 server connections. When finished, configure the Tag Processor to populate these server connection tags with actual values.

Each tag type per the CDC Type 2 protocol supports the maximum number of tags.

Table 2.125 CDC Type 2 Tag Types

Tag Type	Supported Tags Per CDC Server
2bit Change Detect	256
1bit Change Detect	256
Simple Status	256
SOE	256
Analog Inputs	128
Accumulators	32
Trip/Close	64
Raise/Lower	8
Setpoint	16

Controller Pins

Use POU pin settings to view the present state of the CDC Type 2 server operation and to modify some of the behavior of the protocol. See *Table 2.126* for the settings descriptions.

Table 2.126 CDC Type 2 Server Pin Descriptions

Pin Name	Pin Type	Description	Default
EN	Input: BOOL	The EN input enables or disables this specific Function Block instance. Other inputs have no effect while EN is False.	True
Disable_Tag_Updates	Input: BOOL	The POU will not update tags or process changes to tags while this input is True.	False

Pin Name	Pin Type	Description	Default
Disable_Controls	Input: BOOL	While True, the processing of any controllable tags is blocked.	False
Reset_Statistics	Input: BOOL	On rising-edge trigger, all counter POU output are reset.	False
ENO	Output: BOOL	Indicates this specific Function Block Instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False
Offline	Output: BOOL	This output is False when protocol communications are in process. Note: Protocol communications may be in process but no successful data exchange due to settings configuration issue.	False
Controls_Disabled	Output: BOOL	Asserted when Disable_Controls input is asserted. This indicates the client will not issue control commands.	False
Message_Sent_Count	Output: UDINT	Running sum indicating the number of messages transmitted to the remote device.	False
Message_Received_Count	Output: UDINT	Running sum indicating the number of messages received from the remote device.	0
Message_Failure	Output: BOOL	Asserts for a single processor cycle when a message is processed that does not conform to the expected protocol format. This may be because of an incomplete response or corrupted data in the message response.	False
Message_Failure_Count	Output: UDINT	Running sum indicating the number of messages that have successfully been sent or received.	0
Message_Success_Count	Output: UDINT	Running sum indicating the number of messages successively sent and received communicating with the remote device.	0
Accumulator_Freeze	Output: BOOL	Asserts for one processing interval when a freeze request is received by the server.	False
SOE_Overflow	Output: BOOL	Asserted for a single processing interval when an SOE event could not be queued.	False
SOE_overflow_Count	Output: UDINT	Running sum indicating the number of messages that could not be queued.	0

SNMP

SNMP Manager/Client Configuration

Configure a Simple Network Management Protocol (SNMP) Manager/Client connection to communicate via Ethernet to IEDs. The RTAC will poll data from these IEDs and store the data it receives in tags. Use the Tag Processor to map these data to any protocol, logs, user logic, etc. The SNMP protocol typically refers to IEDs that make SNMP requests or collect data as managers (in the RTAC platform, this is the same functionality as a client) and IEDs that provide SNMP data as agents (in the RTAC platform, this is the same functionality as a server). As such, the terms manager/client and agent/server will be used interchangeably in the RTAC platform. The RTAC supports version 2C (v2C) of the SNMP protocol.

To add an SNMP Manager/Client connection, select **SNMP** from the **Other Device** menu.

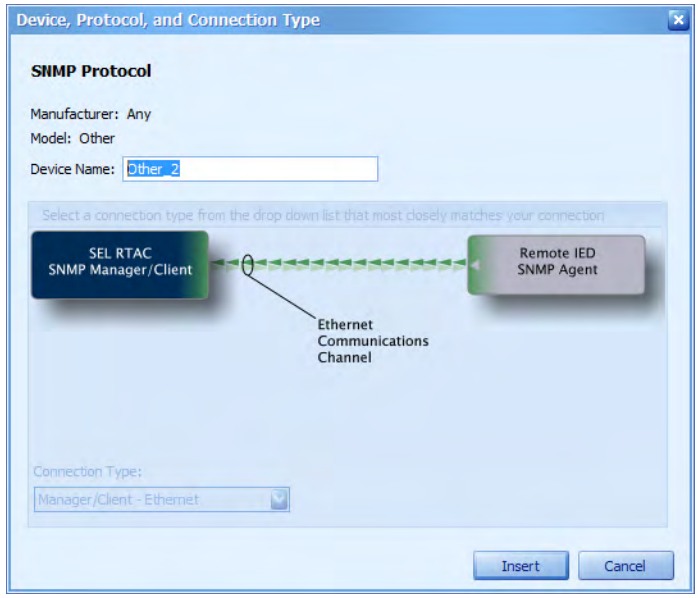


Figure 2.204 Adding SNMP Manager/Client

Give the device connection a unique name and select the connection type, as shown in *Figure 2.204*. Refer to *Client Connection Types on page 31* for a description of each connection type.

Settings Tab

The **Settings** tab contains all configurable items for communications. Check the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description. Type any applicable comments in the blank column to the far right. Click the **Advanced Settings** check box to enable configuration of advanced settings.

Polls Tab

The **Polls** tab contains all the poll requests that the SNMP client will use to collect data from the server. To add additional poll requests use the + button. As many as 50 polls are supported. Each poll will request a list of OIDs, which are configured using the Status OIDs and Control OIDs.

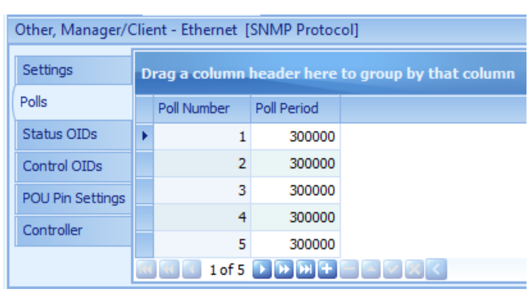


Figure 2.205 SNMP Manager/Client Polls

Table 2.127 Poll Tab Parameters

Column Name	Description
Poll Number	Name of poll group.
Poll Period	The period in which the RTAC will generate this request. The units for poll period are in milliseconds. A value of 0 disables the poll from being issued on a periodic basis. The range of poll period is 0, 1000–100000000 milliseconds.

SNMP OIDs

To configure tags for the SNMP Manager/client, go to the **Status OIDs** or **Control OIDs** tab.

Step 1. Click on the tab for the appropriate tag type to add and configure tags.

Step 2. Click + to add tags. Create only the necessary number of tags to optimize system performance.

Step 3. Rename the tags as necessary.

Step 4. Configure other tag parameters as necessary.

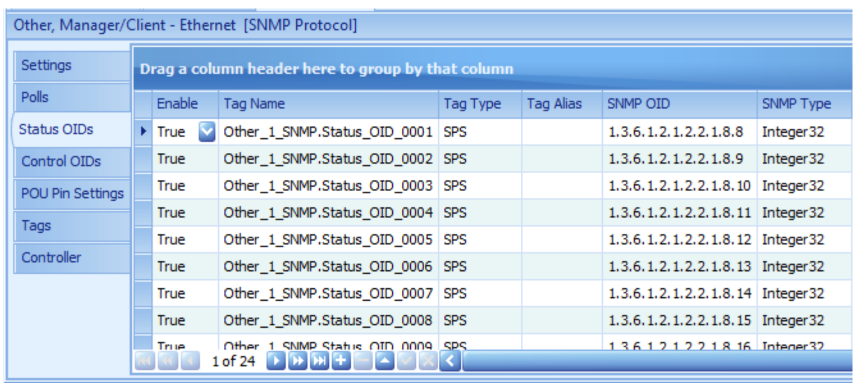


Figure 2.206 SNMP Manager/Client Sector Map

SNMP-Specific Tag Configuration

There are several columns specific to SNMP points. Each have the following behavior.

Table 2.128 SNMP Point Behavior

Column Name	Description
SNMP OID	This is the OID that the RTAC will issue to the server/agent. The period preceding the first digit is optional.
SNMP OID Name	Each digit in the OID has a name correlation. The OID can also be represented by the named format as well as the number format.
SNMP Type	This is the type that the data will be returned to the RTAC via the SNMP protocol definition. This field shows the available tag formats SNMP data can be represented as in the RTACs logic engine.

Column Name	Description
Size	This setting only applies for the data type STRC and SNMP data type Octet String. When a rising edge is applied to the trigger bit in the operSTR structure, the RTAC will write the number of bytes specified by the Size setting from the string in the ctlVal in the operSTR structure.
Treat As	This setting only applies when the RTAC data type is STR and the SNMP Type is OCTET String. Since this SNMP data type can represent both text and number data, these data can be used two ways in the RTAC logic engine. When set to Text, the most significant bit of the data received by the SNMP client will be placed in the most significant bit of the STR tag, creating a visible string. This causes the RTAC to treat the received data as text. When set to Number, the data received by the SNMP client will have the least significant bit placed in the least significant bit of the STR tag. In this mode, the tag will not appear as an ASCII string when viewed while online with the RTAC. This causes the RTAC to treat the data as a received integer.
Poll Number	This setting tells the RTAC which poll request to ask the agent/server for the configured OID. Multiple tags are assignable to the same poll number.

Polling Behavior of SNMP Manager/Client

SNMP uses UDP to exchange information. UDP is a connectionless transport protocol. This means that unlike TCP the RTAC will only know if the agent/server is active and communicating when it receives a response or unsolicited message. If the RTAC generates a request to the agent/server and receives no response for the duration of request time-out setting, the RTAC will issue a series of retry requests. Once the configured series of request retries and subsequent time-outs have expired, the RTAC will assert the offline pin until the next successful communication sequence occurs with the agent/server. During this period of time, all other communication requests will be paused and queued until a response is returned or retries are expired.

Configuring SNMP Traps/Inform in the RTAC

Traps and Inform messages are unsolicited messages sent from the agent/server to the manager/client without a request from the manager/client. When this unsolicited message is sent to the RTAC it will contain the status OID number and the status value. As long as the correct OID is configured in the **Status OIDs** or **Control OIDs** tab, the RTAC will receive the trap and update the appropriate tag. Trap and Inform messages are configured in the agent/server IED for sending data to managers/clients.

POU Pins

Use POU pin settings to view the present state of the SNMP Manager/client operation and to modify some of the behavior of the protocol. Setting the **Visible** field to True will cause the POU pin to appear in the **Controller** tab. See *Table 2.129* for the settings descriptions.

Table 2.129 SNMP Manager/Client POU Pin Settings

Pin Name	Pin Type	Description	Default
EN	Input: BOOL	The EN input enables or disables this specific Function Block instance. Other inputs have no effect while EN is False.	True
Disable_Tag_Updates	Input: BOOL	The POU will not update tags or process changes to tags while this input is True.	False
Disable_Controls	Input: BOOL	While True, the processing of any controllable tags is blocked.	False
Reset Statistics	Input: BOOL	On rising-edge trigger, all counter POU output are reset.	False
Issue_<XX>[a]	Input: BOOL	Request poll to be sent, on the rising edge.	False
Issue_<XX>_Period[a]	Input: Time (0, 25–100000000 milliseconds)	Defines the interval used by the Client to issue the poll. Setting to zero causes the related function to be non-periodic.	Reference Polls–Message_Period
ENO	Output: BOOL	Indicates this specific Function Block Instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False
Offline	Output: BOOL	This output is False when protocol communications are in process. Note: Protocol communications may be in process but no successful data exchange due to settings configuration issue.	False
Controls_Disabled	Output: BOOL	Asserted when Disable_Controls input is asserted. This indicates the client will not issue control commands.	False
Message_Sent_Count	Output: UDINT	Running sum indicating the number of messages transmitted to the remote device.	False
Message_Received_Count	Output: UDINT	Running sum indicating the number of messages received from the remote device.	0
Message_Failure	Output: BOOL	Asserts for a single processor cycle when a message is processed that does not conform to the expected protocol format. This may be because of an incomplete response or corrupted data in the message response.	False
Message_Failure_Count	Output: UDINT	Running sum indicating the number of messages that have successfully been sent or received.	0
Response_Timeout_Count	Output: UDINT	Count of Application Layer Message response timeouts.	0
Issue_<XX>_DN[a]	Output: BOOL	Asserts for a single processing interval once the message is completed successfully. Completion of the message is dependent on the configuration of the message.	False
Issue_<XX>_Delay[a]	Output: BOOL	Asserts when this message is triggered and the previous triggering of this message has not completed. Deasserts when on the rising-edge of Issue_<Name>_DN.	False
Last_Rec_Error_Message	Output: String(255)	Contains the last received error message. The message displayed is the result returned to the RTAC from the SNMP agent/server.	''
Data_Type_Mismatch	Output: BOOL	Will assert if an OID is returned with a data type that does not match the configured data type. This will be cleared on a rising edge of Reset_Statistics.	False

[a]<XX> = Each configured poll.

SNMP Agent/Server Configuration

The SNMP protocol typically refers to IEDs that make SNMP requests or collect data as managers (in the RTAC platform, this is the same functionality as a client) and IEDs that provide SNMP data as agents (in the RTAC platform, this is the same functionality as a server). As such, the terms manager/client and agent/server will be used interchangeably in the RTAC platform. The RTAC supports version 2C (v2C) of the SNMP protocol. To add an SNMP Manager/Client connection, select **SNMP** from the **Other Device** menu.

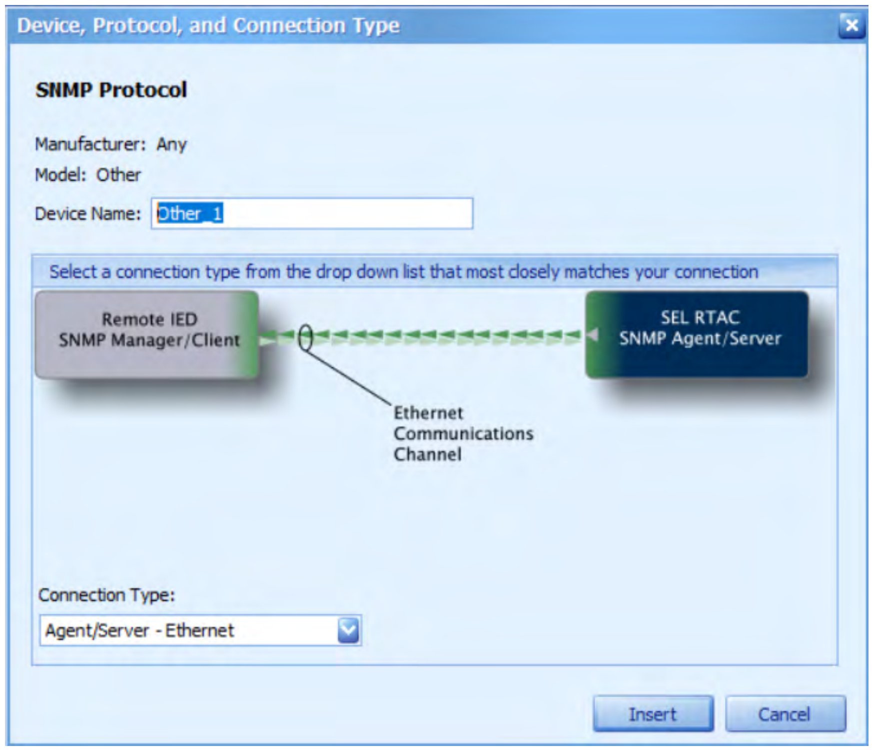


Figure 2.207 Adding SNMP Manager/Client Connection

Give the device connection a unique name and select the connection type, as shown in *Figure 2.207*. Refer to *Client Connection Types on page 31* for a description of each connection type. Only a single instance of an SNMP agent can be added to the RTAC project. Because SNMP is UDP-based, it does not maintain an active connection to a single client. The SNMP agent/server can be configured to answer requests only from a list of manager/client IP addresses or accept requests from any manager/client.

Settings Tab

The **Settings** tab contains all configurable items for communications. Check the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire description. Type any applicable comments in the blank column to the right. Select the **Advanced Settings** check box to enable configuration of advanced settings.

SNMP Agent Data

The RTAC SNMP agent implements the system and interface MIB. It provides additional system data covered in the following table. The SNMP agent/server does not allow for any custom data mappings.

System Tag	Description	SNMP OID	SNMP Data Type
CPU_Burden_Percent_5_Second_Average	The five second average CPU burden percentage	.1.3.6.1.4.1.31823.1.3500.1.1.1.0	Integer32
Memory_KBytes_Used	The number of Kilobytes of memory in use	.1.3.6.1.4.1.31823.1.3500.1.1.2.0	Integer32
Mainboard_Temperature	The temperature of the mainboard in Celsius	.1.3.6.1.4.1.31823.1.3500.1.1.3.0	Integer32

POU Pins

Use POU pin settings to view the present state of the SNMP agent/server operation and to modify some of the behavior of the protocol. Setting the **Visible** field to True will cause the POU pin to appear in the **Controller** tab.

Pin Name	Pin Type	Description	Default
EN	Input: BOOL	The EN input enables or disables this specific Function Block instance. Other inputs have no effect while EN is False.	True
Disable_Tag_Updates	Input: BOOL	The POU will not update tags or process changes to tags while this input is True.	False
Communications_Offline_Timer	Input: TIME	If the SNMP Agent/Server does not receive a request for a time period longer than what is specified by this pin the offline pin will assert.	T#10s
Reset_Statistics	Input: BOOL	On rising-edge trigger, all counter POU output are reset.	False
Message_Sent_Count	Output: UDINT	Running sum indicating the number of messages transmitted to the remote devices.	0
Message_Received_Count	Output: UDINT	Running sum indicating the number of messages received from the remote device.	0
Message_Failure	Output: BOOL	Asserts for a single processor cycle when a message is processed that does not conform to the expected protocol format.	False
Message_Failure_Count	Output: UDINT	Running sum indicating the number of messages that have been detected as failed messages.	0
Invalid_Function_Block_Input	Output: BOOL	The function block has received invalid input which it cannot process.	False
ENO	Output: BOOL	Indicates this specific Function Block Instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False
Offline	Output: BOOL	This output is FALSE when protocol communications are in process. **NOTE** If protocol communications are in process, but there is no successful exchange of data, a settings configuration issue may be present.	False

File Transfer Protocol

Overview

Configure a file transfer protocol (FTP) server POU to communicate with a remote FTP client. The FTP server can use FTP over SSH (SFTP), but does not support FTP over SSL/TLS (FTPS). The FTP server will support as many as 10 simultaneous connections.

The FTP server supports the collection of files from the RTAC under the following directories:

➤ Events: This includes events from SEL CEV, SEL Comtrade, MMS Comtrade, GE Comtrade (Modbus), Alstom Comtrade (Courier), and Axion Comtrade files (from CT/PT cards).

➤ File: This includes any files uploaded or created using the FileIo library.

➤ Projects: If the account type used is administrator or engineer, the FTP server will make any projects uploaded through the web interface available.

➤ CID: If an IEC 61850 configuration is included in the RTAC project, the CID file for the RTAC will be available in this directory.

➤ Logs: This directory contains the log information for the FTP server.

Table 2.130 shows the read/write access to folders based on the account level of the user who accesses the FTP server.

Table 2.130 Read/Write Access By Account Level

Account Type	Events	Files	Projects	CID	Logs
Administrator	Read	Read/Write	Read/Write	Read	Read
Engineer	Read	Read/Write	Read/Write	Read	Read
User Manager	Read	Read	No Access	No Access	No Access
Monitor	Read	Read	No Access	No Access	No Access
HMI Operator	Read	Read	No Access	No Access	No Access
FTP User	Read	Read/Write	No Access	No Access	Read

The FTP server will support authentication either with local RTAC accounts or through any configured LDAP accounts.

FTP Logging

A log file of FTP server interactions is kept on the RTAC. This log file will grow to a maximum of 10 MB. Once it reaches this size, a second log file will begin. Once the second log file reaches 10 MB, the first log file is truncated and new activity is logged to that file. This rotation continues as each log file reaches its maximum file size. These log files are available via the web interface or the FTP server.

FTP Server Configuration

Configure the FTP server to communicate via Ethernet to a FTP client. Since the FTP protocol is designed to transfer files, no tag configuration parameters will be available, unlike other configurable server protocols in the RTAC. Adding the FTP server POU to the project configuration enables the project to communicate with as many as ten FTP client connections with a single defined POU in the RTAC project during runtime.

Settings Tab

The **Settings** tab contains all configurable items necessary for communication. Refer to the description column for information about each setting. Move the main window horizontal slider, resize the column width, or hover your mouse over the description cell to view the cell content in its entirety. Type any applicable comments in the blank column to the far right.

All connections to the RTAC FTP server must use the same connection type. You cannot configure a mix of FTP and SFTP connections. For FTP-based communications, the RTAC contact port is configurable. The **Settings** tab contains the ability to restrict access to a certain set of IP addresses allowed to access the FTP server. You can configure the server to accept all FTP connection requests, or restrict access to a comma-separated list of FTP clients who will be allowed to authenticate and use the FTP server. In addition to whitelisting IP addresses, Ethernet interfaces on the RTAC can be blacklisted from FTP access. In the **Settings** tab, you can enter a comma-separated list that will prevent FTP communications on that Ethernet interface. For example, if Blocked Network Interfaces is set to "Eth_F, Eth_02," only Ethernet Port 1 on that RTAC can receiving FTP connection requests, regardless of the Allow Anonymous IP Client or Client IP Address(es) settings. When the FTP POU is added to the RTAC configuration, multiple clients are able to simultaneously access the file system and read files at the same time.

Table 2.131 POU Pin Settings

Parameter	Description	Default
En	The EN input enables or disables this specific Function Block instance. Other inputs have no effect while EN is False.	True
Reset_Statistics	On rising-edge trigger, all counter POU output are reset.	False
Active Connection	Number of currently connected FTP clients	0
ENO	Indicates this specific Function Block Instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False
Files_Downloaded	The number of files that have downloaded from the RTAC since the last rising edge of ENO or Reset_Statistics. If no manipulation of EN or Reset_Statistics has been performed, this number will often reflect the number of files downloaded since the last project download or power cycle.	0

Parameter	Description	Default
Files_Uploaded	The number of files that have uploaded to the RTAC since the last rising edge of ENO or Reset_Statistics. If no manipulation of EN or Reset_Statistics has been performed, this number will often reflect the number of files uploaded since the last project download or power cycle.	0
Offline	This output is FALSE when protocol communications are in process. Note: If protocol communications are in process, but there is no successful exchange of data, a settings configuration issue may be present.	False

Web API Communications

Web APIs are available for data exchange starting in firmware version R146. All API communications require an authenticated user using basic authentication in https with a valid user account on the RTAC that has the appropriate permissions. Each API documents the necessary permissions.

Any data sent or returned via the APIs that represent a time stamp must conform to the ISO 8601 string definition of a time stamp.

API Documentation

Web API formats, inputs, and outputs are documented on the RTAC web interface. Click on an API category to see additional details about each API.

Only authenticated users can see all details of each API, including the description of each API and its inputs, outputs, and potential returned error messages.

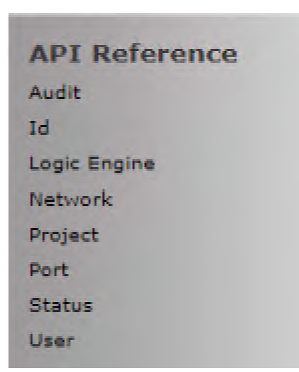


Figure 2.208 API Documentation Example

Network Audit Utilities API

The network audit utilities are only supported on the SEL-3532, SEL-3555, and SEL-3560. The network audit utility has the following capabilities.

The network audit utilities are only supported on the SEL-3532, SEL-3555, and SEL-3560. Audits are restricted to local area networks only, and cannot be performed in the 127.0.0.0/8 network. The network audit utility includes the following functionality:

- Targeting of a single host, multiple hosts, or an entire network
- Host discovery
- Reporting of each open TCP and UDP port for each host listed or discovered

Audit results are available in the AUDITS directory on the RTAC file manager.

Use the API to trigger a network audit and then use the response to retrieve the results in a separate request.

Required Permissions:

- API Login
- Manage Audits
- Retrieve Audits

Starting in firmware version R147, network audits can also be initiated and collected on the RTAC's web interface.

ID API

The ID API provides summary information about the RTAC relating to firmware and licensing. It contains the following information:

- Serial Number
- Product Number
- Firmware Version
- Licensed Features

Logic Engine API

This API allows current tag information to be queried. Tags or variables defined in IEC 61131-3 logic are not available via this API. The logic engine API allows users to get current values from the logic engine and update logic engine values through the API. Each request has its own permission requirements. A tag name referenced in the URL must have the following format: Global Variable list name/variable name. For example, if a DNP client had the configuration name feeder_2_DNP and a binary input name of BI_0000, the tag name in the URL would be feeder_2_DNP/BI_0000. POU pins are also available—to get the offline pin status, the tag name would be feeder_2_DNP_POU/Offline. Tag names are referenced in the same manner in logic, but instead of the . notation, a "/" is used.

Required Permissions:

- API Login
- Logic Engine API Read(GET command)
- Logic Engine API Read(PUT command)

Poll Logic Engine Tag

GET Request for Logic Engine Value

GET https://{RTAC IP Address}/api/v1/logic-engine/symbols/{logic engine tag name}

There is JSON body content for this request.

Example:

GET https://192.168.1.10/api/v1/logic-engine/symbols/SystemTags/DEVCODE

GET Response for Logic Engine Value

The RTAC will return the entire tag structure for the referenced tag. Because the tag response will change based upon the tag structure, there is no defined JSON schema response. Below is an example of the STR tag structure being returned.


```
{
  "strVal": "73",
  "q": {
    "validity": "good",
    "detailQual": {
      "overflow": false,
      "outOfRange": false,
      "badReference": false,
      "oscillatory": false,
      "failure": false,
      "oldData": false,
      "inconsistent": false,
      "inaccurate": false
    },
    "source": "process",
    "test": true,
    "operatorBlocked": false
  },
  "t": {
     "value": "2019-09-24T20:04:29.016667+00:00",
     "quality": {
        "leapSecondsKnown": false,
        "clockFailure": false,
        "clockNotSynchronized": false,
        "accuracy": "T2 (100us)"
     },
     "source": {
       "value": "free running",
       "priority": 129
     }
  }
}
```


Figure 2.209 GET Logic Engine Value Response Example

Write Logic Engine Tag

The write logic engine tag request will update the logic engine value for the requested tag name. There are several requirements for the write to be successful:

- Write format must exactly match the tag structure as returned in the GET request for logic engine value response
- No overflow on any integer or float type
- Enumeration strings are valid
- Strings contain valid ASCII characters
- Strings cannot contain any Unicode special characters
- Hexadecimal values are formatted with a leading 0x
- Datetime value adheres to valid ISO 8601 standard string for date and time
- Tag List name and tag name are separated by "/"

PUT Request for Logic Engine Value Write

PUT https://{RTAC IP Address}/api/v1/logic-engine/symbols/{logic engine tag name}

Example:

> PUT https://192.168.1.10/api/v1/logic-engine/symbols/SystemTags/DEVCODE

```json
{
  "strVal": "Hello World!",
  "q": {
    "validity": "questionable",
    "detailQual": {
      "overflow": false,
      "outOfRange": true,
      "badReference": false,
      "oscillatory": false,
      "failure": true,
      "oldData": true,
      "inconsistent": false,
      "inaccurate": true
    },
    "source": "substituted",
    "test": true,
    "operatorBlocked": true
  },
  "t": {
    "value": "2019-10-08T20:04:29+00:00",
    "quality": {
        "leapSecondsKnown": false,
        "clockFailure": false,
        "clockNotSynchronized": false,
        "accuracy": "T5 (1us)"
    },
    "source": {
      "value": "IRIG-B",
      "priority": 4
    }
  }
}
```


Figure 2.210 PUT Logic Engine Value Request Example

Network API

This API provides information about the network interfaces on the RTAC.

Required Permissions:

- API Login
- Network Read

The following information can be accessed via this API:

- List of interfaces on the RTAC
- MAC addresses
- Operational status of each interface
- IP address and configured gateway address
- Configuration information about the interfaces

Ports API

This API provides information about the open ports on the RTAC interfaces. It will return all TCP and UDP ports that are open on the RTAC including the service that is using the port.

Required Permissions:

- API Login
- Network Read

Projects API

This API provides information about and allows interaction with the active RTAC project.

Required Permissions:

- API Login
- Device Password Read (GET Password Vault)
- Device Password Write (PUT Password Vault update)
- List projects on the RTAC
- Put projects on the RTAC
- Read projects on the RTAC
- Activate projects on the RTAC
- Delete projects on the RTAC

Status API

This API provides information about the project on the RTAC including memory usage, storage usage, and users logged in.

Required Permissions:

- API Login

The following information is returned in this API.

- Active Project Name
- Current Memory Statistics
- Current Storage Statistics
- Logged In User Statistics

Users API

This API provides information about the local users on the RTAC.

Required Permissions:

- API Login
- Account Write

The following information is returned in this API:

- ➤ Returns role associated with account
- ➤ Returns permissions associated with account
- ➤ Time when account was created
- ➤ Time when account password was last changed

EtherNet/IP

Overview

This section describes the EtherNet/IP protocol implementation on the RTAC, including general specifications of the implementation, CIP object model, allocation of CIP connections, ACSELERATOR RTAC settings, and the Electronic Data Sheet (EDS) file.

The RTAC supports three ways of exchanging data via EtherNet/IP protocol:

- ➤ Implicit Message Adapter The I/O data are mapped in Assembly object instances. The RTAC exchanges this I/O data via EtherNet/IP implicit class 1 connections with a remote EtherNet/IP scanner device using UDP packets.
- ➤ Explicit Message Server The I/O data are mapped in Assembly object instances. The RTAC responds to generic TCP EtherNet/IP explicit message requests initiated by a remote EtherNet/IP client device.
- ➤ Explicit Message Client The RTAC initiates vendor-specific explicit message services to exchange data with a remote EtherNet/IP explicit message server.

Specifications

EtherNet/IP Services

Supported Services	Implicit Message Adapter (Class 1)
	Explicit Message Server (Class 3 and unconnected)
	Explicit Message Client (Unconnected)

CIP Model

Implemented objects	Identity object
	Message Router object
	Assembly object
	Connection Manager object
	File object
	TCP/IP object
	Ethernet link object

Implicit Message Adapter Specifications

Number of connections	As many as 128 (Class 1 and Class 3)
Class 1 connection types	Unicast Multicast
Class 1 connection transport types	Exclusive owner Input only Listen only
Class 1 connection trigger types	Cyclic Change of State
Input Only heartbeat connection point	238
Listen Only heartbeat connection point	237

Explicit Message Client Specifications

Supported Message Types	CIP[a] and PCCC[b]
CIP services supported	CIP Data Table Read (0x4C) CIP Data Table Write (0x4D) CIP Read Tag Fragmented (0x52) CIP Write Tag Fragmented (0x53)
PCCC services supported	SLC Protected Typed Logical Read (CMD=0x0F, FNC=0xA2) SLC Protected Typed Logical Write (CMD=0x0F, FNC=0xAA)
Message type	Unconnected (UCMM)

[a]Supported devices: Rockwell Automation controllers ControlLogix and CompactLogix.

[b]Supported devices: Rockwell Automation controllers Micrologix 1000/1400 and SLC5/05.

CIP Data Model

Profile

Class Name	Class ID	Number of Instances
Identity Object	0x01	1
Message Router Object	0x02	1
Assembly Object	0x04	Determined by the user based on application
Connection Manager Object	0x06	1
File Object	0x37	2
TCP/IP Object	0xF5	See *TCP/IP Object (0xF5) on page 384*
Ethernet Link object	0xF6	See *Ethernet Link Object (0xF6) on page 385*

Identity Object (0x01)

Instances Implemented

The RTAC implements one instance (instance ID = 1) of the identity object.

List of Class Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Revision	GET	UINT	1	The revision of this CIP object
2	Max Instance	GET	UINT	1	The maximum Assembly object instance ID
3	Number of Instances	GET	UINT	1	Total number of Assembly objects
4	Optional Attribute List	GET	[UINT, Array of UINT]	[21, [0,...,0]]	[Number of optional instance attributes, List of optional instance attributes]
6	Maximum ID Number Class Attributes	GET	UINT	7	Maximum class attribute ID
7	Maximum ID Number Instance Attributes	GET	UINT	21	Maximum instance attribute ID

List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Vendor ID	GET	UINT	865	
2	Device Type	GET	UINT	0x0E	
3	Product Code	GET	UINT		Refer to *Managing Multiple RTAC EtherNet/IP Configurations in the Same Network on page 390*
4	[Major Revision, Minor Revision]	GET	[USINT, USINT]	[1,1]	
5	Status	GET	WORD		Refer to status WORD bits table
6	Serial Number	GET	UDINT		Lower 6 octets of the Eth_1 port MAC address
7	Product Name	GET	STRING	'RTAC'	Refer to *Managing Multiple RTAC EtherNet/IP Configurations in the Same Network on page 390*
21	Catalog Number	GET	STRING	'SEL-RTAC'	

Status WORD Bits Table

The Status WORD bits not listed are always set to 0.

Bit Number	Name	Description
0	Owned	TRUE if at least one remote scanner has established an Exclusive Owner class 1 connection to the RTAC FALSE if the RTAC has no active Exclusive Owner connections to a scanner
2	Configured	Always TRUE
4 to 7	Extended Device Status	Hexadecimal value: 2: A class 1 connection is timed out 3: No class 1 connection is established 6: At least one class 1 connection is active 7: In any other case

Supported Services

Service Code	Service Name	Class	Instance	Description
0x01	Get Attributes All	Yes	Yes	Returns a list of all of the values of the attributes
0x05	Reset	No	Yes	Restarts the EtherNet/IP service in the RTAC. Only reset type 0 is allowed.
0x0E	Get Attribute Single	Yes	Yes	Returns the value of a specific attribute

Message Router Object (0x02)

List of Class Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Revision	GET	UINT	1	The revision of this CIP object
2	Max Instance	GET	UINT	3	The maximum Message Router object instance ID
3	Number of Instances	GET	UINT	1	Total number of Message Router object instances
4	Optional Attribute List	GET	[UINT, Array of UINT]	[2, [3,1]]	[Number of optional instance attributes, List of optional instance attributes]
5	Optional Service List	GET	[UINT, Array of UINT]	[10, [7,3,0,0,0,0,0,0,0,0]]	[Number of optional service codes, List of optional service codes]
6	Maximum ID Number Class Attributes	GET	UINT	7	Maximum class attribute ID
7	Maximum ID Number Instance Attributes	GET	UINT	3	Maximum instance attribute ID

List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Class List	GET	[UINT, Array of UINT]		Implemented object list
2	Maximum Connections	GET	UINT		Maximum number of connections supported
3	Number of Connections	GET	UINT		Number of connections currently used

Supported Services

Service Code	Service Name	Class	Instance	Description
0x01	Get Attribute All	Yes	Yes	Returns a list of all of the attributes
0x0E	Get Attribute Single	Yes	Yes	Returns the value of a specific attribute
0x0A	Multiple Service Packet	No	Yes	

Assembly Object (0x04)

Instances Implemented

The ACSELERATOR RTAC user settings define the number of Assembly object instances, instance IDs, and data content of each instance.

List of Class Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Revision	GET	UINT	2	The revision of this CIP object
2	Max Instance	GET	UINT	Determined by settings	The maximum Assembly object instance ID defined by the user
3	Number of Instances	GET	UINT	Determined by settings	Total number of Assembly object instances defined by the user
4	Optional Attribute List	GET	[UINT, Array of UINT]	[1,4]	[Number of optional instance attributes, List of optional instance attributes]
6	Maximum ID Number Class Attributes	GET	UINT	7	Maximum class attribute ID
7	Maximum ID Number Instance Attributes	GET	UINT	4	Maximum instance attribute ID

List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Number of Members	GET	UINT		Number of Assembly tag members defined by the user
2	Member List	GET	Array of [UINT, UINT, EPATH]		

Attr. ID	Name	Access	Data Type	Default	Description
3	Data	GET, SET	Array of Bytes		Data map defined in ACSELERATOR RTAC
4	Size	GET	UINT		Number of bytes in attribute 3

Supported Services

Service Code	Service Name	Class	Instance	Description
0x0E	Get Attribute Single	Yes	Yes	Returns the value of a specific attribute
0x10	Get Attribute Single	No	Yes	Sets the value of a specific attribute
0x18	Get Member	No	Yes	Returns the value of a member of the Data attribute
0x19	Set Member	No	Yes	Modifies the value of a member of the Data attribute

Connection Manager Object (0x06)

List of Class Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Revision	GET	UINT	4	The revision of this CIP object
2	Max Instance	GET	UINT		The maximum Connection Manager object instance ID
3	Number of Instances	GET	UINT		Total number of Connection Manager object instances
4	Optional Attribute List	GET	[UINT, Array of UINT]	[8,[0,0,0,0,0,0,0,0]]	[Number of optional instance attributes, List of optional instance attributes]
6	Maximum ID Number Class Attributes	GET	UINT	7	Maximum class attribute ID
7	Maximum ID Number Instance Attributes	GET	UINT	20	Maximum instance attribute ID

List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Open Requests	GET/ SET	UINT		Number of FWD Open service requests received
2	Open Format Rejects	GET/ SET	UINT		Number of FWD Open service requests rejected because of bad format
3	Open Resource Rejects	GET/ SET	UINT		Number of FWD Open service requests rejected because of lack of resources
4	Open Other Rejects	GET/ SET	UINT		Number of FWD Open service requests rejected for reasons other than bad format or lack or resources
5	Close Requests	GET/ SET	UINT		Number of FWD Close service requests received
6	Close Format Rejects	GET/ SET	UINT		Number of FWD Close service requests rejected because of bad format

Attr. ID	Name	Access	Data Type	Default	Description
7	Close Other Rejects	GET/ SET	UINT		Number of FWD Open service requests rejected for reasons other than bad format
8	Connection Timeouts	GET/ SET	UINT		Number of connection timeouts
15	I/O Packets per Second	GET	UDINT		Current I/O packets per second
17	Explicit Packets per Second	GET	UDINT		Total number of explicit packets sent and received over the last second
18	Missed I/O Packets	GET	UDINT		Total number of missed I/O packets
19	CIP I/O Connections	GET	UDINT		Total number of CIP connections in use
20	CIP Explicit Connections	GET	UDINT		Total number of CIP Explicit Messaging connections in use

Supported Services

Service Code	Service Name	Class	Instance	Description
0x01	Get Attribute All	Yes	Yes	Returns a list of all of the attributes
0x0E	Get Attribute Single	Yes	Yes	Returns the value of a specific attribute
0x10	Set Attribute Single	No	Yes	Sets the value of a specific attribute
0x02	Set Attribute All	No	Yes	Sets the value of all attributes
0x54	Forward Open	No	Yes	Establishes a CIP connection
0x4E	Forward Close	No	Yes	Closes a CIP connection
0x5B	Large Forward Open	No	Yes	Establishes a CIP connection that allows a large connection size
0x5A	Get Connection Owner	No	Yes	Returns data about the connection owns the object

File Object (0x37)

The File object stores the Electronic Data Sheet (EDS) and icon files. The EDS file is automatically generated by ACSELERATOR RTAC based on the specific user configuration and transferred to the RTAC when the project is loaded.

Instances Implemented

The RTAC implements two instances of the File object:

- Instance 0xC8 returns an uncompressed version of the EDS file with embedded icons.
- Instance 0xC9 returns a compressed version of just the icon file.

List of Class Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Revision	GET	UINT	1	The revision of this CIP object
2	Max Instance	GET	UINT	201	

Attr. ID	Name	Access	Data Type	Default	Description
3	Number of Instances	GET	UINT	2	
6	Maximum ID Number Class Attributes	GET	UINT	32	Maximum class attribute ID
7	Maximum ID Number Instance Attributes	GET	UINT	11	Maximum instance attribute ID
32	Directory	GET	[UINT, STRINGI, STRINGI] [UINT, STRINGI, STRINGI]	[0xC8, (ENG)'EDS and Icon Files', (ENG)'EDS.txt'] [0xC9, (ENG)'Related EDS and Icon Files', (ENG)'EDSCollection. gx']	List of all File object instance and file names present in the RTAC and the associated instance numbers.

0xC8 List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	State	GET	USINT	2	
2	Instance Name	GET	STRINGI	(ENG)'EDS and Icon Files'	
3	File Format Version	GET	UINT	1	
4	File Name	GET	STRINGI	(ENG)'EDS.txt'	
5	File Revision	GET	[USINT, USINT]		[Major EDS revision user setting, Minor EDS revision user setting]
6	File Size	GET	UDINT		Size of the EDS file in bytes
7	File Checksum	GET	UINT		Checksum of the EDS file (two's complement of the 16-bit sum of all octets in the file)
8	Invocation Method	GET	USINT	255	
9	File Save Parameters	GET	USINT	0	
10	File Access Rule	GET	USINT	1	
11	File Encoding Format	GET	USINT	0	

0xC9 List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	State	GET	USINT	2	
2	Instance Name	GET	STRINGI	(ENG)'Related EDS and Icon Files'	
3	File Format Version	GET	UINT	1	
4	File Name	GET	STRINGI	(ENG)'EDSCollection. gz'	
5	File Revision	GET	[USINT, USINT]	[1,1]	[Major Revision, Minor Revision]
6	File Size	GET	UDINT		Size of the loaded file in bytes

Attr. ID	Name	Access	Data Type	Default	Description
7	File Checksum	GET	UINT		Checksum of the EDSCollection file (two's complement of the 16-bit sum of all octets in the file)
8	Invocation Method	GET	USINT	255	
9	File Save Parameters	GET	USINT	0	
10	File Access Rule	GET	USINT	1	
11	File Encoding Format	GET	USINT	1	

Supported Services

Service Code	Service Name	Class	Instance	Description
0x0E	Get Attribute Single	Yes	Yes	Returns the value of a specific attribute
0x4B	Initiate Upload	No	Yes	
0x4F	Upload Transfer	No	Yes	

TCP/IP Object (0xF5)

Instances Implemented

The number of instances of the TCP/IP object depends on the number of instances of the Ethernet Link object and the Bridge/Bond modes in use, as follows:

- ➤ An instance for each Ethernet Link object instance that is not part of a Bridge/Bond/PRP set.
- ➤ An instance for each set of Ethernet Link object instances that make up a Bridge/Bond/PRP set.

List of Class Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Revision	GET	UINT	4	The revision of this CIP object
2	Max Instance	GET	UINT		The maximum TCP/IP object instance ID
3	Number of Instances	GET	UINT		Total number of TCP/IP object instances
4	Optional Attribute List	GET	[UINT, Array of UINT]	[4, [8,9,16,17]]	[Number of optional instance attributes, List of optional instance attributes]
6	Maximum ID Number Class Attributes	GET	UINT	7	Maximum class attribute ID
7	Maximum ID Number Instance Attributes	GET	UINT	17	Maximum instance attribute ID

List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Status	GET	DWORD	2	
2	Configuration Capability	GET	DWORD	192	Any change in the configuration will be updated when the server is restarted.
3	Configuration Control	GET	DWORD	0	IP addresses must be configured statically. DHCP and DNS are not supported.
4	Physical Link Object	GET	[UINT, EPATH]		[Path size, Path to the corresponding Ethernet link object instance]
5	Interface Configuration	GET	[UDINT, UDINT, UDINT, UDINT, UDINT, STRING]		[IP address, Network mask, Gateway address, 0, 0, null]
6	Host Name	GET	STRING		Host name of the RTAC, as shown in the dashboard of the RTAC web interface
8	TTL Value	GET/ SET	USINT	1	
9	Mcast Config	GET/ SET	[USINT, USINT, UINT, UDINT]		[Alloc control, Reserved, Num Mcast, Mcast Start Address]
13	Encapsulation Inactivity Timeout	GET/ SET	UINT	120	
16	Active TCP Connections	GET	UINT		
17	Non-CIP Encapsulation Messages per second	GET	UDINT		

Supported Services

Service Code	Service Name	Class	Instance	Description
0x01	Get Attribute All	Yes	Yes	Returns a list of all of the attributes
0x0E	Get Attribute Single	Yes	Yes	Returns the value of a specific attribute
0x10	Set Attribute Single	No	Yes	Sets the value of a specific attribute

Ethernet Link Object (0xF6)

Instances Implemented

The number of instances of the Ethernet Link object is equal to the number of Ethernet interfaces listed in the Ethernet Ports setting in ACSELERATOR RTAC. Each Ethernet Link instance corresponds to an Ethernet interface in the list.

When going online, ACSELERATOR RTAC will verify that all configured interfaces for EtherNet/ IP are valid according to the following rules:

- Interfaces in this list are not available for EtherNet/IP:
  - USB interface (172.29.131.1)
  - Any interface that is enabled for EtherCAT
  - Any interface with DHCP enabled
  - Any disabled interface
- Interfaces in Bridge/Bonding/PRP mode require that all of the interfaces within the Bridge/Bonding group or PRP pair must also be listed in the Ethernet Ports setting.

If one or more interfaces in the Ethernet Ports setting are not available for EtherNet/IP, ACSELERATOR RTAC will not send the project.

List of Class Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Revision	GET	UINT	4	The revision of this CIP object
2	Max Instance	GET	UINT		The maximum TCP/IP object instance ID
3	Number of Instances	GET	UINT		Total number of TCP/IP object instances
4	Optional Attribute List	GET	[UINT, Array of UINT]		[Number of optional instance attributes, List of optional instance attributes]
6	Maximum ID Number Class Attributes	GET	UINT	7	Maximum class attribute ID
7	Maximum ID Number Instance Attributes	GET	UINT	11	Maximum instance attribute ID

List of Instance Attributes

Attr. ID	Name	Access	Data Type	Default	Description
1	Interface Speed	GET	UINT		Speed (Mbps) in use on the corresponding interface
2	Interface flags	GET	DWORD		See description in the Interface Flags Bit table
3	Physical Address	GET	USINT[6]		MAC address of the corresponding interface
10	Interface Label	GET	STRING		RTAC interface name. E.g. "Eth_02"
11	Interface Capability	GET	[DWORD, USINT]		[Capability bits, Array Element Count]

Interface Flags Bit Table

Bit Number	Name	Description
0	Link Status	0: The Ethernet interface link is inactive 1: The link is active
1	Half/Full Duplex	0: The interface is running half duplex 1: The interface is running full duplex

Bit Number	Name	Description
2–4	Negotiation Status	Octal unsigned value: 0: Autonegotiation in progress 1: Autonegotiation and speed detection failed. Using default values 2: Autonegotiation failed, but detected speed 3: Successfully negotiated speed and duplex 4: Autonegotiation not attempted
5	Manual Setting Requires Reset	Set to 1

Supported Services

Service Code	Service Name	Class	Instance	Description
0x01	Get Attribute All	Yes	Yes	Returns a list of all of the attributes
0x0E	Get Attribute Single	Yes	Yes	Returns the value of a specific attribute

CIP Connections

The user settings determine the total number of CIP connections used by a project. The number of CIP connections allocated in a project is determined as follows:

- The number of Class 1 connections allocated in a project is determined by adding up the following quantities:
  - The value of the Maximum Input Only Class 1 connections under the **Settings** tab
  - The value of Maximum Listen Only Class 1 connections under the **Settings** tab
  - The number of Exclusive Owner connections defined under the I/O Connections
- The number of Class 3 connections allocated in a project is determined by the value of the Maximum Class 3 connections setting under the **Settings** tab.
- The total number of CIP connections is determined by adding up the number of Class 1 connections and the number of Class 3 connections.

The total number of CIP connections in an RTAC project is limited to 128.

ACSELERATOR RTAC Settings and Configuration

Configuration

Settings

The **Settings** tab contains all the configurable items for the EtherNet/IP communication protocol.

Assembly Instances

The **Assembly Instances** tab is used to define the instances of the Assembly object. Each Assembly instance is a collection of data (as much as 500 bytes worth) that is either transmitted to the scanner (input assembly) or received from the scanner (output assembly).

Assembly Data Map

Assembly Data Map tabs are created automatically for each Assembly instance configured in the **Assembly Instance** tab. Use these tabs to create I/O data points for each Assembly instance. As much as 500 bytes worth of data may be added to each Assembly instance; however, the total bytes used across all Assembly instances cannot exceed 10,000.

I/O Connections

The **I/O Connections** tab is used to define the EtherNet/IP I/O connections (Class 1 connections) that can be accessed from a remote IED/scanner. Depending on the type of connection, each I/O connection defined in this tab has one or two Assembly instances. While Exclusive Owner connections may have an input and an output Assembly instance, Input Only and Listen Only connections may only have input Assembly instances associated with them. All assemblies assigned to I/O connections must be previously defined in the **Assembly Instances** tab. Assemblies used as output assemblies must be defined with read-write access.

Remote Devices

The **Remote Devices** tab is used to enter a list of all EtherNet/IP explicit message targets. The **Remote Devices** tab is only visible when Enable Explicit Message Client is set to True.

CIP Tag Messages

The **CIP Messages** tab is used to configure individual EtherNet/IP explicit messages (CIP Data Table Read and CIP Data Table Write) to devices defined in the **Remote Devices** tab. This tab is only visible when Enable Explicit Message Client is set to True.

PCCC Messages

The **PCCC Messages** tab is used to configure individual EtherNet/IP explicit messages (SLC Protected Typed Logical Read and SLC Protected Typed Logic Write) to devices defined in the **Remote Devices** tab. This tab is only visible when Enable Explicit Message Client is set to True.

POU Pin Setting

The **POU Pin Settings** tab is used to view the present state of the EtherNet/IP protocol and to access protocol statistics.

Table 2.132 SNMP Manager/Client POU Pin Settings

Pin Name	Pin Type	Description	Default
EN	Input: BOOL	The EN input enables or disables this specific Function Block instance. Other inputs have no effect while EN is False.	True
Reset_Statistics	Input: BOOL	On rising-edge trigger, all counter POU output are reset.	False
ENO	Output: BOOL	Indicates this specific Function Block Instance is active if True. If False, the inputs have no effect and the outputs are not updated.	False
Offline	Output: BOOL	If False, indicates that there is at least one active EtherNet/IP session. This pin is set to True when the communication link has been idle for a time specified by the Encapsulation Inactivity Timeout attribute of the corresponding TCP/IP object instance.	False
Class_1_Message_Sent_Count	Output: UDINT	Total number of Class 1 implicit messages sent out by the RTAC	0
Class_1_Message_Received_Count	Output: UDINT	Total number of Class 1 implicit messages received by the RTAC	0
Class_1_Mesage_Failure_Count	Output: UDINT	Total number of Class 1 implicit message failures	0
Class_1_Message_Success_Count	Output: UDINT	Total number of successful Class 1 messages	0
Class_1_Message_Lost_Count	Output: UDINT	Increases when the RTAC receives a Class 1 message with a sequence number greater than the next expected sequence number.	0
Class_1_Active_Connections	Output: UDINT	Current number of active Class 1 connections	0
Class_3_Active_Connections	Output: UDINT	Current number of active Class 3 connections	0
Class_3_Message_Sent_Count	Output: UDINT	Total number of Class 3 explicit message responses sent out by the RTAC	0
Class_3_Message_Received_Count	Output: UDINT	Total number of Class 3 explicit message requests received by the RTAC	0
Unconnected_Message_Sent_Count	Output: UDINT	Total number of unconnected explicit messages sent out by the RTAC	0
Unconnected_Message_Received_Count	Output: UDINT	Total number of unconnected explicit messages received by the RTAC	0
Message_Timeout_Count	Output: UDINT	Increases by one when the RTAC fails to receive a message (Class 1 or Class 3) from the scanner within the expected Requested Packet Interval (RPI).	0
<Connection Name>_Active_Consumers	Output: UDINT	Number of active consumers subscribed to the associated I/O connection	0
<Connection Name>_Timeout_Count	Output: UDINT	Sum of timed-out messages for all associated consumers	0
<Connection Name>_Message_Lost_- Count	Output: UDINT	Sum of lost packets for all associated consumers	0
Send_<Device Name>_<Message Name>	Input: BOOL	On a rising-edge, triggers the message.	FALSE
<Device Name><Message Name>_DN	Output: BOOL	Asserts for one RTAC processing interval to indicate that the message transaction is complete.	FALSE

Pin Name	Pin Type	Description	Default
<Device Name><Message Name>_Status	Output: UDINT	Provides the status of the last message sent	
<Device Name><Message Name>_Data	Input/Output	Array of data. Provides access to the message data.	

Managing Multiple RTAC EtherNet/IP Configurations in the Same Network

When multiple RTAC units with different EtherNet/IP configurations (Assemblies, assembly maps, and I/O Connection Point settings) coexist in the same EtherNet/IP network, the Configuration ID setting is used to uniquely identify each configuration. RTAC units with different EtherNet/IP configurations should use a different value for the Configuration ID setting.

The Configuration ID setting contributes to the Product Code attribute (attribute ID #3 of the Identity object). The RTAC Product Code attribute is a two-byte unsigned integer number in which the most-significant byte is 35 and the least-significant byte is set by user via the Configuration ID setting in ACSELERATOR RTAC. This allows for a Product Code range between 8960 and 9215. Each different EtherNet/IP configuration in the network will have a different Product Code. This ensures that EtherNet/IP scanners will identify each different configuration as a separate device, thus preserving the EDS file for each.

The Product Name setting in ACSELERATOR RTAC shall be used to assign a unique identification to each different Product Code. This setting maps directly into the Product Name attribute (attribute ID #7) of the Identity object. This facilitates the identification of each EtherNet/IP data map when configuring an EtherNet/IP scanner. The following figure illustrates how to use the Configuration ID and Product Name settings to identify different EtherNet/IP configurations in the same network. In the figure, the Adapter 1 and Adapter 2 RTAC devices have the same EtherNet/IP configuration, and therefore the same Configuration ID and Product Name settings are assigned to them. Adapter 3 has a different EtherNet/IP configuration than Adapters 1 and 2, and a different Configuration ID and Product Name settings are assigned to it. Each unique Configuration ID setting results in a different Product Code attribute value in the Identity object and Product Code entry in the EDS file, which results in the EtherNet/IP scanner configurator software identifying each as a different product.

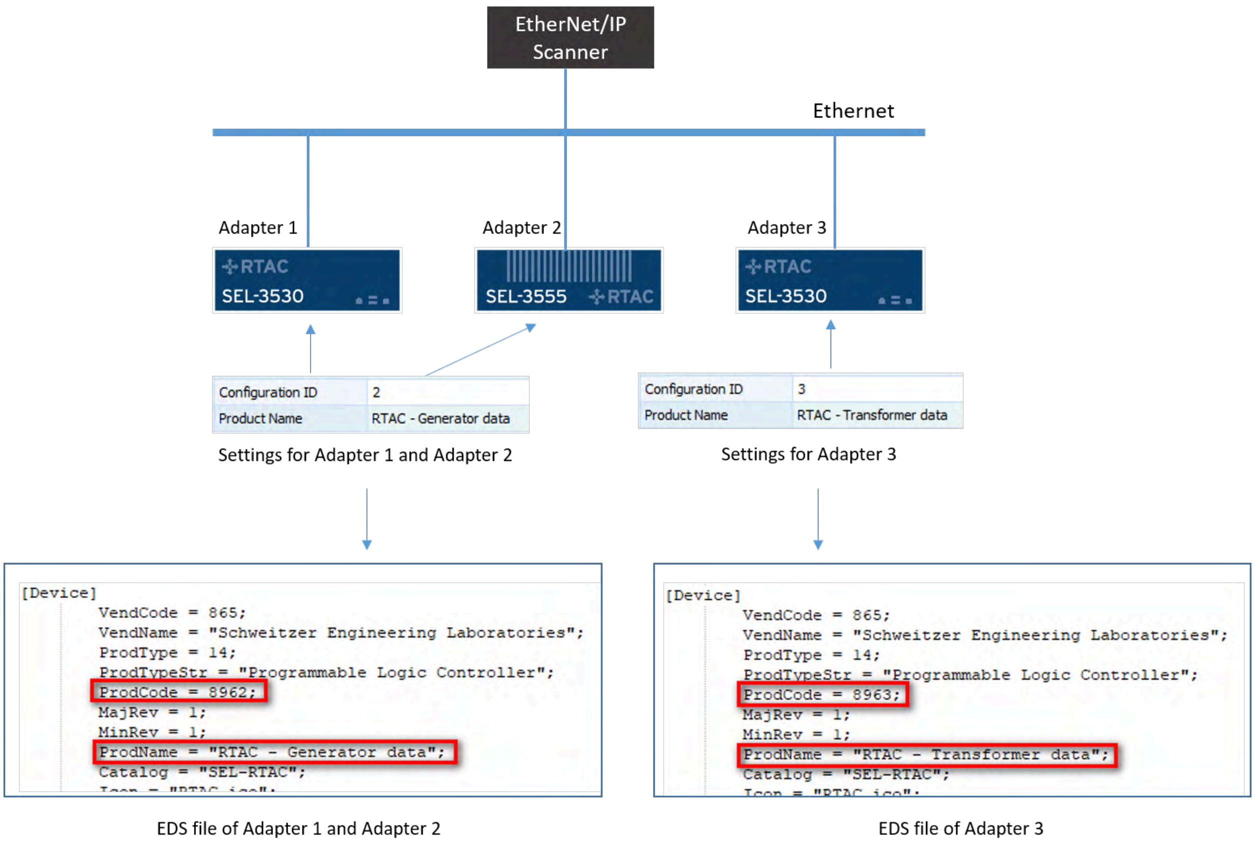


Figure 2.211 Multiple EtherNet/IP Configurations in a Single Network

Implicit Message Adapter Configuration

Step 1. Add an EtherNet/IP device into the project.

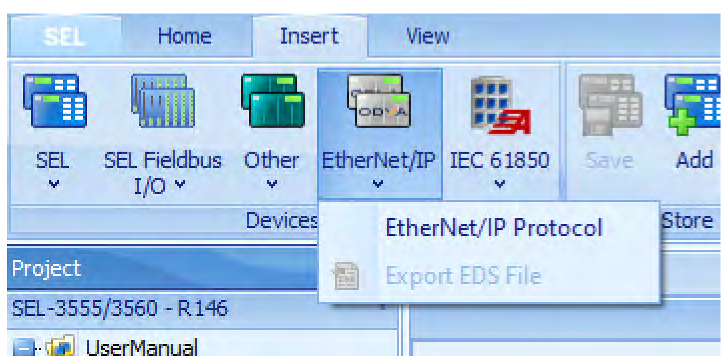


Figure 2.212 Add EtherNet/IP Into Configuration

Step 2. Enter the settings based on the requirements of your project.

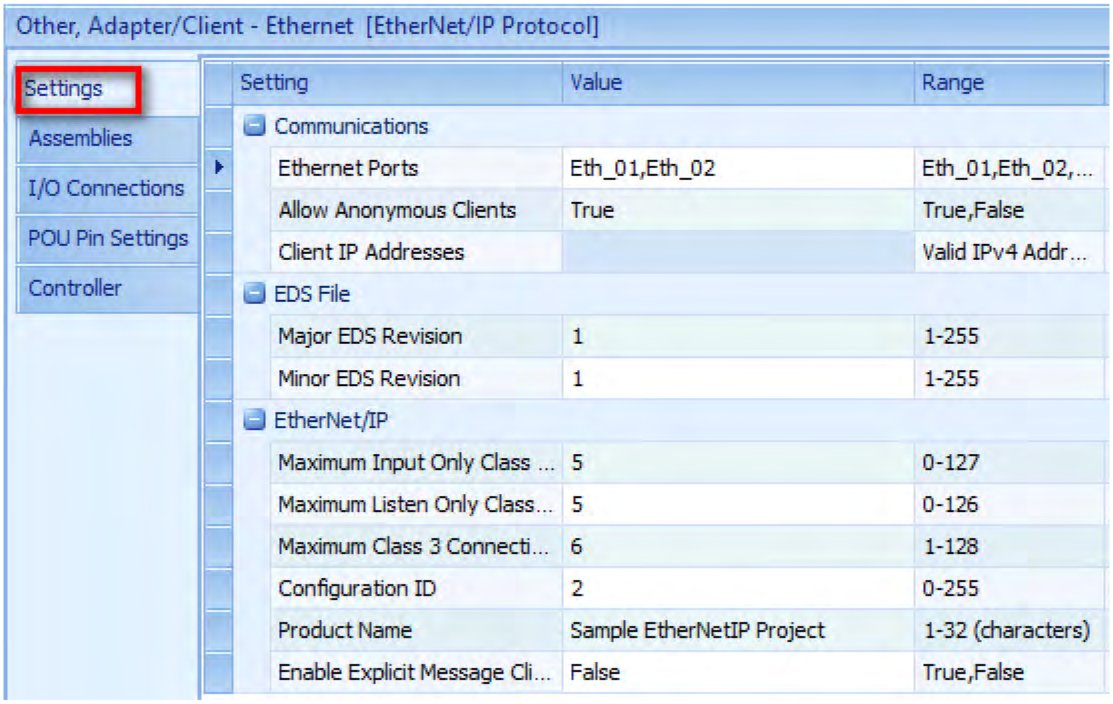


Figure 2.213 Configure EtherNet/IP Settings

Step 3. Create the Assembly object instances.

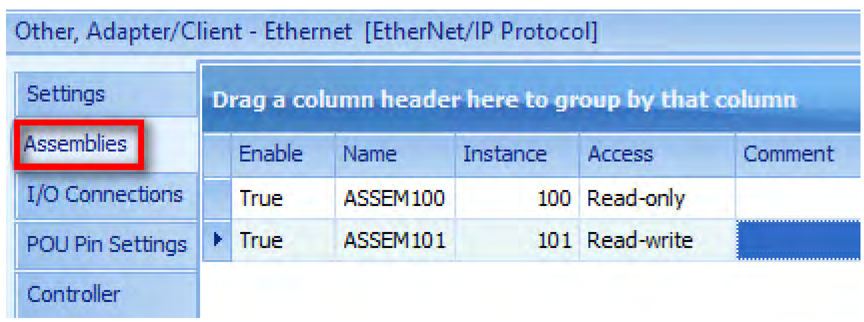


Figure 2.214 Configure Assembly Object Instances

Assembly instances that will serve as output assemblies shall be set with read-write access.

Step 4. Click on the Assembly instance sub-item to configure and map the Assembly members.

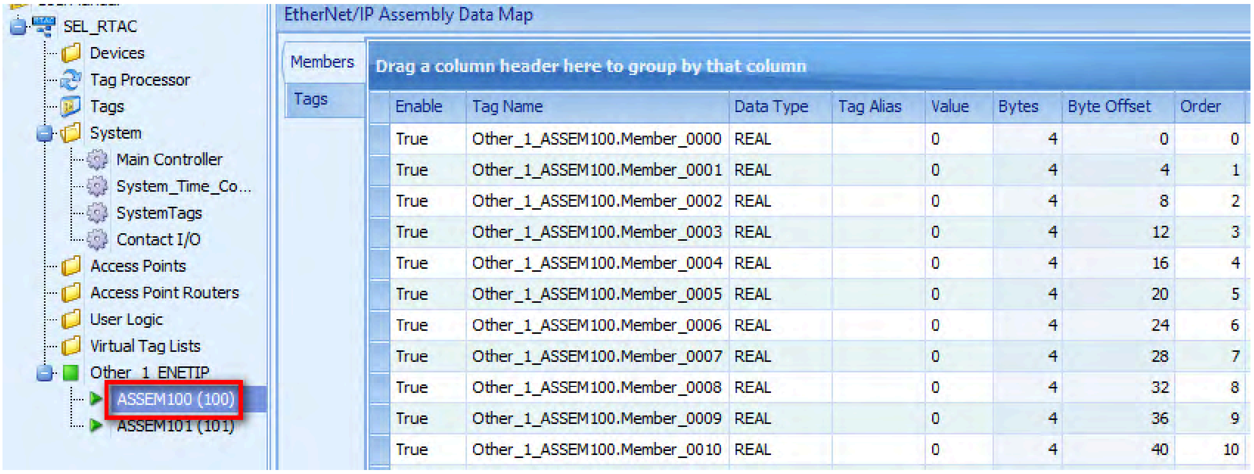


Figure 2.215 Configure Assembly Members

Step 5. Configure the I/O connection points.

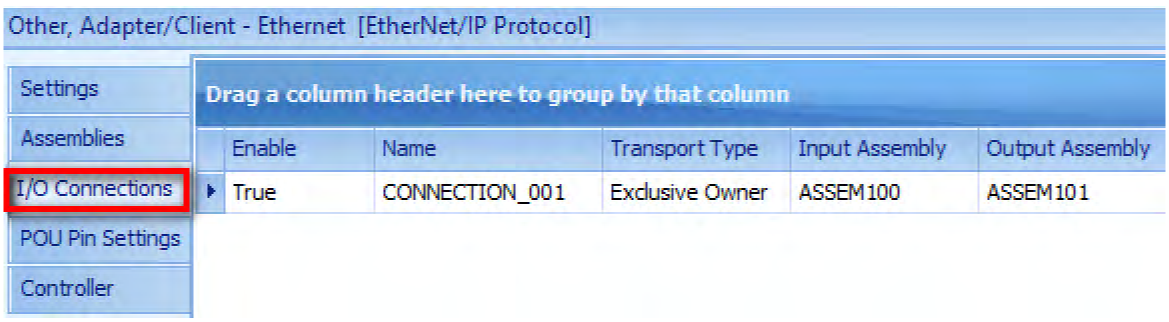


Figure 2.216 Configure I/O Points

Step 6. Export the EDS file and load it into the EtherNet/IP scanner configuration tool. Load the project into the RTAC and then configure the EtherNet/IP scanner.

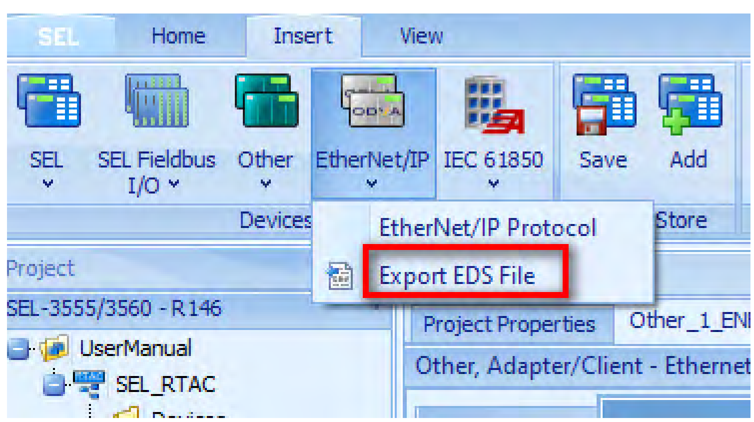


Figure 2.217 Export EDS File

Alternatively, the EDS file may be loaded by the EtherNet/IP scanner configuration tool directly from the RTAC after loading the project.

Explicit Message Adapter Configuration

Step 1. Enable Explicit Message Client.

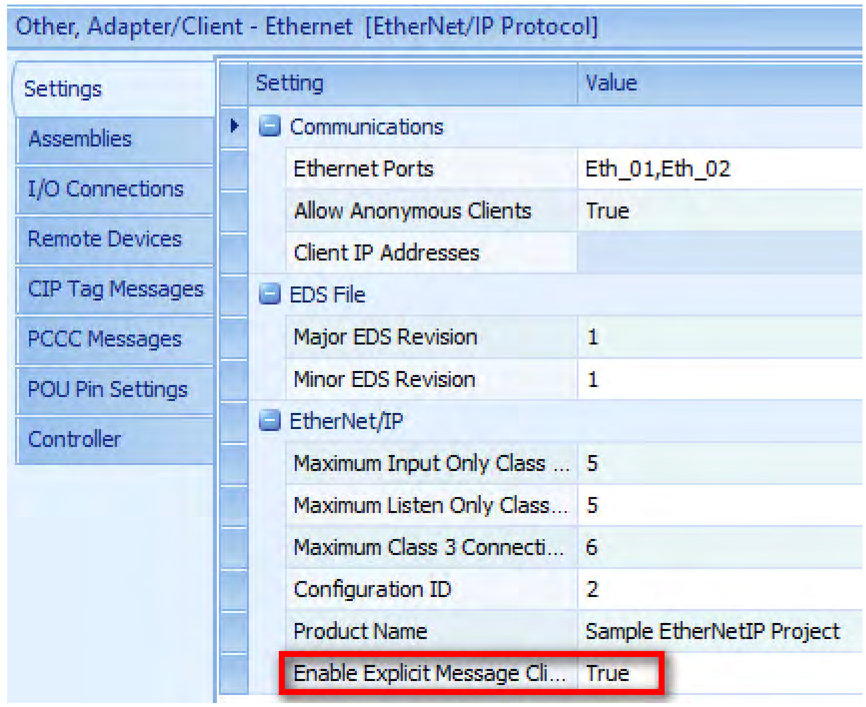


Figure 2.218 Enable Explicit Message Client

Step 2. Enter the list of Explicit Message server devices.

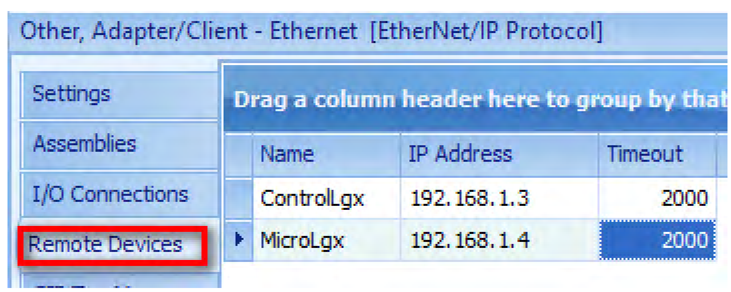


Figure 2.219 Enter Explicit Message List

Step 3. Configure the Explicit Messages.

- ➤ **CIP Tag Messages** tab: Use this tab to configure messages to read or write arrays of data in a remote controller that supports the CIP Data Table Read and CIP Data Table Write services. In the following figure, the message reads the first 5 elements of the tag array MotorTemperature defined in the remote PLC.

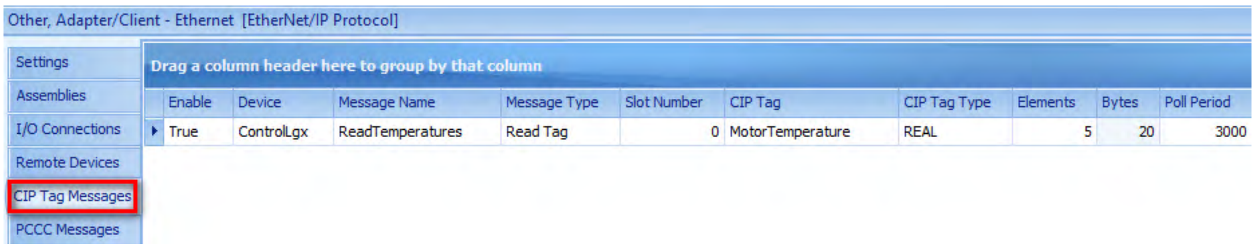


Figure 2.220 CIP Tag Messages

- ➤ **PCCC Messages** tab: Use this tab to configure messages to read or write arrays of data in a remote controller that supports the SLC Protected Typed Logical Read/Write services. In the following figure, the message writes in the data points N7:10, N7:11, and N7:12 of the remote PLC.

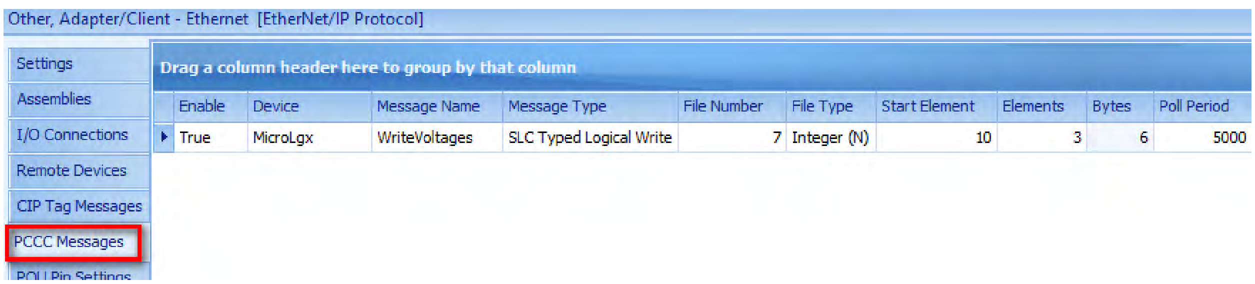


Figure 2.221 PCCC Messages

Step 4. Map the data. Tag arrays will be created as POU input pins (in case of write messages) or POU output pins (in case of read messages) to map the data that is written to or read from the remote PLC. These POU pins have the following format: [Device Name]_[Message Name]_Data.

Electronic Data Sheet (EDS) File

User-Configurable EDS Entries

The following EDS entries are based on ACSELERATOR RTAC user settings.

- ➤ EDS Revision—The value of the Revision entry in the File section of the EDS file is set by the Major EDS Revision and Minor EDS Revision settings in ACSELERATOR RTAC.
- ➤ Product Code—The ProdCode field in the Device section of the EDS file is mapped from the Product Code attribute of the Identity object.
- ➤ Product Name—The ProdName field in the Device section of the EDS file is mapped from the Product Code attribute of the Identity object.

- Parameter—The ParamN entries in the Parameter section of the EDS file correspond to the tags mapped into Assembly object instances in ACSELERATOR RTAC.
- Assembly—The AssemN entries in the Assembly section of the EDS file correspond to the Assembly instances defined in the **Assembly Instances** tab in ACSELERATOR RTAC.
- Connection—The ConnectionN entries in the Connection Manager section of the EDS file correspond to the implicit connections defined in the **I/O Connections** tab of ACSELERATOR RTAC.

Comments entered in Assembly data maps and the **I/O Connections** tab in ACSELERATOR RTAC are mapped to the EDS file as line comments. ACSELERATOR RTAC accepts UTF-8 characters in comment fields. Non-8-bit characters are replaced with a question mark ("?") in the EDS file.

EDS File Export

There are three methods by which to export the EDS file:

- ACSELERATOR RTAC software
- FTP/SFTP Server
- CIP File Object

Export EDS File Via ACSELERATOR RTAC

The EDS file can be exported by using the **Export EDS File** menu option under the **EtherNet/IP** ribbon menu.

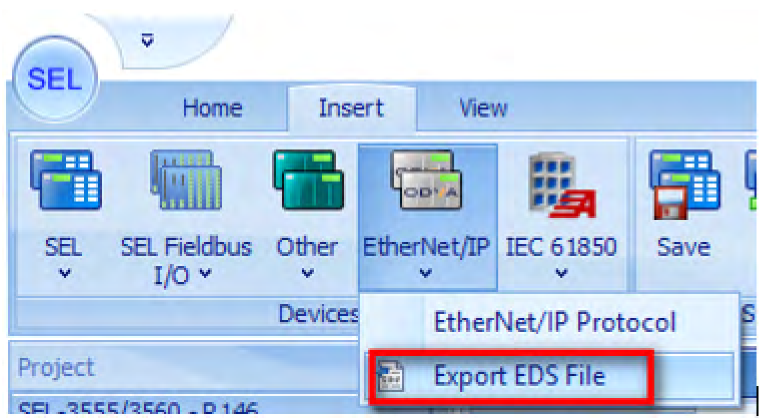


Figure 2.222 Export EDS File Via ACSELERATOR RTAC

Export EDS File Via FTP/SFTP

If an SFTP server is added to the RTAC, the EDS file can be downloaded from the ETHERNETIP folder in the SFTP root directory.

Export EDS File Via CIP File Object

The EDS file is stored in the File object (Object Class 0x37, Instance 0xC8) implemented as part of the EtherNet/IP object model in the RTAC. An EtherNet/IP scanner can retrieve the EDS file directly from the RTAC by using the File object services.

This page intentionally left blank

SECTION 3

Tag Processor

Overview

Use the Tag Processor to log tag values, enable tag value visualization in the RTAC web interface, and map data values from sources to destinations. Source data typically enter the SEL Real-Time Automation Controller (RTAC) from an IED or from a global variable. Destination data are typically data leaving the RTAC to go to SCADA, an HMI, or some other remote polling device. A Source Expression can be a simple data tag (such as Relay1_ DNP.AI_0000) or an IEC 61131-3 expression. The Tag Processor moves data values from Source Expressions on the right into Destination Tags on the left in the same row of the Tag Processor. Data mapping in the Tag Processor does not dictate data position or order in a given protocol or application. The RTAC completes all assignments in the Tag Processor before passing the data to the protocols. You can use the Tag Processor to map any tag and any attribute of a tag in the ACSELERATOR RTAC SEL-5033 Software project.

Use the Tag Processor grid for the following example tasks:

- Simple mapping of tags from IED protocols to remote polling protocols
- Mapping attributes of different source tags to a destination tag
- Simple calculations such as analog scaling
- Complex IEC 61131-3 calculations
- Conditional statements using IEC 61131-3 logic expressions
- Logging of changes in tag values (the first several tags come preconfigured for logging)
- Configuring real-time tag value visualization in the RTAC web interface

Tag Processor Data Entry

Although the syntax is not the same, data entry in the Tag Processor grid is similar to data entry in a spreadsheet. Use one of the following methods to enter tag names into the Tag Processor.

- Type the name of the point. Note that the autocompletion feature will display tags by matching against what you have typed. Begin with the device name followed by the type, and use periods (.) between each tag component (e.g., Relay_1_DNP.AI_0000.instMag).
  - Relay_1_DNP = Client device name
  - AI_0000 = First analog input point
  - instMag = Instantaneous value for this point (see *Data Types on page 747* for details)
- Highlight and copy the columns of tag names from previously configured client and server connections and paste these tag names into the **Source Expression** and **Destination Tag Name** columns, respectively. Note that pasting a group of tags in the Tag Processor will add new rows automatically in the grid.
- Copy and paste directly from device documentation by highlighting and copying a column from an Microsoft Excel spreadsheet, text file, or other tabular column of names, and then pasting this information into the appropriate **Tag Processor** column.
- Create the RTAC database mapping from a SCADA system database. Populate a spreadsheet with SCADA and IED tag names as mapped in the SCADA database. Copy the two columns of tags and paste these columns directly into the **Source Expression** and **Destination Tag Name** columns in the Tag Processor.

Press the icon to save and compile the project. The error icon () typically indicates a syntax error or an undefined destination tag. The alert symbol () typically indicates an IEC 61131-3 logic error on that line. Compile errors appear in the output window at the bottom of the Tag Processor screen. Double-click on errors to display the error location. The save operation generates code that you can view by clicking the **Code** tab at the bottom of the Tag Processor. ACSELERATOR RTAC generates this code automatically each time you save the project. This code serves as a noneditable debugging aid.

Notice the Tag Processor POU capacity gauge at the top of the Tag Processor screen. This gauge indicates how close you are to exceeding the maximum recommended amount of generated code. The maximum allowed number of assignment rows in the Tag Processor is 10,000, but the lines of generated code will vary depending on what assignments are configured. A reading of 100 percent on the Tag Processor POU capacity gauge indicates ACSELERATOR RTAC has generated 300,000 characters of structured text to perform the configured mapping. Exceeding 100 percent could cause the application to run out of memory on your PC. You can reduce memory usage and still retain the same functionality by copying generated code and placing it in an IEC 61131-3 custom logic program. Then, delete the assignments from the Tag Processor to clear those lines of code.

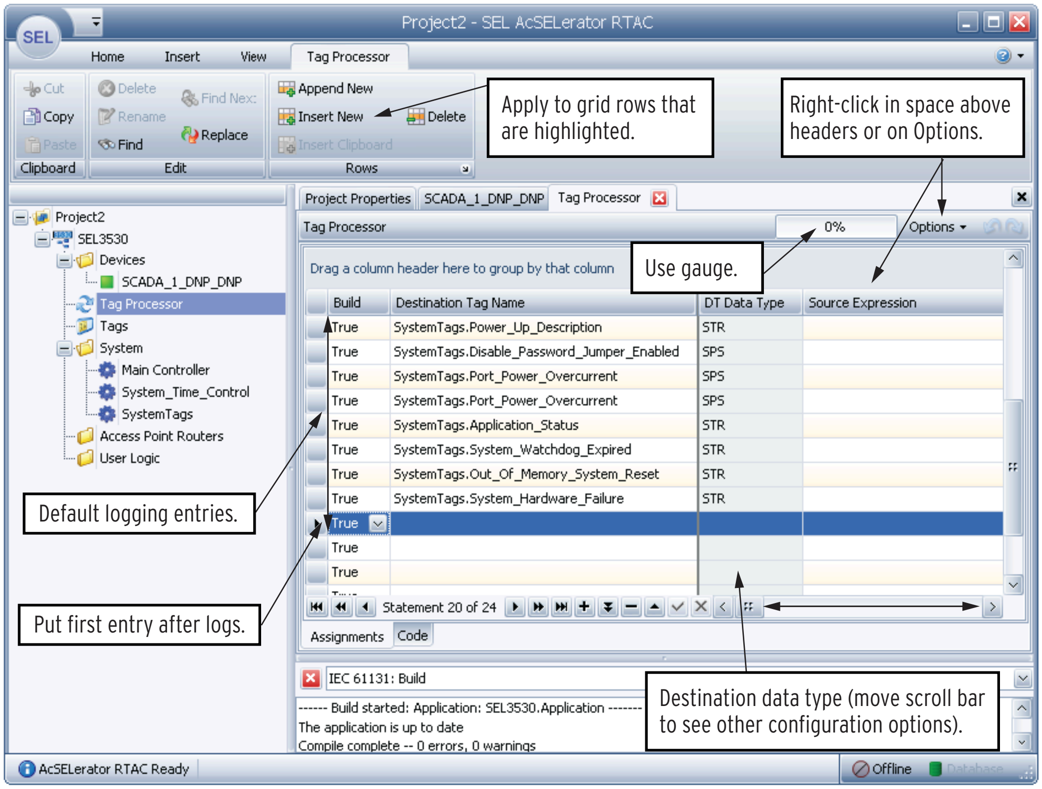


Figure 3.1 Tag Processor Grid

The two main columns in the Tag Processor are Destination Tag Name and Source Expression. View other columns by sliding the horizontal scroll bar.

Customize your view of the Tag Processor by right-clicking in space above headers or by selecting **Options** and then selecting the following:

- **Columns**: Show/hide all configuration parameter columns.
- **Run Time Column Customization**: Configure which columns appear and in which order.
- **Various Layouts**: Apply filters to view only certain data rows. Defaults include analog, digital, show all, and logging layout filters. Create and save custom layout filters as necessary. Select Default Layout to undo any changes you have made and to display all columns.
- **Export to XLS**: Export visible columns to an XLS (Excel spreadsheet) format.
- **Export to XML**: Export visible columns to an XML format.
- **Auto-Fill Time and Quality Source**: When a tag has a status, quality, and time stamp in the destination column and only a status mapped in the source column, this option automatically populates the quality source and time source columns. This makes it so that the destination tag will receive status updates with corresponding time stamps and good quality.

- **Treat Code not Generated as Error**: Tag processor statements that do not generate a valid mapping or logic statement indicate this with an exclamation mark on the left side of the tag processor row. Normally, this warning does not produce an IEC 61131 compile error and the project can be sent to the RTAC as-is. When this option is selected, these warnings generate compile errors that must be resolved before sending the project to the RTAC.
- Right-click on any column header to do the following:

NOTE
By default, the first several rows in the table are assigned to system tags for logging. You can delete these default logging assignments, but SEL recommends keeping them for NERC CIP requirements.

  - **Sort Ascending**: Sort Tag Processor by this column from low to high (A–Z, 0–9).
  - **Sort Descending**: Sort Tag Processor by this column from high to low (Z–A, 9–0).
  - **Clear Sorting**: Clear active sort options (return to default).
  - **Group By This Column**: Group data by this column type rather than by tag name.
  - **Group by Box**: Group data by this box rather than by tag name.
  - **Column Chooser**: Organize the order and which columns to view.
  - **Best Fit**: Adjust automatically the width of a column you select to the largest size entry.
  - **Clear Filter**: Clear active filter.
  - **Filter Editor**: Provide logic to implement a filter on tag names or other point attributes.
  - **Best Fit (all columns)**: Adjust automatically the width of all columns to accommodate largest entry sizes.
  - **Select Column**: Highlight an entire column for copy/paste, etc., operations.

Table 3.1 Available Columns in the Tag Processor

Parameter	Description	Default	Field Type
Build	Determine if the Tag Processor will evaluate this row.	True	Constant (BOOL)
Destination Tag Name	Any tag name in the system that can accept a value mapped to it.		Any tag name (operAPC, BCR, STR, STRING, TIM, TIME, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, MV, CMV, operINC, INS, SPS, operSPC)
DT Data Type	Data type of the destination tag.	Supplied by system lookup	Not assignable. Generated automatically.
DT Bit Reference	The bit in the Destination Tag that the Tag Processor will replace with the value of the Source Expression.		Constant (0–31)

Parameter	Description	Default	Field Type
Source Expression	Any tag in the database. Can also contain any structured text with tag(s).		Any tag name or expression (operAPC, BCR, STR, STRING, TIM, TIME, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, MV, CMV, operINC, INS, SPS, operSPC). May also use instructions if not associated with a Destination Tag.
SE Data Type	Source Expression data type or global variable definitions (see *Example 3.4*).	Supplied by system lookup	Generated automatically
Live Data Enable	Set this to Viewable to enable visibility of the Destination Tag on the Live Data page in the RTAC web interface. Set to Forceable to allow the tag to be forced from the Live Data page. Live Data supports a wide variety of tag and variable types. See *Live Data on page 449* for more information.	FALSE	Constant (BOOL)
Live Data Labels	Provide a comma-separated list of labels (as many as 255 characters) to associate with this tag in Live Data. Live Data can be filtered by labels to make locating data faster. See *Live Data on page 449* for more information.		Constant (STRING(255))
SE Bit Reference	The bit in the source expression that the Tag Processor will assign to the destination tag.		Constant (0–31)
Time Source	The time stamp the Tag Processor assigns to the Destination Tag. A time stamp from any tag can be used, and this overrides the time stamp of the Source Expression tag. Example values: System_Time or SEL_421_1_DNP.BI_0001.t. Using a different time source when mapping non control types to control types may cause incorrect control behavior.		Variable (timestamp_t)
Time Offset	ACSELERATOR RTAC will add the value in this field to the Destination Tag time-stamp value. The format is T#*n*s, where *n* is any whole number.		Constant (TIME) Variable (TIME)
Quality Source	The quality value the Tag Processor assigns to the Destination Tag. A quality source from any tag can be used, and this overrides the quality of the source expression tag. Example value: SEL_421_1_DNP.BI_0001.q.		Variable (quality_t)
Instruction Number	Noneditable reference value linked to the structured text shown under the **Code** tab at the bottom of the Tag Processor grid.	Unique Identifier	Not assignable. Generated automatically.
Solve Order	Configurable order for the solution of Tag Processor logic. Numbers do not need to be unique.	Follows instruction number	Constant (DINT)
Description	A user-editable field for applicable comments. Not used in the logic engine.		Constant (Any string)
Control Type	Editable only on control type tags.		Constant (Pulse, Persist)
Operator Blocked	Enter a tag or constant of type Boolean. The Tag Processor maps this value to the Destination Tag operatorBlocked attribute.		Constant (BOOL) Variable (BOOL)

Parameter	Description	Default	Field Type
On Pulse Dur	Editable only on control type tags. Length of on pulse in milliseconds.	1000	Constant (UDINT) Variable (UDINT)
Off Pulse Dur	Editable only on control type tags. Length of off pulse in milliseconds.	1000	Constant (UDINT) Variable (UDINT)
Number of Pulses	Editable only on control type tags.	1	Constant (USINT) Variable (USINT)
PU Delay	Pick-up delay.		Constant (Between T#10ms and T#1000000s) Variable (TIME)
DO Delay	Drop-out delay.		Constant (Between T#10ms and T#1000000s) Variable (TIME)
Dead Band	Editable only on analog tags of type MV or CMV. The number of units that dictates a significant value change. The excursion of the dead band moves instMag to mag.		Constant (REAL) Variable (REAL)
Zero Dead Band	Editable only on analog tags of type MV or CMV. If Mag < Zero Dead Band, then Mag := 0.		Constant (REAL) Variable (REAL)
Range Min	User-defined threshold value.		Constant (REAL or DINT) Variable (REAL or DINT)
Range Max	User-defined threshold value.		Constant (REAL or DINT) Variable (REAL or DINT)
Range Low Low Lim	User-defined threshold value.		Constant (REAL or DINT) Variable (REAL or DINT)
Range Low Lim	User-defined threshold value.		Constant (REAL or DINT) Variable (REAL or DINT)
Range High Lim	User-defined threshold value.		Constant (REAL or DINT) Variable (REAL or DINT)
Range High High Lim	User-defined threshold value.		Constant (REAL or DINT) Variable (REAL or DINT)
Logging Enable	Set to True to enable the logic engine logging of this tag.	True for default logged points. False for all others.	Constant (BOOL)
Logging - Priority	User-defined priority for log item filtering.		Constant (STRING(255)) Variable (STRING(255))
Logging - Category	User-defined category for log item filtering.		Constant (STRING(255)) Variable (STRING(255))
Log Initial State	Set to True to log startup value of tags. If False, the RTAC system will wait to log changes until after the system reaches a ready state.	False	Constant (BOOL)
Logging - On Message	User-configurable message that logic engine enters into the log when the tag value asserts.		Constant (STRING(255)) Variable (STRING(255))
Logging - Off Message	User-configurable message that logic engine enters into the log when the tag value deasserts.		Constant (STRING(255)) Variable (STRING(255))
Logging - Time Change Trigger	If True, the logic engine will log the tag upon a change in the tags time stamp.		Constant (BOOL) Variable (BOOL)

Parameter	Description	Default	Field Type
Logging Time Change Trigger Message	The text that the logic engine logs when the time stamp of the destination tag changes. The RTAC will use this message if no Logging-On or Logging-Off messages are defined. Applies to types SPS, MV, CMV, INS, and STR.		Constant (STRING(255)) Variable (STRING(255))
Logging Alarm Enable	When true, the logic engine considers the logged value an alarm. These tags are logged in the SOE log but also appear in the Alarm Summary of the web interface. The alarm entry is red when in alarm, then returns to black when no longer in alarm state. Applies to types SPS, MV, CMV, INS, and STR.	False	Constant (BOOL) Variable (BOOL)
Logging Alarm State	If True, the logged item will go into alarm when state changes from deasserted to asserted. Applicable only when Logging Alarm Enable is True.		Constant (BOOL) Variable (BOOL)
Logging Chatter Count	Number of changes allowed within the Chatter Time before the logic engine removes a tag's log items because of chattering.		Constant (UINT)
Logging Chatter Time	The time window in which tag value changes are counted to determine a chattering condition.		Constant (TIME)
Logging Comment	User-defined comment.		Constant (STRING(255)) Variable (STRING(255))

Example 3.1 Simple Source-to-Destination Tag Mapping

The following example illustrates simple tag mapping between a relay and a SCADA system using the DNP3 protocol. For more information regarding how to add and configure DNP client and server devices, see *Section 2: Communications*.

Step 1. Insert a DNP device and configure it as a server. Name it **SCADA_1**.

Step 2. Insert a DNP shared map and name it **SCADA_1**. Configure the DNP server device to use this shared map.

Step 3. Configure DNP shared map SCADA_1 with two binary inputs, two analog inputs, and one binary output.

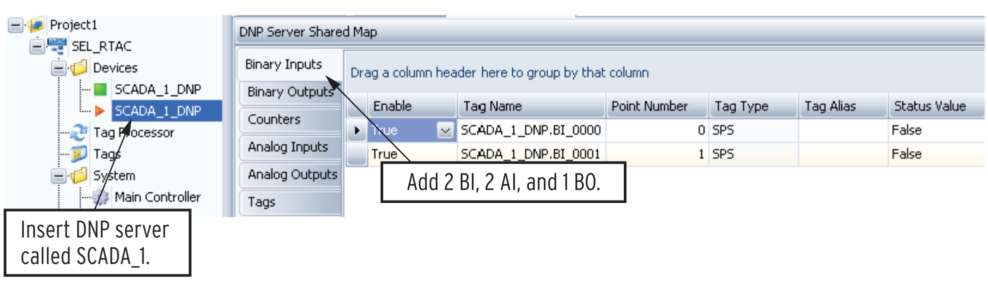


Figure 3.2 Insert DNP Server

Step 4. Insert a DNP device, and configure it as a client. Name it **Feeder_1**.

Step 5. Configure Feeder_1 with two binary inputs, two analog inputs, and one binary output.

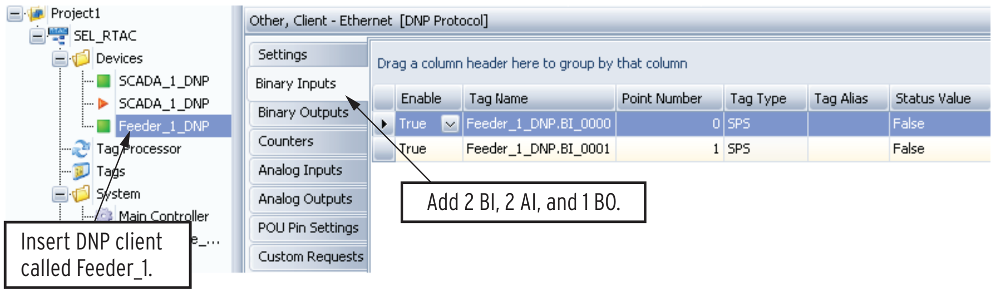


Figure 3.3 Insert DNP Client

Step 6. Click on the **Tag Processor** tab from the list view in the left pane of the screen.

Step 7. Click on + to add the total number of SCADA_1 tags. In this example, that number will be 10.

NOTE

The Tag Processor inserts new rows automatically if the number of tags you paste exceeds the number of available rows.

Step 8. Populate the **Destination Tag Name** column starting on the first available row with the two binary input tags from SCADA_1. Either type in the tags directly into the Tag Processor grid or select/copy/paste from the SCADA_1 tag lists. See *Section 3: Tag Processor*.

Step 9. Populate the **Destination Tag Name** column starting on the first available row after the binary inputs with the two analog input tags from SCADA_1.

Step 10. Populate the **Destination Tag Name** column starting on the first available row after the analog inputs with the binary output status from SCADA_1. See *Figure 3.5*.

Step 11. Finish populating the destination column with the BO structure from Feeder_- 1, as in *Figure 3.5*.

The source column is for data entering the RTAC, and the destination column is for data exiting the RTAC, so control output mapping is the reverse of input mapping in the Tag Processor. Data enter the RTAC from Feeder_1 and exit via the SCADA_1 tags. Control data, however, enter the RTAC from SCADA_1 and exit via the Feeder_1 tag. Data quality for destination analog, status, and counter tags also comes from the source or IED tag. Data quality for control values, however, is provided by the server tags and transferred to the client tags. This infers that control data tags from an IED will inherit invalid quality if the server connection is not working even if the client connection has no problems. Note that one element of the control structure (the control status) is an input the RTAC provides, and that mapping is similar to any other status input. See *Figure 3.6*.

Note also that DNP binary outputs have special structures to facilitate different control types as well as to enable mapping of one control type to another. To forward to Feeder_1 all binary output commands the RTAC receives from SCADA_1, map the entire control output type structure. This ensures that DNP commands will pass transparently to the IED as the RTAC receives these commands from SCADA_1. Were you to map a control from Modbus, there would be only one entry in the Tag Processor per control point, because Modbus lacks the complex control structure of DNP.

Step 12. Populate the Source Expression columns with Feeder_1 inputs and outputs as in *Figure 3.5*.

In some cases, you do not have to map all DNP BO control types. For example, the DNP server may always receive Trip and Close commands and the IED may only need Trip and Close commands. In that case, only map the Trip and Close control types.

Some situations require a conversion from one DNP BO type to another. For example, the polling master may only be able to send trip/close commands, but the IED can only accept latch on/off commands. Map the different DNP BO types to each other in the Tag Processor. Change the Control Type column to force the source control type to match the destination control type. In this example, change Control Type to Persist to match that of the Latch controls in the Destination Tag Name column. See *Figure 3.4*. When SCADA_1 sends a trip command to BO_0000, the RTAC will issue a DNP message with the BO LatchOff command to the configured point on Feeder_1.

NOTE

Control types of Source Expressions must match those of the Destination Tag Names to which they are assigned. To force a match, change the type in the Control Type column.

Destination Tag Name	DT Data Type	Source Expression	Control Type
Feeder_1_DNP.BO_0000.operLatchOn	OPERSPC	SCADA_1_DNP.BO_0000.operClose	Persist
Feeder_1_DNP.BO_0000.operLatchOff	OPERSPC	SCADA_1_DNP.BO_0000.operTrip	Persist

Figure 3.4 DNP BO Type Conversion

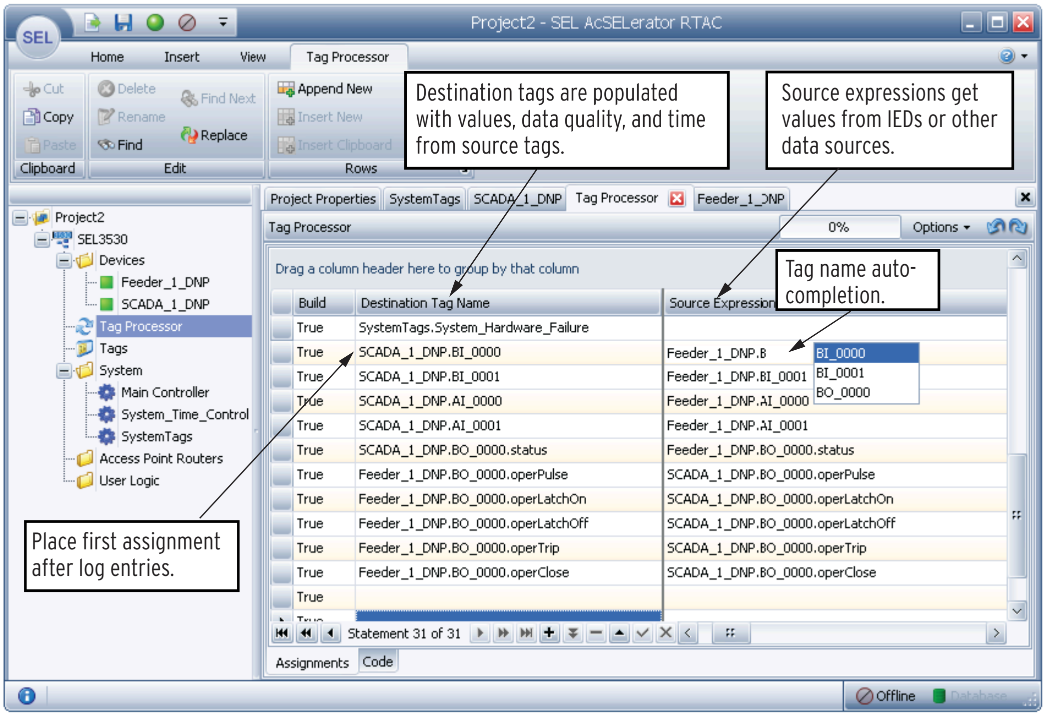


Figure 3.5 Simple Tag Mapping

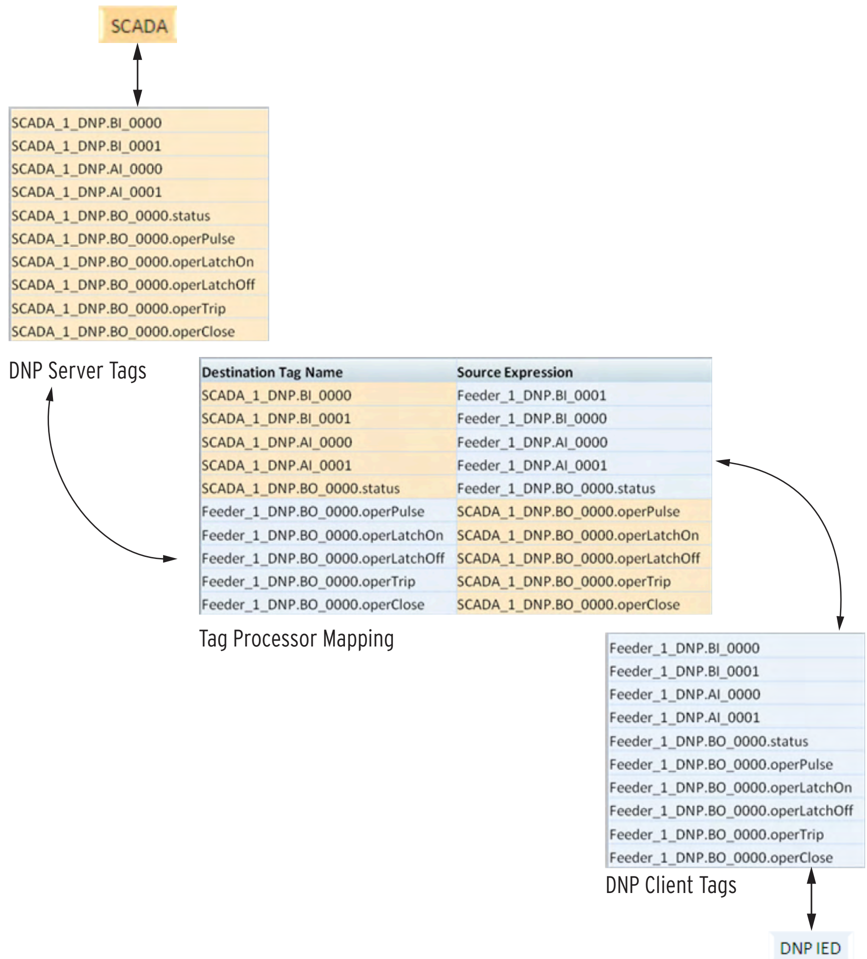


Figure 3.6 Tag Processor Data Flow

Example 3.2 IEC 61131-3 Code in the Tag Processor

Use any structured text statements in the Tag Processor to modify values or select conditionally which tags to use. This example illustrates the following Tag Processor features:

- Conditional statements that you can use to create a primary/backup IED relay.
- A SELECT statement that you can use to choose automatically between two tag values.
- Boolean logic that you can use on two binary input points.
- A complex calculation to create a derived value.

Step 1. Start with the project in *Example 3.1*.

Step 2. Add two more binary input and two analog input tags to SCADA_1.

Step 3. Insert a DNP device and configure the device as a client. Name it **Line_1**. This will represent a transmission line protective relay.

Step 4. Add two binary and two analog inputs to Line_1.

Step 5. Copy and paste Line_1. Rename **Line_1copy** to **Line_1_Bak**. This will represent the backup relay on Transmission Line 1. See *Figure 3.7*.

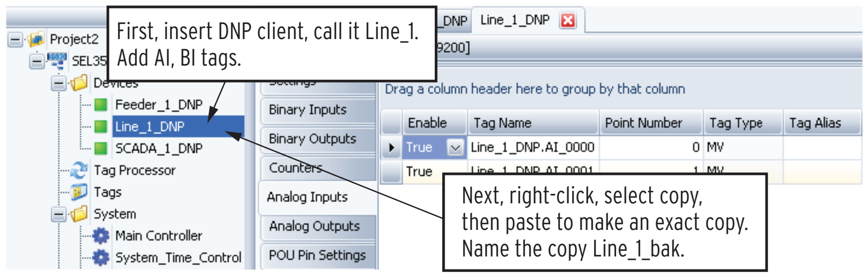


Figure 3.7 Insert Primary and Backup IED

Step 6. Highlight the entries in the Tag Processor from the previous example, and click on the − icon to delete the entries.

Step 7. Click the + icon to add lines to the Tag Processor as necessary.

Step 8. Type in or copy/paste the tags from Feeder_1, Line_1, and Line_1_Bak to the Tag Processor as in *Figure 3.8*.

Step 9. Type in or copy/paste the tags from SCADA_1 to the Tag Processor as in *Figure 3.8*.

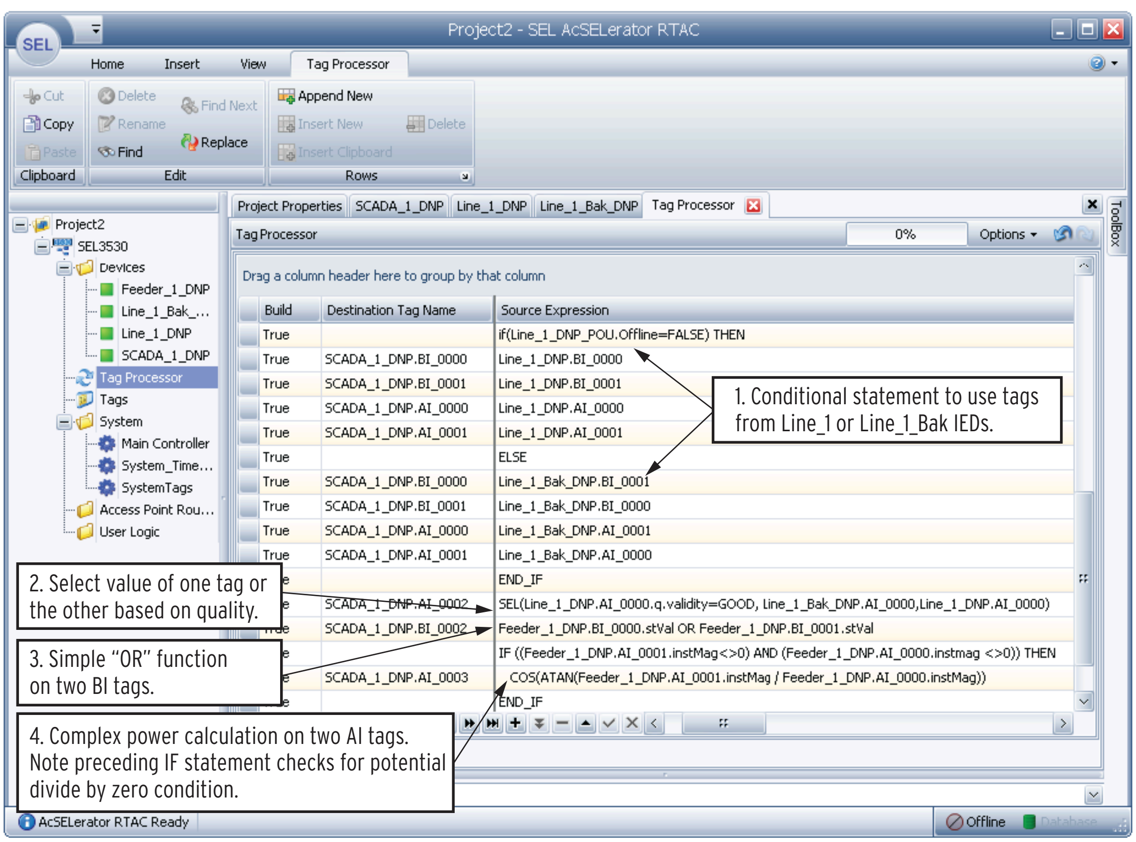


Figure 3.8 IEC 61131-3 Logic in Tag Processor

The following describes the specific syntax of the embedded IEC 61131-3 structured text in *Figure 3.8*:

1. The IF statement is checking to ensure Line_1 is not offline. If there is an unrecoverable communication error, the Offline POU pin will become True. As long as the Line_1 Offline POU pin = FALSE, Line_1 data will populate SCADA_1 tags. If Line_1 Offline = True, the backup relay (Line_1_Bak) tags will populate SCADA_! tags.
2. The RTAC uses the **SELECT** command to send the value of one tag or the other to the SCADA_1 tag. If the expression "Line_1_DNP.AI_0000.q.validity=GOOD" is true, then the value of the Line_1 tag maps to the SCADA_1 tag. Otherwise (if the quality is not good for that point), the RTAC uses the Line_1_Bak tag. This is a good feature to use if you have only one or two values that are acting as a primary and backup.

NOTE
If you use IEC 61131-3 logic in the Tag Processor, review the generated structured text in the Code tab after you save the project. Verify that the code the Tag Processor generates matches your intentions and look for any "invalid statement" warnings.

3. You can use any combination of Boolean logic operators as a source expression. In this example, SCADA_1_DNP.BI_0002 is true if either BI_0000 OR BI_0001 are true. By ORing the stval of the tag, you will lose the time and quality of the original two tags being ORed.
4. You can use any combination of math operations on any tags as a source expression. In this example, we illustrate how we can calculate power from the reactive and real power values of two analog input tags.

Example 3.3 Defining a Local Variable in the Tag Processor

This example will further demonstrate embedding IEC 61131-3 logic by defining a local variable directly in the Tag Processor.

Step 1. Start with any project in ACSELERATOR RTAC.

Step 2. Open the Tag Processor.

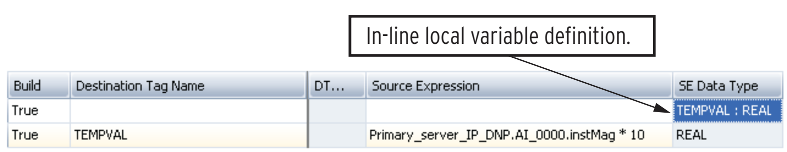
Step 3. In a blank source expression (SE) Data Type column, type **TEMPVAL : REAL**. You can use any supported data type when defining the new local variable. Variable names are not case sensitive.

Step 4. Type **TEMPVAL** in the Destination Tag Name column.

Step 5. Create an assignment in the Source Expression column. See *Figure 3.9*.

Step 6. Press **<Ctrl+S>** to save and compile the project.

You will notice there are no compile errors. The new variable is created as a global and recognized throughout the ACSELERATOR RTAC project.



Build	Destination Tag Name	DT...	Source Expression	SE Data Type
True				TEMPVAL : REAL
True	TEMPVAL		Primary_server_IP_DNP.AI_0000.instMag * 10	REAL

Figure 3.9 Local Variable Definition in Tag Processor

Example 3.4 Populating the Tag Processor From Excel

This example illustrates how we can use a preconfigured Excel spreadsheet to populate the Tag Processor. We presume in this example that the control center developed the spreadsheet with predefined names to which the RTAC must adhere.

Step 1. Start with a blank project in ACSELERATOR RTAC.

Step 2. See the spreadsheet in *Figure 3.10*. It has SCADA points and IED points as we recorded these points from the master control center configuration sheets (presumed for this example).

Step 3. Select, copy, and paste the rows from the spreadsheet. Note that the spreadsheet must have either a blank middle column for a bulk copy/paste, or you must copy the columns one at a time. Correct columns are as follows:

- Column 1 = Destination or SCADA tag names
- Column 2 = blank
- Column 3 = Source or IED tag names
- Column 4 = Optional RTAC data type for the source tag (see *Appendix B: IEC 61131-3 Programming Reference* for data types). Typically, it is best to let the system determine this value.

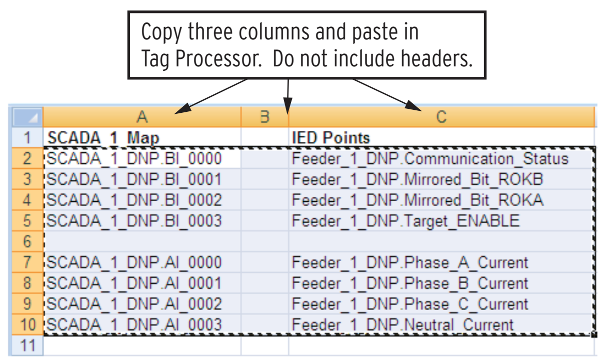


Figure 3.10 SCADA Mapping Spreadsheet

Step 4. Paste the columns you copied into the first destination tag name entry available. If no blank destination tag name entry exists, press + to create one. The Tag Processor will create and populate as many rows as you paste from the spreadsheet. You must select the entire cell for the paste operation to work correctly.

Step 5. The source and destination tags are now mapped, but these tags still do not exist in the ACSELERATOR RTAC database. Note in *Figure 3.11* that ACSELERATOR RTAC does not fill out the source and destination data types because the tags are not yet defined. A appears beside each undefined tag.

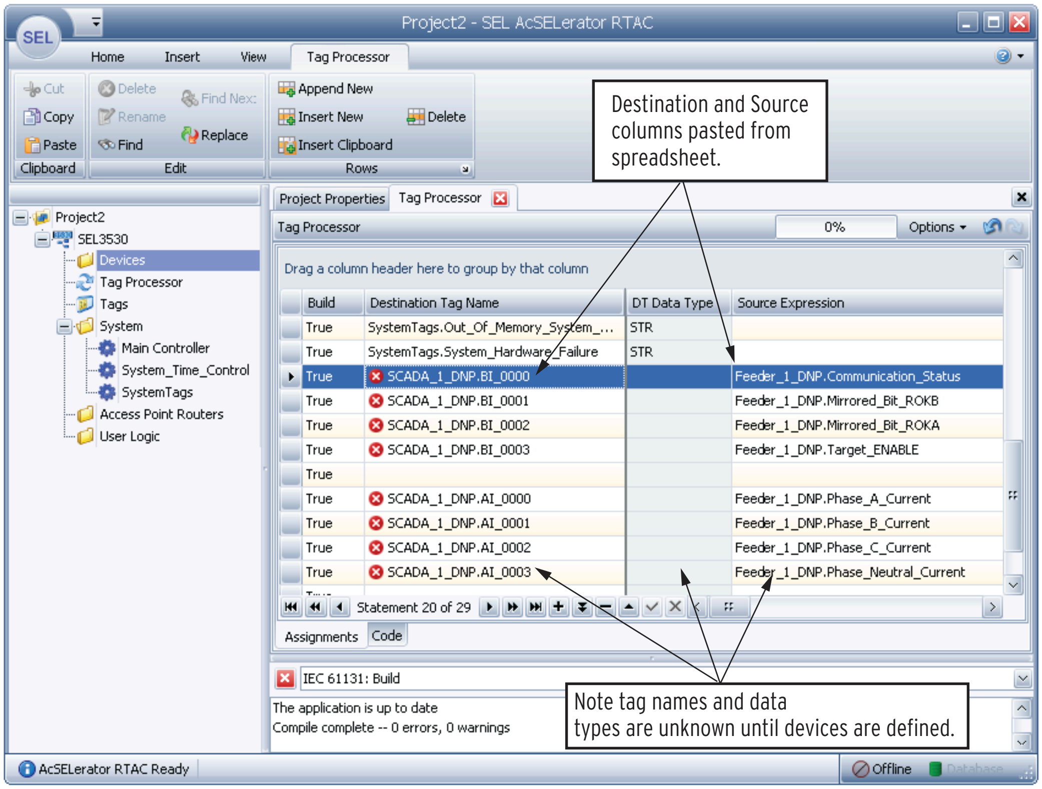


Figure 3.11 Copy/Paste Spreadsheet Tags

Create Destination Tags:

Step 6. From the menu ribbon, insert a DNP device and define this device as a DNP server. Name it **SCADA_1**.

Step 7. Add four binary input and four analog input tags. In this example, the names that ACSELERATOR RTAC generated automatically match those of the spreadsheet, so nothing more is necessary.

Create Source Tags:

Step 8. From the menu ribbon, insert a DNP device, and define this device as a DNP client. Name it **Feeder_1**.

Step 9. Add four binary input and four analog input tags.

In this example, the names that ACSELERATOR RTAC automatically generated do *not* match those of the spreadsheet.

Step 10. Copy the binary input names from the spreadsheet and paste these names on top of the binary input tags in the Feeder_1 device. See *Figure 3.12* and *Figure 3.13*. Alternatively, you can paste custom tag names into the Alias columns within each data type configuration page.

Step 11. Repeat *Step 10* for the analog input names in the spreadsheet.

Step 12. Press the **Tag Processor** tab to return to the Tag Processor view.

Step 13. Press or **<Ctrl+S>** to save. Note that data types are filled out correctly and that the symbols are gone.

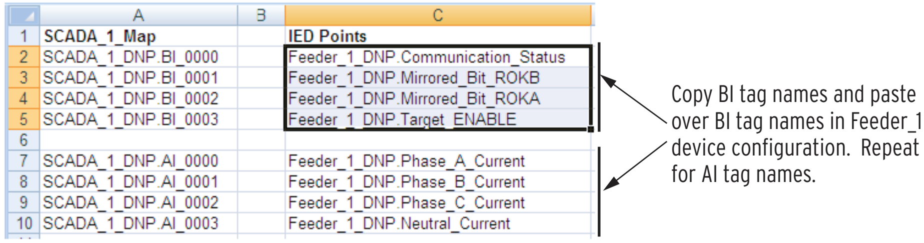


Figure 3.12 Custom Tag Names in Spreadsheet

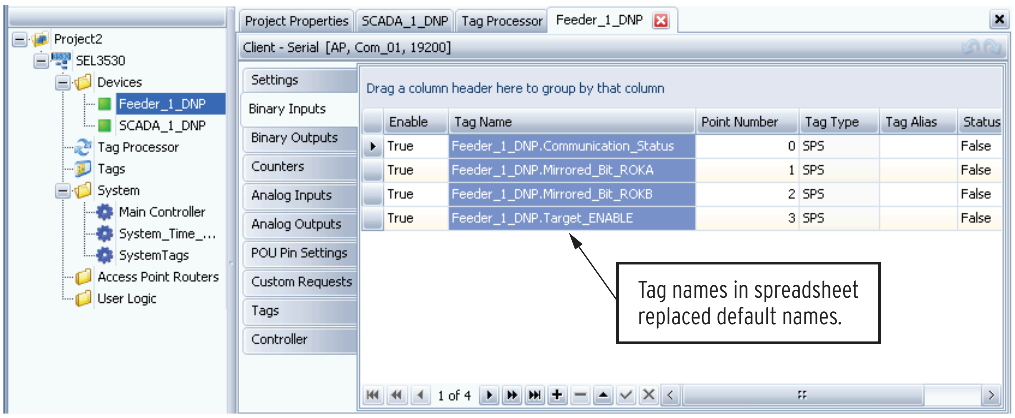


Figure 3.13 Feeder_1 Default Tag Names Replaced

Example 3.5 Access Individual Bits in the Tag Processor

This example illustrates the syntax to access individual bits from tags in the Tag Processor. This is an alternate method for packing/accessing individual bits into a tag and is valid in user logic as well as in the Tag Processor. Care should be taken when performing bit-wise operations. Note that floating point tags and signed integers are represented in memory differently than unsigned tags.

Figure 3.14 shows the syntax for accessing individual bits of a tag by placing a .0, .1, etc., after the tag name. While online with the project, click on **Tools > Display Mode > Binary** from the Home ribbon menu to see the binary representation of the tag (see *Figure 3.15*).

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.0	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.1	BOOL	FALSE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.2	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.3	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.4	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.5	BOOL	FALSE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.6	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.7	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.8	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.9	BOOL	FALSE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.10	BOOL	FALSE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.11	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.12	BOOL	FALSE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.13	BOOL	TRUE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.14	BOOL	FALSE	BOOL
Modbus_IED_MODBUS.HREG_00000.oper.ctlVal.15	BOOL	FALSE	BOOL

Figure 3.14 Bit Access in the Tag Processor

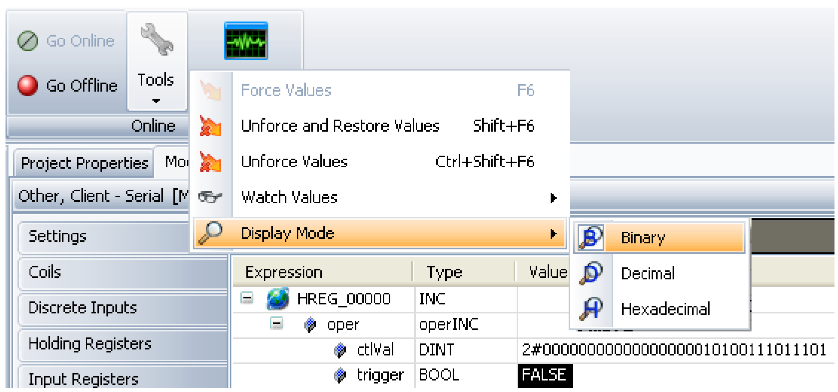


Figure 3.15 Binary Display Mode

Example 3.6 Visualize Tag Data on RTAC Web Interface

Step 1. Configure a valid assignment in the Tag Processor, including a Source Expression and Destination Tag Name.

Step 2. In the column Live Data Enabled, set the value = Viewable.

Step 3. Save the project and send to the RTAC.

Step 4. In the RTAC web interface, click the menu item Live Data.

The live data for the tags selected is displayed on the web interface.

Alias Tags

The format of TagNames is always as follows: devicename_protocol.pointname (i.e., Line_- 1_DNP.BI_0001). You can change the pointname and devicename, but the structure must remain the same. Alias tags do not follow this format, and you have greater flexibility in naming tags. Place user-created alias tags in the alias tags column in each device setup. Once you assign an alias name to a tag, it replaces that tag throughout the system. For example, use alias tags in the Tag Processor in place of device tag names. Use the following rules when creating alias tags:

- Make each alias tag a unique name within a project.
- Start with a letter.
- Use any combination of letters and numbers you want.
- Instead of spaces or dashes, use the underscore (_) character.
- Do not have two underscore characters together.
- Do not use special characters.

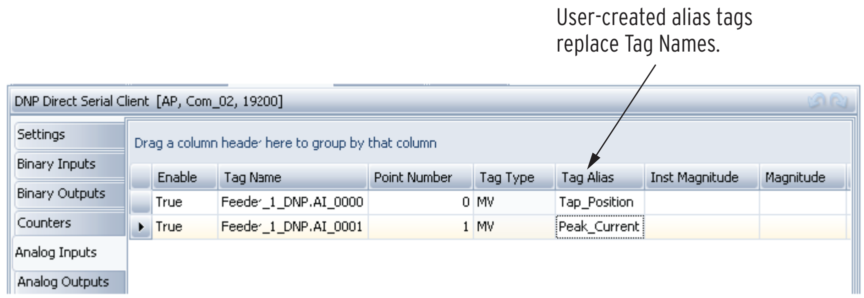


Figure 3.16 Alias Tags

Special Tag Processor Functions

Although you can perform 61131 logic on individual variables in the Tag Processor, most device tag types are structures of variables. The structures contain not only the present value of the tag but also time data, quality data, etc. For example, this statement adds two instantaneous analog values:

✓ Feeder_1_DNP.AI_0000.instMag + Feeder_1_DNP.AI_0001.instMag

But this example will not:

× Feeder_1_DNP.AI_0000 + Feeder_1_DNP.AI_0001

To perform math or logic on an entire tag structure, you can use the special built-in Tag Processor functions described in this section.

ADD_INS

This function adds the stVal of two INS tags.

Inputs:

IN1, IN2 : INS

Output:

Returns modified INS tag as follows:

- stVal is the result of IN1.stVal + IN2.stVal
- time is the most recent time between the two input INS tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
HMI_MODBUS.IREG_00000	INS	ADD_INS (IED1_MODBUS.IREG_00000, IED1_MODBUS.IREG_00001)	INS

Figure 3.17 ADD_INS Example

ADD_MV

This function adds the instMag of two MV tags.

Inputs:

IN1, IN2 : MV

Output:

Returns modified MV tag as follows:

- instMag is the result of IN1.instMag + IN2.instMag
- time is the most recent time between the two input MV tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.AI_0000	MV	ADD_MV (Feeder1_DNP.AI_0000, Feeder1_DNP.AI_0001)	MV

Figure 3.18 ADD_MV Example

AND_SPS

This function performs a logical AND on the stVal of two SPS tags.

Inputs:

IN1, IN2 : SPS

Output:

Returns modified SPS tag as follows:

- stVal is the result of IN1.stVal AND IN2.stVal
- time is the most recent time between the two input SPS tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.BI_0000	SPS	AND_SPS (Feeder1_DNP.BI_0000, Feeder1_DNP.BI_0001)	SPS

Figure 3.19 AND_SPS Example

DIV_INS

This function divides the stVal of the first INS tag by that of the second INS tag.

Inputs:

IN1, IN2 : INS

Output:

Returns modified INS tag as follows:

- stVal is the result of IN1.stVal / IN2.stVal
- time is the most recent time between the two input INS tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
HMI_MODBUS.IREG_00000	INS	DIV_INS (IED1_MODBUS.IREG_00000, IED1_MODBUS.IREG_00001)	INS

Figure 3.20 DIV_INS Example

DIV_MV

This function divides the instMag of the first MV tag by that of the second MV tag.

Inputs:

IN1, IN2 : MV

Output:

Returns modified MV tag as follows:

- instMag is the result of IN1.instMag / IN2.instMag
- time is the most recent time between the two input MV tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.AI_0000	MV	DIV_MV (Feeder1_DNP.AI_0000, Feeder1_DNP.AI_0001)	MV

Figure 3.21 DIV_MV Example

MULT_INS

This function multiplies the stVal of the first INS tag by that of the second INS tag.

Inputs:

IN1, IN2 : INS

Output:

Returns modified INS tag as follows:

- stVal is the result of IN1.stVal * IN2.stVal
- time is the most recent time between the two input INS tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
HMI_MODBUS.IREG_00000	INS	MULT_INS (IED1_MODBUS.IREG_00000, IED1_MODBUS.IREG_00001)	INS

Figure 3.22 MULT_INS Example

MULT_MV

This function multiplies the instMag of the first MV tag by that of the second MV tag.

Inputs:

IN1, IN2 : MV

Output:

Returns modified MV tag as follows:

- instMag is the result of IN1.instmag * IN2.instMag
- time is the most recent time between the two input MV tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.AI_0000	MV	MULT_MV (Feeder1_DNP.AI_0000, Feeder1_DNP.AI_0001)	MV

Figure 3.23 MULT_MV Example

NOT_SPS

This function performs a logical NOT on the stVal of SPS tag.

Inputs:

IN : SPS

Output:

Returns modified SPS tag as follows:

- stVal is the result of NOT (IN1.stVal)
- time is the time of IN
- q is the quality of IN

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.BI_0000	SPS	NOT_SPS (Feeder1_DNP.BI_0000)	SPS

Figure 3.24 NOT_SPS Example

OR_operSPC

This function effectively performs a logical OR of ctlVal of the first operSPC and that of the second operSPC.

Inputs:

IN1, IN2 : operSPC

Output:

Returns modified operSPC tag as follows:

IF (IN1.ctlVal = True) AND (IN1.q.operatorBlocked = FALSE)
OR_operSPC := IN1

ELSEIF IN2.ctlVal = True) AND (IN2.q.operatorBlocked = FALSE)
OR_operSPC := IN2

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
Feeder1_DNP.BO_0000.operClose	OPERSPC	OR_operSPC (SCADA1_DNP.BO_0000.operClose, SCADA2_DNP.BO_0000.operClose)	OPERSPC

Figure 3.25 OR_operSPC Example

OR_SPS

This function performs a logical OR on the stVal of two SPS tags.

Inputs:

IN1, IN2 : SPS

Output:

Returns modified SPS tag as follows:

- stVal is the result of IN1.stVal OR IN2.stVal
- time is the most recent time between the two input SPS tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.BI_0000	SPS	OR_SPS (Feeder1_DNP.BI_0000, Feeder1_DNP. BI_0001)	SPS

Figure 3.26 OR_SPS Example

SCALE_CMV

This function scales the instCval.mag, using the input parameters.

Inputs:

IN : CMV

SCALE : REAL

OFFSET : REAL

CEILING : REAL

FLOOR : REAL

Output:

Returns modified CMV tag as follows:

- instCVal.mag is the result of

```
(* Calculate the scaling *)
SCALE_CMV.instCVal.mag := IN.instCVal.mag * SCALE +
OFFSET;
(* Check for ceiling and floor *)
IF SCALE_CMV.instCVal.mag > CEILING THEN
SCALE_CMV.instCVal.mag := CEILING;
END_IF
IF SCALE_CMV.instCVal.mag < FLOOR THEN
SCALE_CMV.instCVal.mag := FLOOR;
END_IF
```

- time is the time of IN
- q is the quality of IN

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA_DNP.AI_0020	MV	SCALE_CMV (SEL_351S_1_SEL.VA , 100.0, 10000, 1E7, 0)	CMV

Figure 3.27 SCALE_CMV Example

SCALE_INS

This function scales the stVal, using the input parameters.

Inputs:

IN : INS

SCALE : DINT

OFFSET : DINT

CEILING : DINT

FLOOR : DINT

Output:

Returns modified INS tag as follows:

- stVal is the result of

```
(* Calculate the scaling *)
SCALE_INS.stVal := IN.stVal * SCALE + OFFSET;
(* Check for ceiling and floor *)
IF SCALE_INS.stVal > CEILING THEN
SCALE_INS.stVal := CEILING;
END_IF
IF SCALE_INS.stVal < FLOOR THEN
SCALE_INS.stVal := FLOOR;
END_IF
```

- time is the time of IN
- q is the quality of IN

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA_DNP.AI_0022	MV	SCALE_INS (SEL_3530_1_SEL.Rx_UW_1_F800, 10, 0, 10000000, -10000000)	INS

Figure 3.28 SCALE_INS Example

SCALE_MV

This function scales the instMag, using the input parameters.

Inputs:

IN : MV

SCALE : REAL

OFFSET : REAL

CEILING : REAL

FLOOR : REAL

Output:

Returns modified MV tag as follows:

- instMag is the result of

```
(* Calculate the scaling *)
SCALE_MV.instMag := IN.instMag * SCALE + OFFSET;
(* Check for ceiling and floor *)
```

IF SCALE_MV.instMag > CEILING THEN
SCALE_MV.instMag := CEILING;
END_IF
IF SCALE_MV.instMag < FLOOR THEN
SCALE_MV.instMag := FLOOR;
END_IF

- time is the time of IN
- q is the quality of IN

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA_DNP.AI_0021	MV	SCALE_MV (SEL_3515_1_SEL.P_WATTS , 50, 200, 1000000, 0)	MV

Figure 3.29 SCALE_MV Example

SUB_INS

This function subtracts the stVal of two INS tags.

Inputs:

IN1, IN2 : INS

Output:

Returns modified INS tag as follows:

- stVal is the result of IN1.stVal - IN2.stVal
- time is the most recent time between the two input INS tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
HMI_MODBUS.IREG_00000	INS	SUB_INS (IED1_MODBUS.IREG_00000, IED1_MODBUS.IREG_00001)	INS

Figure 3.30 SUB_INS Example

SUB_MV

This function subtracts the instMag of two MV tags.

Inputs:

IN1, IN2 : MV

Output:

Returns modified MV tag as follows:

- instMag is the result of IN1.instMag - IN2.instMag
- time is the most recent time between the two input MV tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.AI_0000	MV	SUB_MV (Feeder1_DNP.AI_0000, Feeder1_DNP.AI_0001)	MV

Figure 3.31 SUB_MV Example

XOR_SPS

This function performs a logical XOR on the stVal of two SPS tags.

Inputs:

IN1, IN2 : SPS

Output:

Returns modified SPS tag as follows:

- stVal is the result of IN1.stVal XOR IN2.stVal
- time is the most recent time between the two input SPS tags
- q is the result of merging the quality attributes of IN1 and IN2 to reflect the worse case

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.BI_0000	SPS	XOR_SPS (Feeder1_DNP.BI_0000, Feeder1_DNP.BI_0001)	SPS

Figure 3.32 XOR_SPS Example

SECTION 4

Engineering Access

Overview

A transparent connection provides a data tunnel between two or more Ethernet or serial communications ports. Each port configured in a transparent connection is an access point. The SEL Real-Time Automation Controller (RTAC) redirects communications traffic from one access point, or port, through a second access point and vice versa. We designate one access point as the source. This access point, typically a PC used for engineering access, third-party software, etc., initiates communication. Subsequent access points are the destinations of the connection. These are typically IEDs connected to the RTAC. The RTAC links access points together through an access point router (APR). In this way, the APR enables the RTAC as a port switch. Configure access points and a maximum of 100 APRs in ACSELERATOR RTAC® SEL-5033 Software to accommodate one or many transparent connections.

There are two main categories of transparent connections:

1. Standard transparent: enables ASCII tunneling through a port while leaving undisturbed the underlying binary communication occurring on that port. An example would be SEL protocol interleaving ASCII engineering access with an SEL relay while the RTAC is polling the relay via Fast Meter commands.
2. Direct transparent: enables binary or ASCII tunneling through a port to an IED but turns off the underlying communication occurring on that port. An example would be third-party software configuring an IED with Modbus. The RTAC will discontinue polling the Modbus IED on that port during a direct transparent session and will direct tunneled messages through to the IED instead.

Within an APR, you can also perform actions on the strings as they pass from source to destination and back. Actions you can perform include issuing commands, stripping characters from a string, inserting tag values into a string, and searching a string for a given value.

Configuration

The following examples illustrate how to configure the RTAC for standard transparent and direct transparent connections.

Example 4.1 Standard Transparent Connection

Configure the RTAC as a port switch to access four SEL relays from a remote engineering PC. The RTAC will prompt you for a password. After you enter the password, you can use the **PORT** command to switch to the ports attached to the relays. This example contains five access points:

- One Ethernet connection into the RTAC (source)
- Four SEL relays connected to the RTAC via direct serial connections (destinations)

1. Open a blank project in ACSELERATOR RTAC and insert the relays and access point.
2. On the ribbon menu, click **Insert** and then **SEL**.
3. Select an SEL-351S relay that uses the SEL protocol as a Client-Serial.
4. Right-click on the inserted relay, select **Copy**, and then paste the copy on Devices. Paste three copies for a total of four SEL-351S IEDs for this example. Rename the copies as shown in *Figure 4.1*.

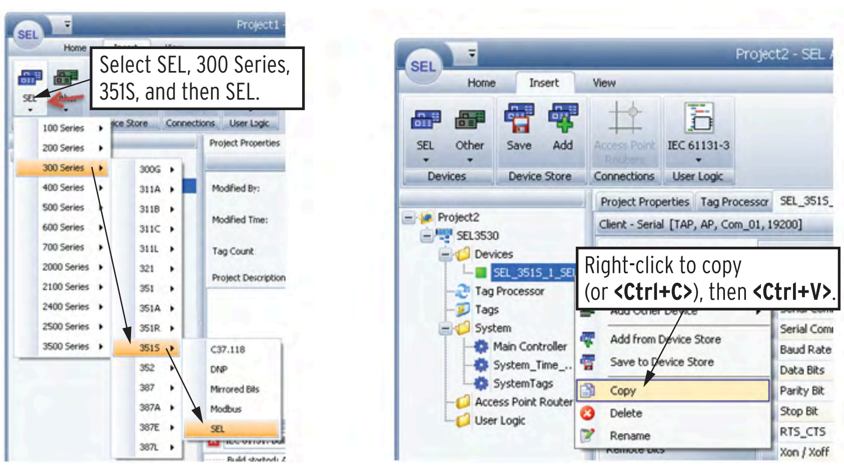


Figure 4.1 Configure Four SEL Access Points

5. On the ribbon menu, click **Insert** and then **Other**.

6. Insert an access point and configure this access point as **Ethernet Incoming**. This tells the RTAC that the access point will be listening for a connection. Conversely, you can configure other access points as outgoing to create connections to generic IEDs, etc. Client protocols, such as the SEL relays we inserted, already have an outgoing access point built in.

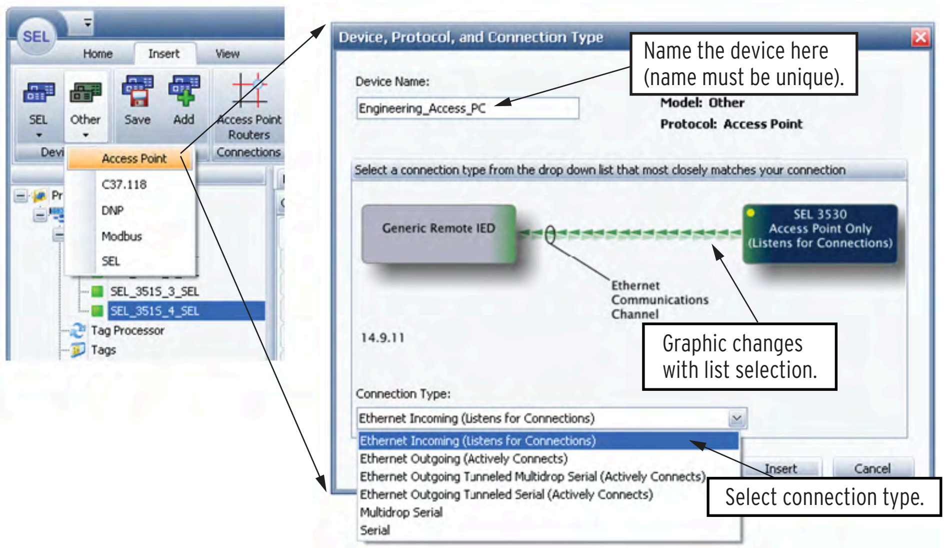


Figure 4.2 Insert Source Access Point

7. Configure the access point **Local Port Number** in the **Settings** tab of the new access point. You will address this IP port when you create the tunnel session with the RTAC. There is no IP address setting because all active network interfaces on the RTAC will listen for the connection.

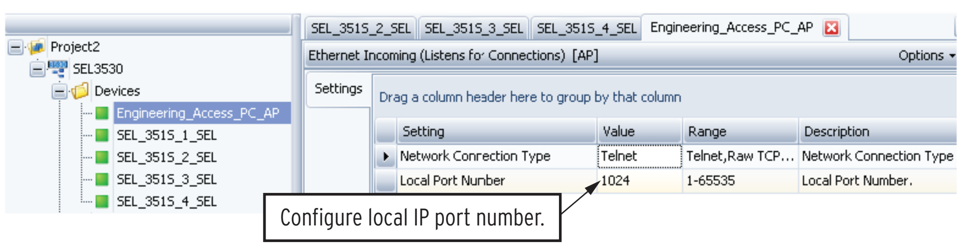


Figure 4.3 Configure Source Access Point

8. On the **Insert** ribbon, click **Access Point Routers**. In this example, we want the RTAC to emulate an SEL-2030, so select each of the SEL_351S_n_SEL_TAP entries. See *Figure 4.4*.

   Every device added to the RTAC is an access point. SEL IEDs have an extra access point designated _TAP for transparent connection sessions. Select the _TAP entry to permit the SEL binary protocol to continue to poll the SEL IED during an ASCII transparent connection session. Select the _AP entry to create a direct transparent connection that will stop all polling during a transparent connection with the device. *Example 4.2* describes the setup of this type of connection.

9. Select **Enable Legacy Commands** to force the RTAC to require a login and the **PORT** command in the tunnel session. Configure as illustrated in *Figure 4.4*.

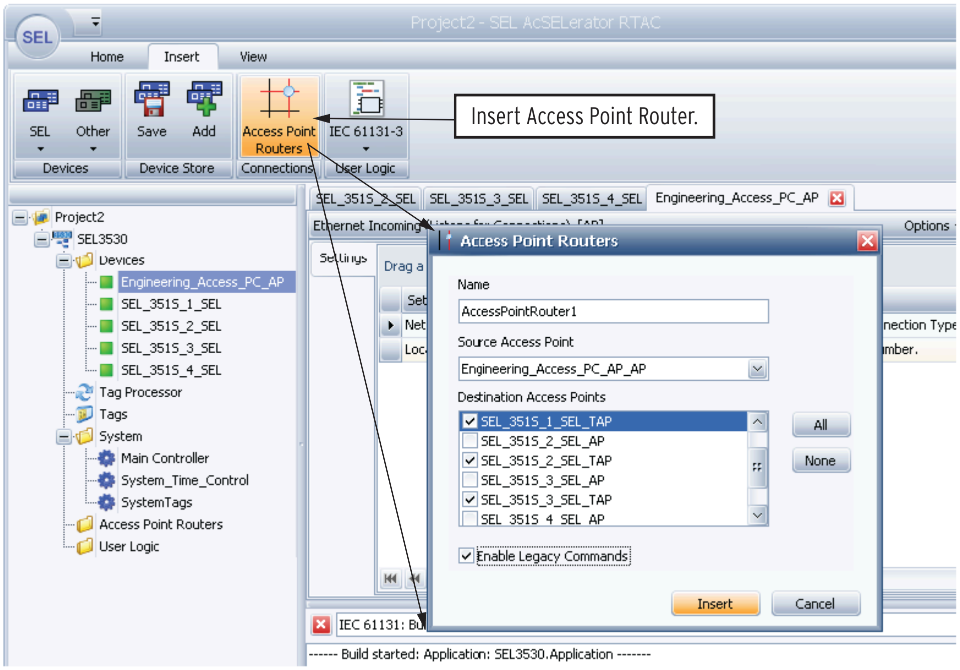


Figure 4.4 Insert Access Point Router

10. Save the project and notice that this example created one APR automatically for each device. Each of the four APRs has the same source access point.
11. Click on **AccessPointRouter1_1**.
12. Change the **Source_Authentication** setting to **TRUE** and save.

    This setting will force the RTAC to ask for a username and password to gain access to this APR. The username and password are configured via the RTAC web interface as described in *Section 7: Security and Account Management*. This setting is also global to all APRs. Configure for a particular group, so you only have to change it in one of the four APRs that are created in this example.
13. Load the project into the RTAC.

14. Test the project by opening HyperTerminal (or other terminal emulation program) as a Winsock connection. Enter the IP address of the connected RTAC and the Local Port Number you configured in Engineering_Access_PC_AP. Click **OK**.

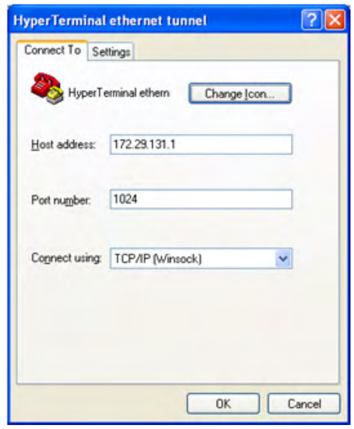


Figure 4.5 HyperTerminal Configuration

15. Press the connect icon  in HyperTerminal to connect and log in to the RTAC.

    Type **PORT *n***, where *n* is the Legacy_Port_Command_ID corresponding to the desired port number, to connect transparently to one of the SEL-351S relays. Press **<Ctrl+D>** to exit a port.

    The Legacy_Port_Command_ID setting field displays the ID that you will use for each APR. The default ID for the first APR is 1, with subsequent APR IDs incrementing thereafter. Had we also selected the _AP (direct transparent) APs as shown in *Figure 4.4*, we would have available additional APRs with unique port command IDs for each direct transparent connection. To open a transparent connection to PORT 1, type **Port 1** while connected to the RTAC through HyperTerminal to gain interleaved access to the device on PORT 1. To open a direct transparent connection on PORT1, type **Port 1 D** while connected to the RTAC through HyperTerminal to gain direct access to the relay on PORT 1. Configure the unique Legacy_Port_Command_ID, along with the default **PORT** command in Source_Rx_Message under POU Pin Settings within each APR.

Example 4.2 Direct Transparent Connection

This example shows how you can transparently connect to an IED without logging on to the RTAC or entering a **PORT** command. As soon as you connect to the RTAC, the APR directs communications to the access point you have configured in your project. In this example, that access point is a Modbus device. But you can follow the same steps to create a standard interleaved transparent connection to an SEL relay by configuring an SEL client instead of a Modbus client in this example.

Configure the RTAC to poll a Modbus IED. Then, use third-party software running on a PC to communicate with that IED through the RTAC. This example assumes that you have a Modbus IED connected to the RTAC on one serial port and an application running on your PC that can use Modbus protocol to configure the IED. The PC is connected to the RTAC on another serial port.

1. On the **Insert** ribbon, click **Other** and then **Modbus**.
2. Configure the Modbus device as **Client-Serial**.
3. Configure to poll some points from a connected Modbus IED. See *Section 2: Communications* for details on configuring a Modbus client.
4. Ensure that communications parameters (data rate, etc.) are correct in the Modbus client connection for the Modbus IED.
5. From the **Insert** ribbon, click **Other** and then **Access Point**. Configure as **Serial**.
6. Ensure that communications parameters are correct in the Serial access point connection for the Modbus PC application.
7. Insert an APR. This time, do not select Enable Legacy Commands. See *Figure 4.6*.

   We want the APR to connect the source access point (configuration software on the PC) automatically with the destination access point (Modbus IED) when the source initiates a connection.

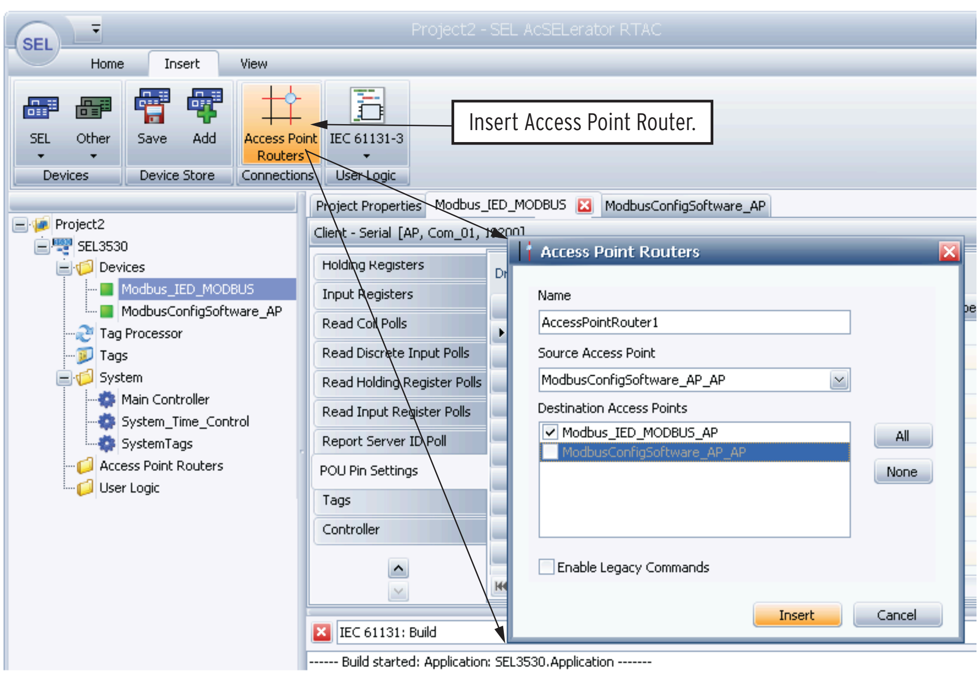


Figure 4.6 Direct Transparent Modbus

8. Select the new access point router from the left menu. In this example, it is AccessPointRouter_1.
9. Click on the **POU Pin Settings** tab.

10. Set **Auto_Connect = TRUE**. This enables automatic connection and eliminates the need to log in or use the **PORT** command to connect manually to a port.

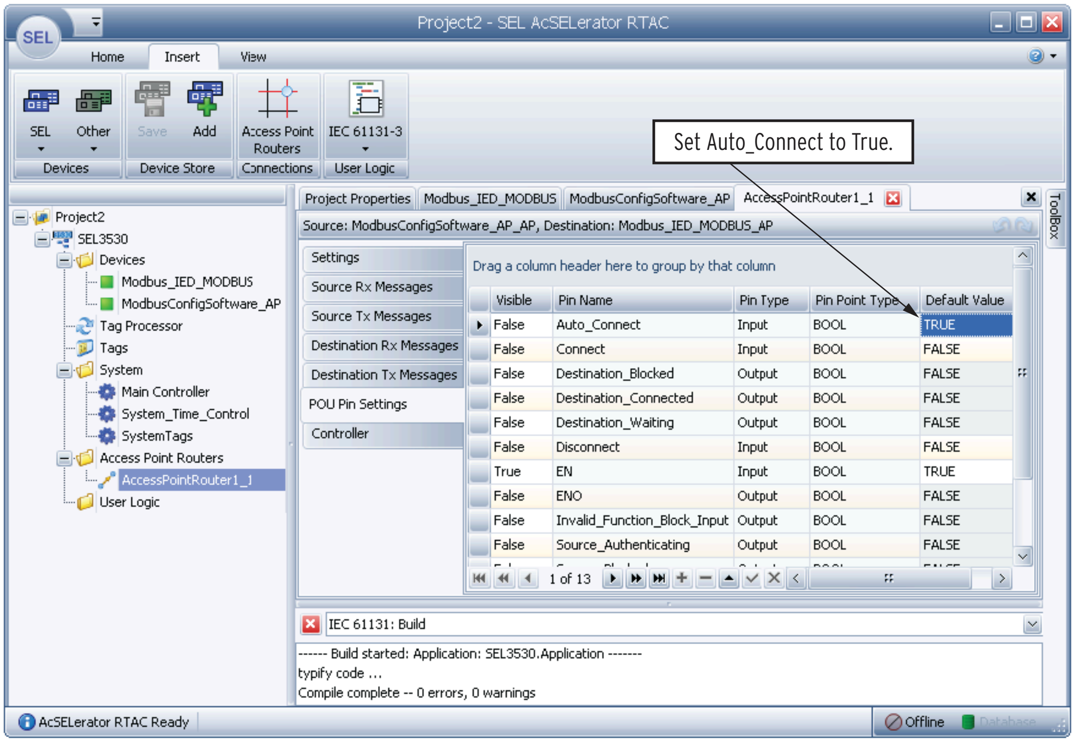


Figure 4.7 Enable Autoconnection

11. Save the project and load it into the RTAC.

    To test this project, you will need a Modbus IED connected to the serial port Com_01. You will also need a PC that is running a Modbus software application designed to connect to the IED. Connect the PC to Com_02. When the Modbus software attempts to communicate, the RTAC will automatically stop polling the Modbus IED and direct all communications traffic from Com_02 to Com_01. The RTAC will redirect data it receives on Com_01 to Com_02. When Com_2 stops receiving data from the PC, the RTAC will allow its Modbus polling task to resume polling the IED on Com_01.

Along with transparent or direct transparent communications, you can also configure the APR to perform special actions. Each action is associated with either all messages or, more typically, with the messages sent to or received from the source or the destination. To create an action on a message, perform the following steps:

1. Determine what message you want the action to apply to. Options are **Source Rx Messages**, **Destination Rx Messages**, **Source Tx Messages**, and **Destination Tx Messages**.
2. Click on the corresponding tab in the APR configuration and then click on the + box at the bottom of the screen to add the number of messages needed. For example, if you needed the action to send the MET command to the destination, click on Destination Tx Messages and add one

message. When you add a message, the software will create three new POU pins in the POU pin settings with a name associated with the new message you created. For example, if you created one new Destination Tx Message, the software creates the following new pins:

Pin Name	Pin Type	Usage
Destination_Tx_Message_1	Input	Use this pin to configure the string value of the message.
Trigger_Destination_Tx_Message_1	Input	Toggle this pin from **False** to **True** to force the transmit of this message to the destination.
Trigger_Destination_Tx_Message_1_DN	Output	This pin toggles **TRUE** for one process cycle after the trigger input has been asserted.

3. Configure the string value in the new message.
4. Optionally, use the trigger and DN pins to perform the actions needed.

By entering information in the transmit and/or receive strings in a special syntax, you can cause the APR to perform four main categories of special functions:

- Commands
- ASCII control and key characters
- Tag name replacement
- ASCII string hex representation

Commands

Commands are special syntax that perform actions when they are interpreted by the RTAC firmware. Commands are always enclosed in curly brackets { } and are ASCII characters that are not case sensitive. Some commands apply to both source and destination, whereas other commands are specific to only source or destination. *Table 4.1* shows a list of available commands with *XX* indicating you must specify **SOURCE** or **DESTINATION** as part of the command.

Table 4.1 APR Command Summary

Command	Function
{CONNECT}	Connects the source and destination access points. Same functionality as asserting the Connect input pin.
{DISCONNECT}	Disconnects the source and destination access points. Same functionality as asserting the Disconnect input pin.
{ENABLE_HW_FLOW_*XX*}	Enables hardware flow control on access points associated with serial port.
{DISABLE_HW_FLOW_*XX*}	Disables hardware flow control on access points associated with serial port.
{ENABLE_SW_FLOW_*XX*}	Enables software flow control on access points associated with serial port.
{DISABLE_SW_FLOW_*XX*}	Disables software flow control on access points associated with serial port.

Command	**Function**
{ASSERT_RTS_*XX*}	Forces RTS high on serial ports associated with this access point. Only applies if HW flow control is turned off by **{DISABLE_HW_FLOW_*XX*}** or the setting RTS_CTS is FALSE.
{DEASSERT_RTS_*XX*}	Removes the force RTS high condition on serials ports associated with this access point. Only applies if HW flow control is turned off by **{DISABLE_HW_FLOW_*XX*}** or the setting RTS_CTS is FALSE.
{WAIT_FOR_CTS_*XX*}	Inhibits transmit until CTS asserts. This command is reset when the transparent connection is terminated; so, it must be issued after a connection is made to be active.
{ASSERT_SERIAL_PORT_POWER}	Asserts serial-port power.
{DEASSERT_SERIAL_PORT_POWER}	Deasserts forced serial-port power.
{DELAY_*Y*_*XX*}	Adds a delay of *Y*, where *Y* is an integer between 0–60000 ms.
{ECHO_ON_*XX*}	Enables character echo for Source Tx or Destination Tx or both.
{ECHO_OFF_*XX*}	Disables character echo for Source Tx or Destination Tx or both.
{TIME_OUT_*Y*_*XX*}	Terminates the transparent connection if no characters are received in *Y* seconds, where *Y* is an integer between 0–60000 seconds. A value of 0 disables time out.
{REPLACE_*XX arg1* | *arg2*}	Replaces characters in a data stream entering the RTAC in the direction specified by *XX* or both directions when *XX* is left blank. For example, **REPLACE_SOURCE** replaces characters coming in the source access point.
{STOP_REPLACE_*XX*}	Used to stop replacing characters/bytes in the direction specified by *XX* or both directions when *XX* is left blank.
{STRIP_*XX argument*}	Used to strip characters, where argument is a sequence of characters that will be stripped from the data stream entering the RTAC in the direction specified by *XX* or both directions when *XX* is left blank (i.e., **STRIP_SOURCE** strips characters coming in the Source Access point). If numerous strip commands are issued, the strip function is processed in the order that they were received.
{STOP_STRIP_*XX*}	Stops all stripping of data. However, the **{FAST_OP_ENABLE}** and **{FAST_OP_DISABLE}** are still in effect.

Command	Function
{FAST_OP_ENABLE}	Enables transmitting of SEL Fast Operate commands from the source to destination (destination to source are always permitted). The status of this command is not maintained through device restart.
{FAST_OP_DISABLE}	Disables SEL Fast Operate commands from source to destination. The status of this command is not maintained through device restart.

String Stripping

Using the **STRIP** command, you can search for and strip out any characters from the incoming string. The syntax is {STRIP_*XX*_*argument*}, where *XX* is either **SOURCE** or **DESTINATION**, and *argument* is a sequence of characters to be stripped from the data stream. If *XX* is left out, the APR will strip the character sequence from both source and destination streams. Multiple consecutive strip commands are executed in the order they are configured. For example, {STRIP ab} followed by {STRIP bc} applied to the string 'abcdefg' will result in 'cdefg' because the 'ab' was stripped off first, so there was not a match for 'bc'.

Examples:

- Source_Tx_Message_1 = {STRIP_SOURCE<SP>} will strip all space characters from the Tx data.
- Source_Tx_Message_8 = {STRIP_SOURCE<0104>} would disable a MODBUS Force Coil for Outstation Address 1.

String Replacement

Use the **REPLACE** command to match character strings and replace with other character strings. Arg1 is the sequence of characters to be replaced by the sequence of characters in arg2. If the length of arg2 is greater than the length of arg1, arg1 is fully populated by arg2 and extra characters from arg2 that do not fit in arg1 are dropped. If the length of arg1 is greater than arg2, arg1 is populated by arg2 characters and all trailing characters in arg1 not replaced by arg2 characters are stripped. White spaces after arg1 and before arg2 are removed. To use literal white spaces at the start or end of these arguments, use the binary notation, i.e., <20>.

Examples:

- Source_Tx_Message_6 = {REPLACE <HT> | <SP>} would replace all horizontal tabs with a space.
- Source_Tx_Message_7 = {REPLACE abc | defg} would replace 'abc' with 'def'.
- Source_Tx_Message_8 = {REPLACE abc | d} would change the string 'abc' to 'd'.

ASCII Control and Key Characters

You can place ASCII control and key characters in a message string in the SPR. These single-byte strings must be preceded with the dollar sign special character $. For example, this string has a carriage control and line feed RL. The $ also escapes a following special character so that it is treated like a string rather than a special character. You must use two dollar signs, ($$), for the following characters: { } [ ] < >. For example, this is a less than sign $$<.

Table 4.2 shows examples of using the $ character. Notice control and key characters are not case sensitive.

Table 4.2 ASCII Controls and Keys

String	Result
$$	One dollar sign
$'	A single quote
$"	A double quote
$L or $1	Line feed
$N or $n	Newline
$P or $p	Form feed
$R or $r	Carriage return
$T or $t	Tab
$${	{
$$<	<
$$]	]

You can also wrap the control and key characters with the greater than and less than signs and use the delimiters space, +, -, or _. For example, all of the following are equivalent representations of carriage return, line feed:

- $0D$0A
- rL
- <CR><LF>
- <CTRL-M><CTRL-J>
- <CTRL M><CTRL J>
- <CTRL_M><CTRL_J>
- <CTRL+M><CTRL+J>

A special case ASCII control can be used to append a security check, such as a checksum or cyclic redundancy check (CRC), onto the end of a message. The format is <[*XXXX*]>, where you replace *XXXX* with the respective CRC special string. For example, you could send the following string as a Modbus message:

<01047E910009><[CRC_16_MODBUS]>

The security check strings available are shown in *Table 4.3*.

Table 4.3 Security Check Strings

Control	Attribute	Attribute Value
CRC_8	Width	8
	Polynomial	07h
	Initial	00h
	Reflect data bit order	True
	Reflect CRC bit order	True
	Final XOR	00h
CRC_16	Width	16
	Polynomial	8005h
	Initial	0000h
	Reverse data byte order	False
	Reverse CRC byte order	False
	Reflect data bit order	True
	Reflect CRC bit order	True
	Final XOR	0000h
CRC_16_CCITT	Width	16
	Polynomial	1021h
	Initial	FFFFh
	Reverse data byte order	False
	Reverse CRC byte order	False
	Reflect data bit order	False
	Reflect CRC bit order	False
	Final XOR	0000h
CRC_16_X25	Width	16
	Polynomial	1021h
	Initial	FFFFh
	Reverse data byte order	False
	Reverse CRC byte order	False
	Reflect data bit order	False
	Reflect CRC bit order	False
	Final XOR	0000h
CRC_16_MODBUS	Width	16
	Polynomial	8005h
	Initial	FFFFh
	Reverse data byte order	False
	Reverse CRC byte order	True

Control	Attribute	Attribute Value
	Reflect data bit order	True
	Reflect CRC bit order	True
	Final XOR	0000h
CRC_16_DNP	Width	16
	Polynomial	3D65h
	Initial	0000h
	Reverse data byte order	False
	Reverse CRC byte order	False
	Reflect data bit order	True
	Reflect CRC bit order	True
	Final XOR	FFFFh
CRC_16_SEL	Width	16
	Polynomial	8005h
	Initial	FFFFh
	Reverse data byte order	False
	Reverse CRC byte order	True
	Reflect data bit order	True
	Reflect CRC bit order	True
	Final XOR	0000h
CRC_32	Width	32
	Polynomial	04C11DB7h
	Initial	FFFFFFFFh
	Reverse data byte order	False
	Reverse CRC byte order	False
	Reflect data bit order	True
	Reflect CRC bit order	True
	Final XOR	FFFFFFFFh
CRC_32_I363	Width	32
	Polynomial	04C11DB7h
	Initial	FFFFFFFFh
	Reverse data byte order	False
	Reverse CRC byte order	False
	Reflect data bit order	False
	Reflect CRC bit order	False
	Final XOR	FFFFFFFFh
Checksum_8_Standard		

Control	Attribute	Attribute Value
Checksum_8_XOR		
Checksum_SEL_FastOp		
Checksum_16_Standard		

Tag Name Replacement

If you enter a Virtual Tag tag name in a string in the APR and enclose the tag name in [ ], the actual value of the tag will be used instead of the tag name. For example, if a global tag VirtualTagList1.User_logged_off = Joe, then the string, User [VirtualTagList1.User_logged_off.strVal] has logged off, will result in `User Joe has logged off`. You must use a Virtual Tag as the tag name.

String Capture

You can also capture a value from a string into a Virtual Tag of type STR. The APR will search for the string you have configured and place the data into the Virtual Tag name enclosed in [ ]. For example, configure a Virtual Tag of type STR called VirtualTagList1.new_password. If the received message is `PAS 1 OTTER<0A>` and the configured string in the APR is SET PAS 1 [VirtualTagList1.new_password]<0A>, then VirtualTagList1.new_password.strVal will be set to OTTER.

The APR will gather as many as 255 characters or until the terminating sequence is reached. In the example given, the terminating sequence is a carriage return.

ASCII String Hex Representation

As in *ASCII Control and Key Characters on page 435*, all characters in the APR strings are considered ASCII characters unless framed in the following special characters: < >, [ ], { } or preceded by $ or $$. You can, therefore, send out a string of hex values that will not be interpreted as ASCII characters. For example, if you want to send the decimal value 17, you would enter the string <11>. In hexadecimal 11 is decimal 17. This allows you to create and send binary messages, such as Modbus, DNP, or other binary strings through the APR. For example, the string <010407D00005><[CRC_16_MODBUS]> is a Modbus read input register poll (Opcode 4) to IED Address 1, polling 5 registers starting at 2000, or 07D0 in hexadecimal.

Example 4.3 Source Receive String Value Capture

You can monitor incoming messages in an Access Point Router for a certain sequence of characters and then capture the data following the matched sequence into a Virtual Tag of STR type. This example demonstrates monitoring a remote access connection for a set password command to the connected relay, capturing that password, and applying the new password to the SEL client. All of the original receive message is still passed through the APR and only a copy of the password is captured in this example. This example uses an SEL-451-5 relay with version R305 firmware. More current versions of SEL-451 firmware perform password management somewhat differently and will not work with this example, but the concepts of capturing data are still applicable.

1. Insert an SEL-451 as SEL protocol, serial client.
2. Insert an access point as **Other**, **Ethernet Incoming (Listens for Connections)**. In this example, the access point is renamed Incoming_Access_AP. Set the **Network Connection Type** to **Telnet** for this example.
3. Insert an access point router and select the access point from Step 2 as the source and the SEL-451 TAP from Step 1 as the destination. See *Figure 4.8*.

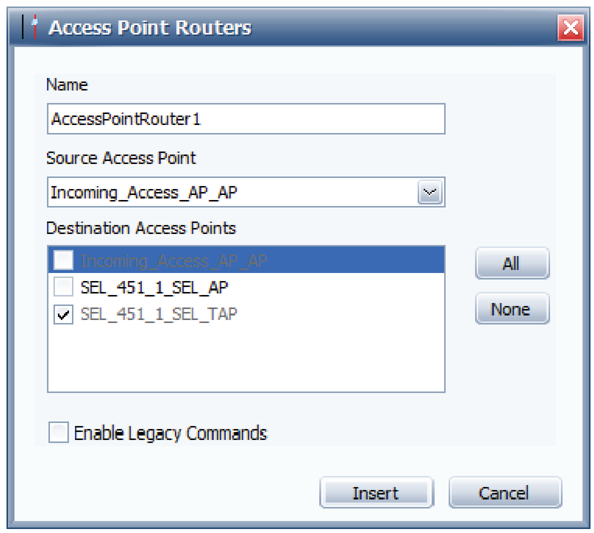


Figure 4.8 Insert Access Point Router

4. Create a virtual tag list by clicking on **Tag Lists** under the **Insert** ribbon.
5. Create a new virtual tag by clicking on the + on the bottom of the screen and change the tag type to **STR** and save the project.
6. In the access point router, click **Source RX Messages**, and then click on the + to add one message.
7. In the access point router, click on **POU PinSettings**. Set the Default Value for Auto_Connect = TRUE. Notice there is a new input pin called **Source_Rx_Message_1**. This is where we will define what string the RTAC will look for from the source access point.
8. Configure the Default Value for the POU Pin Setting Source_Rx_Message_1 as follows:

   PAS 1 [VirtualTagList1.Status_0000]<CR>

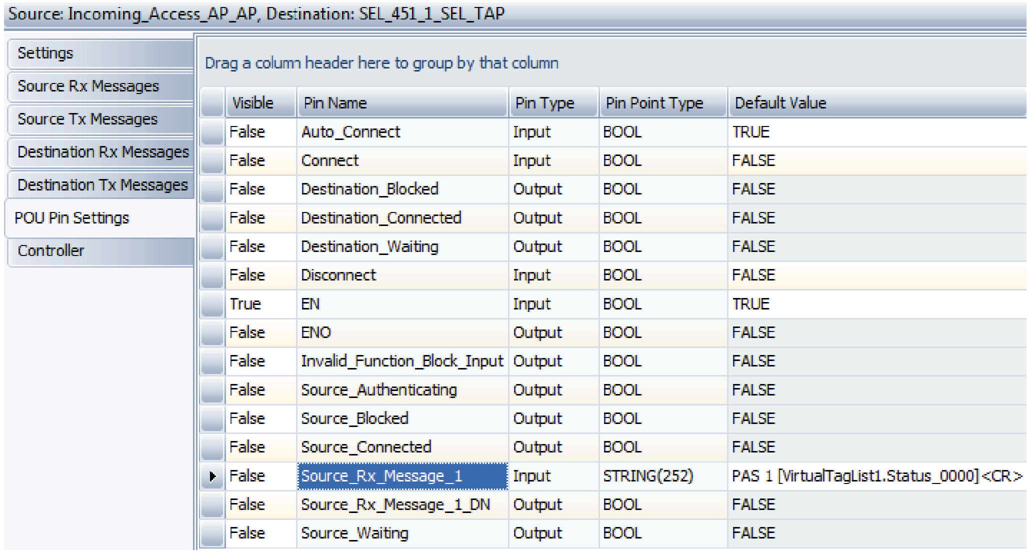


Figure 4.9 Configure Capture Message

The RTAC monitors the source receive messages for a string that matches 'PAS 1'. When it finds the string, it will capture a copy of all characters following the matched string until it reaches the termination (which in this example is <CR>) or 252 bytes, whichever comes first. The captured characters are placed into a global virtual tag of type STR called **VirtualTagList1.Status_0000**. The entire message, including the characters that were copied and captured, is still passed through the access point router.

9. On a new line in the Tag Processor, type **VirtualTagList1.Status_0000.str- Val** in the Source Expression column and **SEL_451_1_SEL_POU.Level_1_Password** in the Destination Tag Name column to assign the new password value to the SEL-451 client level 1 password POU pin setting.
10. Save the project and load it into the RTAC.

To test the project, wait until the RTAC completes auto configuration with the SEL-451. Then transparently connect to the relay through the RTAC. Log in to level 1, then level 2 on the relay (ACC and 2AC, respectively) by using the correct passwords. Use the **PAS 1** command to change the level 1 password.

PAS 1 OTTER123

POU Pin Settings

While on-line with ACSELERATOR RTAC, check the POU pin SEL_451_1_SEL_POU.Level_1_Password. It should change to OTTER123 to match the new level 1 password you configured in the relay through the transparent connection.

POU Pin Settings

Use POU pin settings to customize how the access point router operates in your project and to view the present state of the APR operations. Setting the Visible field to **True** will cause the POU pin to appear in the **Controller** tab. See *Table 4.4* for the settings descriptions. You can specify destination and source message strings as a manually entered string contained within single quotes or as a tag of type STRING. If you use a tag of type STRING, you can manipulate that string value in user logic and then trigger when the string is sent or automatically change the string you are trying to match.

Table 4.4 APR POU Pin Settings

Pin Name	Pin Type	Description	Default Value
Auto_Connect	Input BOOL	Set TRUE to connect as soon as traffic is detected on the source AP.	FALSE
Connect	Input BOOL	Automatically toggles when a connection is made. You can force to TRUE to force a connection.	FALSE
Disconnect	Input BOOL	Automatically toggles when connection is closed. You can force to TRUE to close a connection.	FALSE
EN	Input BOOL	Enables this APR. You can force to FALSE to disable this APR.	TRUE
*xx*_Rx_Message_*n*	Input STR	The *n*th user-configurable message string for the message received from the *xx* (source or destination) AP. Typically this is used to match what is received from the source.	
*xx*_Tx_Message_*n*	Input STR	The *n*th user-configurable message string to transmit to the *xx* (source or destination) AP.	
Trigger_*xx*_Tx_Message_*n*	Input BOOL	Trigger the transmission to *xx* (source or destination)	FALSE
*xx*_Blocked	Output BOOL	*xx* (source or destination) is blocking a connection to APR.	FALSE
*xx*_Connected	Output BOOL	*xx* (source or destination) is connected to APR	FALSE
Trigger_*xx*_Tx_Message_*n*_DN	Output BOOL	Indicates the *n*th transmission of the *xx* (source or destination) message has completed.	FALSE
*xx*_Rx_Message_*n*_DN	Output BOOL	Indicates the *n*th receive message of the *xx* (source or destination) has completed. This is determined by comparing the incoming message with the user-configured *xx*_RX_*n*_Message.	
Source_Authenticating	Output BOOL	Source is in the process of authenticating connection.	FALSE
*xx*_waiting	Output BOOL	*xx* (source or destination) is waiting for connection.	FALSE
Invalid_Function_Block_Input	Output BOOL	The APR function block has received some invalid input.	FALSE
ENO	Output BOOL	The APR is enabled.	FALSE

SECTION 5

Web HMI and Reports

Overview

The devices in the SEL Real-Time Automation Controller (RTAC) and SEL Axion® family provide a variety of ways to access data from configured settings and connected IEDs using a web browser. Remote intelligent electronic device (IED) status and reports, local sequence of event reports, an optional full-featured web-based human-machine interface (HMI), and more, provide a capable set of tools for both live data viewing and troubleshooting.

NOTE
The SEL-3505 and SEL-3505-3 currently do not support the RTAC HMI option but do support Live Data.

The Connected IED Report displays remote IEDs by physical port. It provides information about connection type, settings, and communication status. It includes a ping utility that can be used to help determine if the RTAC is able to reach a remote Ethernet IED. Multiple simultaneous serial and network traffic captures can be recorded; these are useful for detailed communications and protocol analysis.

The Alarm Summary and Sequence of Events (SOE) reports provide comprehensive logging for all tags in the RTAC, including those that connected IEDs provide. The logging system compensates for time-stamp differential between data from different IEDs, so all data are in sequence and on the same time reference. The RTAC can log changes in state for Booleans, changes in value for strings, and for a change in time stamp for Booleans, analogs, or strings. The RTAC can also log alarms on analog values that cross defined thresholds. Configure tags for logging in the Tag Processor or use one of the logger function blocks in IEC 61131 custom programs. The logger stores as many as 30,000 records. The RTAC can log bursts of as many as 500 log entries in 100 ms or 100 entries per second continuously without losing information. In addition to the secure web interface, a log-viewing mechanism is provided in the form of an Open Database Connectivity (ODBC) connection. Use the ODBC connection for standard data transfer between the logged data and database or spreadsheet software.

The RTAC makes events collected from remote IEDs available in an Event Collection table. Event collection is configured in ACSELERATOR RTAC SEL-5033 Software. These events can be sorted by time or device and can be downloaded or deleted from the web interface.

Values of any tag or POU pin in the RTAC settings can be viewed on the **Live Data** page. This feature must be enabled by tag in ACSELERATOR RTAC. Not only do these values update in close to real time, they can also be forced from the web interface. This is useful for testing, troubleshooting, and commissioning.

The **Diagnostics** page provides a detailed description of the RTAC network interface state, including information about the firewall, physical interface, routing table, exe-GUARD®, PRP, and the ARP table.

Most devices in the RTAC and Axion family support the optional web-based HMI (the SEL-3505 and SEL-3505-3 currently do not). The HMI is a configurable option and can either be included in the order from the factory or can be added after the fact by contacting your SEL sales representative. The web-based HMI allows you to view and control any tags configured in the RTAC. Use ACSELERATOR Diagram Builder™ SEL-5035 Software to build custom HMI screens and load them into the RTAC. Build one-line diagrams, annunciators, and graphical representations that contain control buttons and display data from any tag. Because the HMI is web-based, multiple people can view it simultaneously.

Web HMI and Dashboard Settings

ACSELERATOR Diagram Builder provides not only a full-featured HMI development environment, but it also allows you to select HMI tags directly from the database in the RTAC or in an ACSELERATOR RTAC project. Use ACSELERATOR Diagram Builder to build new HMI screens, link HMI objects to RTAC database tags, and load the new HMI screens into the RTAC. Once you have loaded the screens into the RTAC, you can view and operate the screens from the RTAC's web interface (see *Figure 5.1*).

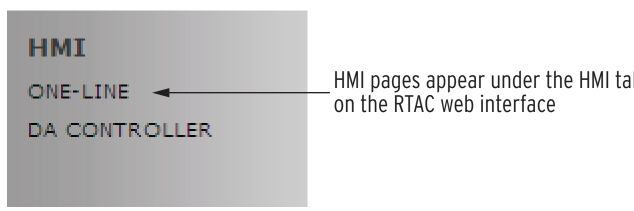


Figure 5.1 HMI Page Display

Settings on the RTAC dashboard allow you to configure when the web interface should log out of a user session because of inactivity. You can also configure the RTAC to allow HMI pages to remain active indefinitely, but in a read-only mode that prevents users from issuing controls unless they are logged on. The following table describes RTAC web Dashboard tab settings, including those pertaining to session time-out behavior.

Label	Description
Host Name	The network hostname as defined in the **Network** tab under Ethernet > Edit Global Settings.
Device Name	User-configurable descriptive name.
Device Location	User-configurable location description.
Device Description	User-configurable description of this device.

Label	Description
Allowed Web Connections	The number of concurrent web connections this device will accept. **Note**: This is not the same as Number of Users Logged In, which includes ODBC connections through ACSELERATOR RTAC and other programs. The number of users logged on includes sessions not logged off that are closed. The counter will not decrease until those sessions time out.
Web Session Time-out	The Web Session Time-out has a range of 0–720 minutes. A setting from 1 to 720 minutes (12 hours) will cause the web interface to log out of the active session after the defined period of inactivity. If the range is set to 0, the web interface will not log out after any period of inactivity as long as the network connection is good and the browser session remains open. When this setting is set to 0, the HMI will remain open and users can acknowledge alarms and send controls without authenticating. If Enable HMI Read-Only Mode is selected and an HMI diagram is presently in focus, the RTAC changes the HMI session to read-only mode. If HMI Read-Only Mode is selected, the defined behavior will occur regardless of the Web Session Time-Out value.
HMI Read-Only Mode Time-out	Once the HMI session has entered read-only mode, this is the maximum amount of time a communications failure can exist between the RTAC and HMI before the RTAC logs off the user and closes the web session.
Enable HMI Read-Only Mode	A flag that enables read-only mode in an HMI screen when the Web Session Time-out expires. Once the HMI is in read-only mode, a user can view data in the HMI, but the RTAC will prompt the user to log in if the user attempts to operate a control from the HMI or navigate to another non-HMI tab on the RTAC web interface.
Tie Alarm LED to OUT101	A flag that asserts the output OUT101 if the Alarm LED is asserted. Default value is enabled. OUT101 is NC when this box is selected and NO when this box is not selected.

Kiosk Feature

In firmware versions R142 and later, a Kiosk feature is available via the local display on the SEL RTAC. Enter or exit the web display of the interface by pressing **<F11>**. Kiosk mode allows specific applications to be run on the RTAC.

In Kiosk mode, the **Applications** menu (accessible via the bottom left corner of the display) provides the following functionality.

Virtual Keyboard

A virtual keyboard that can be docked to the bottom of the screen can be used with all local applications.

Settings

Calibrate Touchscreen

This option brings up the same options as the touchscreen calibration shortcut keys.

Display

This option allows you to adjust the following display settings:

- Connected Monitors
- Resolution
- Refresh Rate
- Rotation
- Reflection

Local RTAC UI

This option allows you to access any web servers that are listed in the URL Whitelist section through the local web client on the local display.

Table 5.1 Kiosk Feature Shortcut Keys

Shortcut Keys	Description
<F11>	Enter or exit full-screen mode
<Alt+T>	Calibrate touchscreen
<Ctrl+T>	Calibrate touchscreen
<Ctrl+Shift+Del>	Delete browser cache
<Alt+F4>	Close the selected/active application

Web Proxy

Web Proxies allow for configuration of a URL on the RTAC that will trigger the RTAC to create a proxy connection to another web server via one of its Ethernet interfaces.

Access to web proxies can be configured to be used by either unauthorized or authorized users. Authorized user confirmation is done by account type. A web proxy is a role-based permission that can be tied to a user account to allow access to the configured proxy.

Web proxies can be accessed via a remote connection or from the local display interface.

Figure 5.2 New Web Proxy Configuration

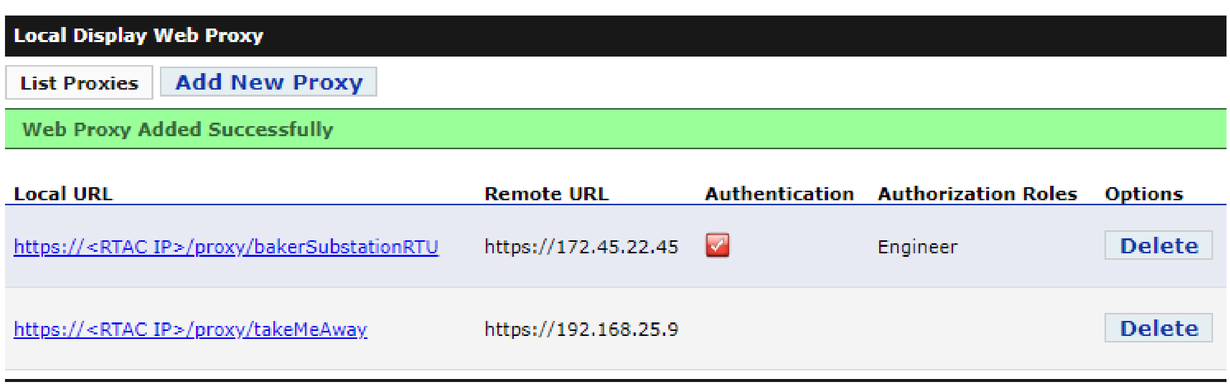


Figure 5.3 Web Proxy List

Viewing and Troubleshooting Connected IEDs

View information about port configuration, access the ping utility, and record network and serial device traffic using the Connected IED Report on the RTAC web interface. Graphics representing serial and Ethernet ports are shown, with grey indicating that the port is not configured and green indicating that the port has been configured. Click **All Serial**, **All Ethernet**, or **All IEDs** to see a summary list of all configured reports in respective categories. Or, click on an individual port to view only information for that port. Serial ports list the configured device or devices, communication settings, the number of bytes transmitted, the number of bytes received, RTS, and CTS status (if applicable), and provide controls to start a serial traffic capture. Because the byte count and RTS/CTS information are by port (not per IED) and obtained at a low level, the numbers will not necessarily reflect what is happening at a protocol level. The Ethernet IED report lists configured devices, IP addresses, TCP ports, a check box to include device traffic in a network capture, and a link to that device under the **Remote Device** column. You can access the web interface of Ethernet connected IEDs by clicking on the link. A pop-up window gives you an opportunity to specify the TCP port number as well as cancel the navigation. See *Figure 5.4*.

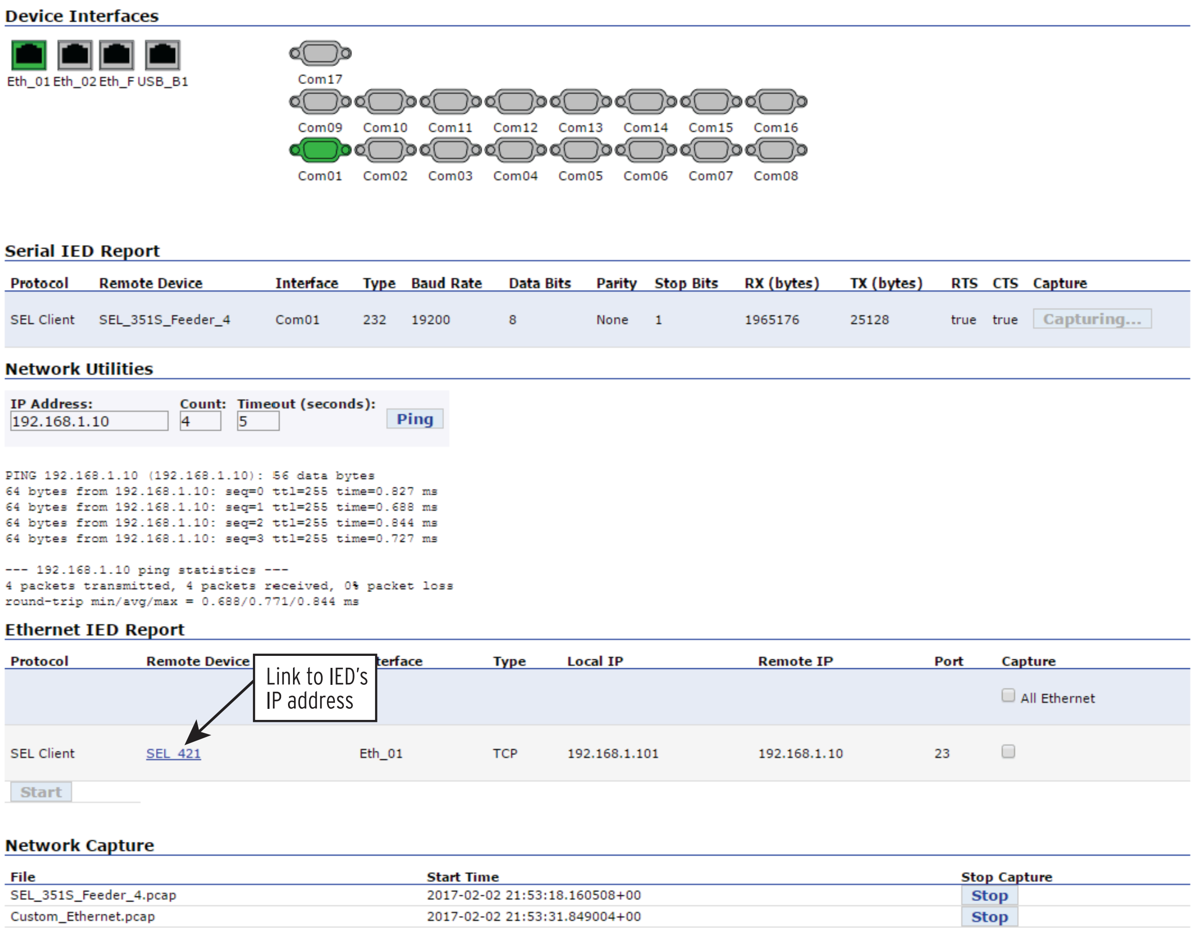


Figure 5.4 Connected IED Report

Web-Based Ping Utility

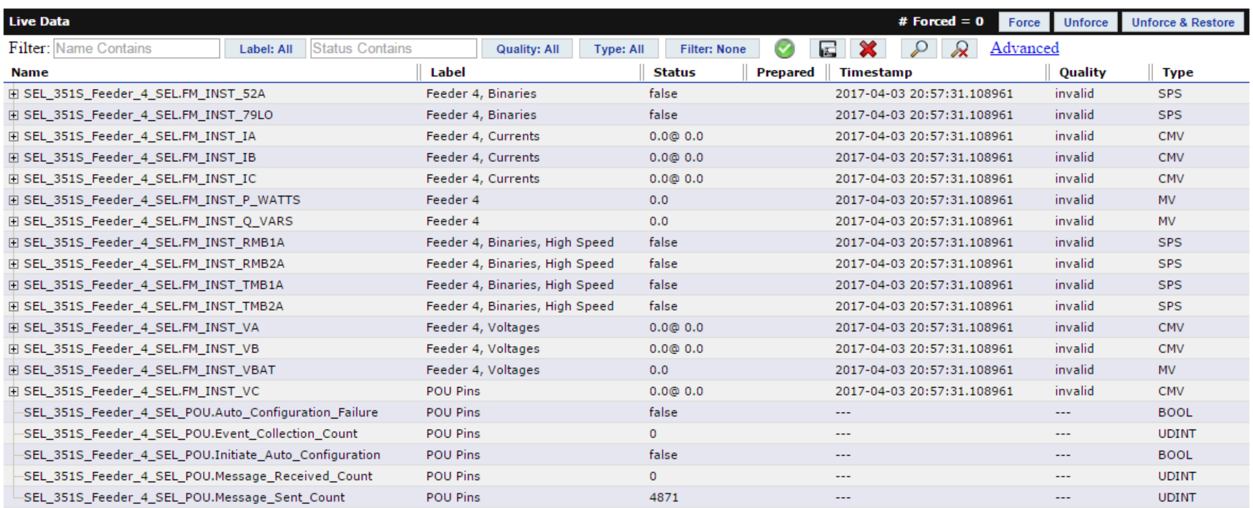
The web-based ping utility is found in **Network Utilities** on the **Connected IED Report** page. It is used to determine whether connected IEDs can be reached by the RTAC over an Ethernet network. Enter an IP address, the number of pings to send (Count), and the number of seconds to allow before a ping is considered failed because of lack of response (Timeout). Once these three items are set as desired, press the **Ping** button.

Web-Based Serial and Network Traffic Captures

The RTAC can record as many as 12 simultaneous captures of network or serial device traffic. A capture, once started, appears as a file download in the web browser. Once the capture is stopped, the file download completes. The completed capture appears as a file with the extension .pcap, which can be opened by network packet analyzer software. To start a serial capture, locate the **Capture** column in the **Serial IED Report** section of the page and press the **Start** button next to the desired IED. To start an Ethernet traffic capture, locate the **Capture** column in the **Ethernet IED Report** section of the page. You can either select one or more check boxes associated with the desired IEDs, or you can select the **All Ethernet** check box, then press the **Start** button below the IED table. To stop a capture, locate it in the list of running captures in the **Network Capture** section of the page and click the associated **Stop** button.

Live Data

The **Live Data** page displays the near real-time status of any Live Data enabled tag (this is done in the **Tag Processor** or in custom logic). Many data types are supported.



Live Data   # Forced = 0   Force   Unforce   Unforce & Restore
Filter: Name Contains   Label: All   Status Contains   Quality: All   Type: All   Filter: None   Advanced

Name	Label	Status	Prepared	Timestamp	Quality	Type
⊞ SEL_351S_Feeder_4_SEL.FM_INST_52A	Feeder 4, Binaries	false		2017-04-03 20:57:31.108961	invalid	SPS
⊞ SEL_351S_Feeder_4_SEL.FM_INST_79LO	Feeder 4, Binaries	false		2017-04-03 20:57:31.108961	invalid	SPS
⊞ SEL_351S_Feeder_4_SEL.FM_INST_IA	Feeder 4, Currents	0.0@ 0.0		2017-04-03 20:57:31.108961	invalid	CMV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_IB	Feeder 4, Currents	0.0@ 0.0		2017-04-03 20:57:31.108961	invalid	CMV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_IC	Feeder 4, Currents	0.0@ 0.0		2017-04-03 20:57:31.108961	invalid	CMV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_P_WATTS	Feeder 4	0.0		2017-04-03 20:57:31.108961	invalid	MV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_Q_VARS	Feeder 4	0.0		2017-04-03 20:57:31.108961	invalid	MV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_RMB1A	Feeder 4, Binaries, High Speed	false		2017-04-03 20:57:31.108961	invalid	SPS
⊞ SEL_351S_Feeder_4_SEL.FM_INST_RMB2A	Feeder 4, Binaries, High Speed	false		2017-04-03 20:57:31.108961	invalid	SPS
⊞ SEL_351S_Feeder_4_SEL.FM_INST_TMB1A	Feeder 4, Binaries, High Speed	false		2017-04-03 20:57:31.108961	invalid	SPS
⊞ SEL_351S_Feeder_4_SEL.FM_INST_TMB2A	Feeder 4, Binaries, High Speed	false		2017-04-03 20:57:31.108961	invalid	SPS
⊞ SEL_351S_Feeder_4_SEL.FM_INST_VA	Feeder 4, Voltages	0.0@ 0.0		2017-04-03 20:57:31.108961	invalid	CMV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_VB	Feeder 4, Voltages	0.0@ 0.0		2017-04-03 20:57:31.108961	invalid	CMV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_VBAT	Feeder 4, Voltages	0.0		2017-04-03 20:57:31.108961	invalid	MV
⊞ SEL_351S_Feeder_4_SEL.FM_INST_VC	POU Pins	0.0@ 0.0		2017-04-03 20:57:31.108961	invalid	CMV
SEL_351S_Feeder_4_SEL_POU.Auto_Configuration_Failure	POU Pins	false		---	---	BOOL
SEL_351S_Feeder_4_SEL_POU.Event_Collection_Count	POU Pins	0		---	---	UDINT
SEL_351S_Feeder_4_SEL_POU.Initiate_Auto_Configuration	POU Pins	false		---	---	BOOL
SEL_351S_Feeder_4_SEL_POU.Message_Received_Count	POU Pins	0		---	---	UDINT
SEL_351S_Feeder_4_SEL_POU.Message_Sent_Count	POU Pins	4871		---	---	UDINT

Figure 5.5 Live Data Page

Tag structures can be expanded by clicking the + icon to the left of the tag name. Tags can be filtered using the filter controls at the top of the page. Basic and Advanced filter controls can be toggled by clicking the **Basic** and **Advanced** links to the right of the filter controls. Data can be filtered by name, label (as defined in either the Tag Processor or user logic), status, quality, or type. Filters can be saved and reused. Saved filters are saved on the RTAC itself and are made available across user accounts and web browsers.

When using the advanced filter input more complex filter rules can be written. See *Table 5.2* for special operators available for advanced filter expressions.

Table 5.2 Live Data Advanced Filtering Operators

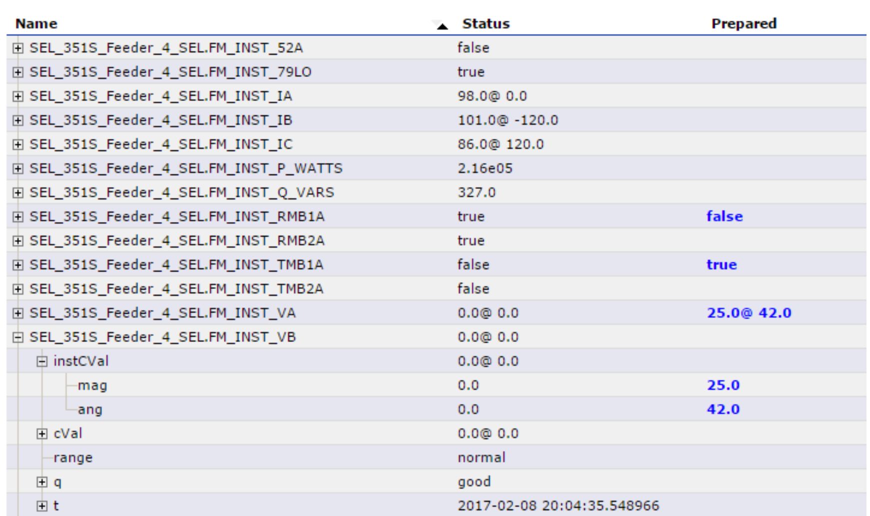
Character	Function	Example
OR	Logical OR	Name ~ 'IA' OR Name ~ 'IB'
AND	Logical AND	Name ~ '52A' AND Status ~ 'false'
NOT	Logical NOT	Name ~ '52A' AND NOT Status ~ 'false'
=	Equals	Status = 'false'
~	Contains	Name ~ '79LO'
in	Is left side found in right side list	Name ~ '79LO' AND Label in ('Binaries', 'Analogs')

Forcing Values

Tag values can be forced from the webpage by using the **Prepared** column in the Live Data table. Because forcing from the Live Data page is functionally equivalent to forcing values in Custom Logic, see *Using Online Debug on page 599*. Most elements of the tag can be forced in this manner, including elements of the quality and time substructures (if present). When preparing to force an analog value that includes both magnitude and angle, you can either expand the instMag substructure and prepare both magnitude and angle in their own cell, or you can type **mag@ang** in the cell associated with the top level of the tag. *Figure 5.6* illustrates both force preparation methods, as well as Boolean force preparation.

NOTE

Use caution when forcing values. Depending on the RTAC configuration, it may be possible to send controls or operate contacts.



Name	Status	Prepared
SEL_351S_Feeder_4_SEL.FM_INST_52A	false	
SEL_351S_Feeder_4_SEL.FM_INST_79LO	true	
SEL_351S_Feeder_4_SEL.FM_INST_IA	98.0@ 0.0	
SEL_351S_Feeder_4_SEL.FM_INST_IB	101.0@ -120.0	
SEL_351S_Feeder_4_SEL.FM_INST_IC	86.0@ 120.0	
SEL_351S_Feeder_4_SEL.FM_INST_P_WATTS	2.16e05	
SEL_351S_Feeder_4_SEL.FM_INST_Q_VARS	327.0	
SEL_351S_Feeder_4_SEL.FM_INST_RMB1A	true	false
SEL_351S_Feeder_4_SEL.FM_INST_RMB2A	true	
SEL_351S_Feeder_4_SEL.FM_INST_TMB1A	false	true
SEL_351S_Feeder_4_SEL.FM_INST_TMB2A	false	
SEL_351S_Feeder_4_SEL.FM_INST_VA	0.0@ 0.0	25.0@ 42.0
SEL_351S_Feeder_4_SEL.FM_INST_VB	0.0@ 0.0	
instCVal	0.0@ 0.0	
mag	0.0	25.0
ang	0.0	42.0
cVal	0.0@ 0.0	
range	normal	
q	good	
t	2017-02-08 20:04:35.548966	

Figure 5.6 Forcing Live Data

Once all force values have been prepared, press **Force**. *Figure 5.7* shows an example of the Live Data page with values forced.

Name	Status	Prepared
⊞ SEL_351S_Feeder_4_SEL.FM_INST_52A	false	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_79LO	true	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_IA	98.0@ 0.0	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_IB	101.0@ -120.0	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_IC	86.0@ 120.0	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_P_WATTS	2.16e05	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_Q_VARS	327.0	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_RMB1A	F false	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_RMB2A	true	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_TMB1A	F true	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_TMB2A	false	
⊞ SEL_351S_Feeder_4_SEL.FM_INST_VA	F 25.0@ 42.0	
⊟ SEL_351S_Feeder_4_SEL.FM_INST_VB	25.0@ 42.0	
⊟ instCVal	25.0@ 42.0	
mag	F 25.0	
ang	F 42.0	
⊞ cVal	0.0@ 0.0	
range	normal	
⊞ q	good	
⊞ t	2017-02-08 20:04:35.548966	

Figure 5.7 Live Data After Force

Press **Unforce** to release the force but leave the values in place until they are overwritten by some other mechanism, or press **Unforce & Restore** to release the force and restore the tags to their preforce values.

Adding Tags to Live Data

Tags can be added to the Live Data page by using either settings in the Tag Processor or code in custom logic. The following data types are supported (see *Data Types on page 747* for additional information about these types):

ACD	LEDC
ACT	LINT
APC	LMV
BCR	LREAL
BCRC	LWORD
BOOL	MDBC
BYTE	MRBC
CMV	MV
DATE	REAL
dateTime_T	SBRC
DBPOS	SPC
DINT	SPS
DNPC	SRBC
DPC	STRC
DPS	STRING
DT	TIM
DWORD	TIME
I870DC	timeStamp_t

I870SC	UDINT
INC	UINT
INS	ULINT
INT	UNC
IOC	UNS
LBCR	USINT
LBCRC	WORD
vector_t	

To add tags to Live Data from the Tag Processor, use the **Live Data Enabled** column. Set the value of this column to **Forceable** to allow the value to be forced from the **Live Data** page, or **Viewable** to make the value read-only on the **Live Data** page. Use the **Live Data Labels** column to configure a comma-separated list of labels (limit 255 characters per label) to associate the tag with. These labels can be used to sort data on the Live Data page. *Figure 5.8* provides an example of Live Data configuration in the Tag Processor.

Build	Destination Tag Name	DT Data Type	Source Expression	SE Data Type	Live Data Enabled	Live Data Labels
True					False	
True	SEL_351S_Feeder_4_SEL.FM_INST_52A	SPS			Forceable	'Feeder 4, Binaries'
True	SEL_351S_Feeder_4_SEL.FM_INST_79LO	SPS			Forceable	'Feeder 4, Binaries'
True	SEL_351S_Feeder_4_SEL.FM_INST_RMB1A	SPS			Forceable	'Feeder 4, Binaries, High Speed'
True	SEL_351S_Feeder_4_SEL.FM_INST_RMB2A	SPS			Forceable	'Feeder 4, Binaries, High Speed'
True	SEL_351S_Feeder_4_SEL.FM_INST_TMB1A	SPS			Forceable	'Feeder 4, Binaries, High Speed'
True	SEL_351S_Feeder_4_SEL.FM_INST_TMB2A	SPS			Forceable	'Feeder 4, Binaries, High Speed'
True	SEL_351S_Feeder_4_SEL.FM_INST_IA	CMV			Forceable	'Feeder 4, Currents'
True	SEL_351S_Feeder_4_SEL.FM_INST_IB	CMV			Forceable	'Feeder 4, Currents'
True	SEL_351S_Feeder_4_SEL.FM_INST_IC	CMV			Forceable	'Feeder 4, Currents'
True	SEL_351S_Feeder_4_SEL.FM_INST_P_WATTS	MV			Forceable	'Feeder 4'
True	SEL_351S_Feeder_4_SEL.FM_INST_Q_VARS	MV			Forceable	'Feeder 4'
True	SEL_351S_Feeder_4_SEL.FM_INST_VA	CMV			Forceable	'Feeder 4, Voltages'
True	SEL_351S_Feeder_4_SEL.FM_INST_VB	CMV			Forceable	'Feeder 4, Voltages'
True	SEL_351S_Feeder_4_SEL.FM_INST_VBAT	MV			Forceable	'Feeder 4, Voltages'
True	SEL_351S_Feeder_4_SEL.FM_INST_VC	CMV			Forceable	'POU Pins'
True	SEL_351S_Feeder_4_SEL_POU.Auto_Configur...	BOOL			Viewable	'POU Pins'
True	SEL_351S_Feeder_4_SEL_POU.Initiate_Auto_...	BOOL			Viewable	'POU Pins'
True	SEL_351S_Feeder_4_SEL_POU.Event_Collecti...	UDINT			Viewable	'POU Pins'
True	SEL_351S_Feeder_4_SEL_POU.Message_Rece...	UDINT			Viewable	'POU Pins'
True	SEL_351S_Feeder_4_SEL_POU.Message_Sent...	UDINT			Viewable	'POU Pins'

Figure 5.8 Enable Live Data in the Tag Processor

To add tags to Live Data from user logic, use the LiveData function block in a custom logic program. The LiveData function block requires exactly six inputs at declaration. *Figure 5.9* provides an example of how to declare an instance of this function block.

```
VAR
    Recloser52A : LiveData(
        ADR(SEL_651RA_SEL.FM_INST_52A3P),   // Address of tag to be added to Live Data
        '651RA Breaker Status',             // Text to be used on Live Data page
        'SPS',                              // Data type
        '651RA, Binaries',                  // Label list (comma separated)
        TRUE,                               // Forceable (TRUE = yes)
        0);                                 // Task ID (0 = Main Task, 1 = Automation Task)
END_VAR
```

Figure 5.9 Example LiveData Function Block Declaration

The LiveData function block does not need to be called or executed for data to appear on the Live Data page; it must only be declared. However, you can control some aspects of the value's appearance on the Live Data page, and doing this requires calling the function block. *Figure 5.10* provides an example of how to control the visibility and the color of data when displayed on the Live Data page.

```
PROGRAM LiveDataProgram
VAR
    Recloser52A : LiveData(
        ADR(SEL_651RA_SEL.FM_INST_52A3P),    // Address of tag to be added to Live Data
        '651RA Breaker Status',              // Text to be used on Live Data page
        'SPS',                               // Data type
        '651RA, Binaries',                   // Label list (comma separated)
        TRUE,                                // Forceable (TRUE = yes)
        0);                                  // Task ID (0 = Main Task, 1 = Automation Task)

    visibility          : BOOL;              // TRUE = visible, FALSE = invisible
    breakerDataColor    : colorRG_t := RED; // Options: red, green, off
END_VAR
```

```
Recloser52A(EN := visibility, Color := breakerDataColor);
```

Figure 5.10 Control Visibility and Color of Data on Live Data Page

Log Settings

Default Log Entries

Within the Tag Processor, the RTAC has several default log entries of certain system tags to provide common security and audit logs. Default log entries include user log in/log off, user settings changes, the number of logon attempts for a user, and other audit-related tags. These default log items are valuable for security and for troubleshooting system issues and should not be removed.

You can configure the RTAC to log changes of value with time-stamped log messages. The log messages are visible in the RTAC web interface and can be downloaded as a comma-separated variable (CSV) file and viewed via ODBC. You can use the Tag Processor to configure all log settings or you can configure logging in user-defined programs.

The RTAC supports logging of the following tag types:

- **SPS**: log triggered by change of state of .stVal. You must enter a logging on and logging off message for the RTAC to log the change. The function block is called Logger_SPS.
- **MV**: log triggered by change of the instMag value (based on range parameters in the tag configuration or in the Tag Processor). Logging on and off messages are not configurable, because the range attribute is the message. For example, if the value exceeds the .rangeC.hhLim, the log contains the time of the excursion and the message "High_High." The function block is called Logger_MV.
- **CMV**: log triggered by change of the instCval.mag value (based on range parameters in the tag configuration or in the Tag Processor). Logging on and off messages are not configurable, because the range attribute is the message. The function block is called Logger_CMV.

- **INS**: log triggered by change of the .stVal value (based on range parameters in the tag configuration or in the Tag Processor). Logging on and off messages are not configurable, because the range attribute is the message. The function block is called Logger_INS.
- **STR**: log triggered by change of string value. Configure a logging on and logging off message. The function block is called Logger_STR.
- **BOOL**: log triggered by change of Boolean value. Configure a logging on and logging off message. The function block name is Logger_Basic.
- **Time Change Trigger**: log triggered by the change of the configure time source or the time of the tag if no source is configured. Configure a Time Change Trigger message and check the box for Time Change Trigger. Time change logs and logs based on tag values are mutually exclusive.

Use the following steps to enable logging of a tag.

Step 1. Click on the **Tag Processor** tab.

Step 2. Right-click on the **Tag Processor** column header, and select **Logging Layout** from the pop-up window. This will show columns related to logging configuration.

NOTE
You must configure the Logging On message, Logging Off message, or the Logging Time Change Trigger message to generate log entries. The RTAC does not log blank messages.

Step 3. Place the tag you want to log in the **Destination Tag Name** column.

Step 4. Set **Logging Enable = True**.

Step 5. Set up other parameters you want to log based on time, value change, edge trigger, etc.

See *Section 3: Tag Processor* for a description of each logging parameter in the Tag Processor grid.

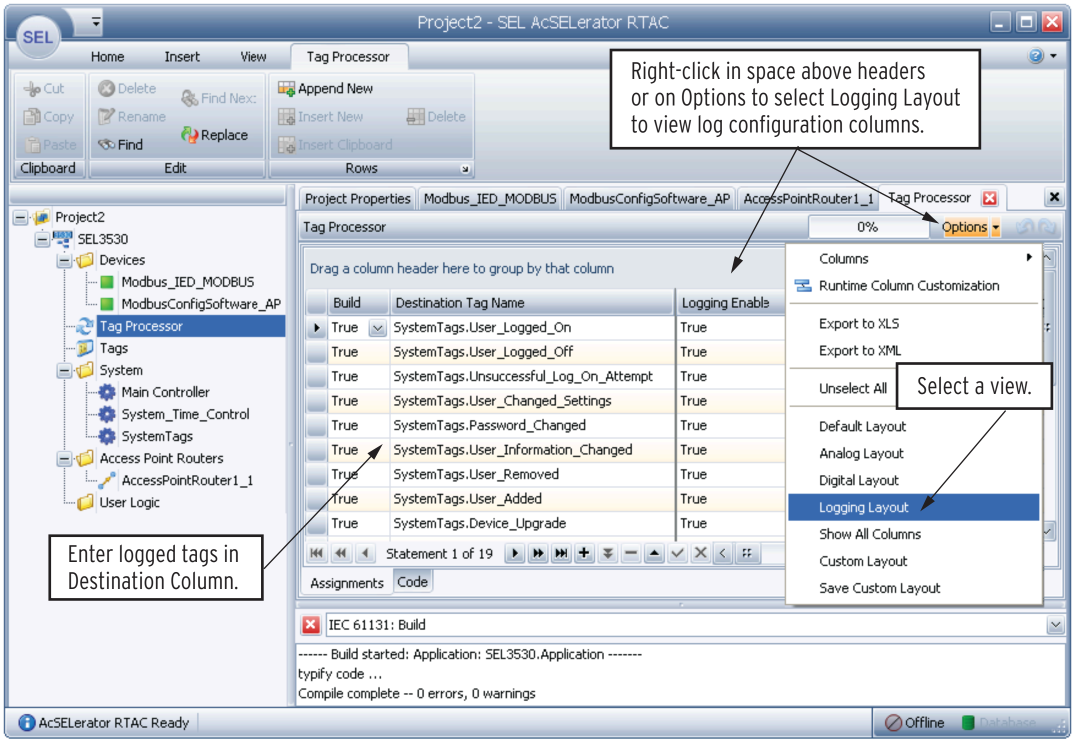


Figure 5.11 Logging a Tag

Other Settings

You can use other logging settings in the Tag Processor to do the following:

- Set logging categories and priorities.
- Enable logging of alarms based on high and low alarm limits, change of time or change of state. The logged alarms appear in the event log and flash when in alarm state.

NOTE
Log messages must contain only US-ASCII characters.

- Filter alarming and logging with an antichatter filter. This defines the maximum number of changes (Logging Chatter Counts) within a period of time (Logging Chatter Time) that the RTAC permits before it considers the point to be chattering. When the Tag Processor determines the point is chattering, it will remove all but the first entry of this chattering point and log a message indicating this action has been taken.

Example 5.1 Log Lamp Test Button Usage

This example will illustrate how to log a status change. We will configure the Tag Processor to create a log entry each time you depress or release the lamp test button. See *Figure 5.12* for the completed example. We show here only columns necessary for this example.

Step 1. In the Tag Processor, insert the tag SystemTags.Lamp_Test_Button_Status.stVal into an open line in the Destination Tag Name column.

This is the status indication of whether the lamp test button has been depressed.

Step 2. Right-click in space above headers, and select **Logging Layout** to show only the columns related to logging.

Step 3. Scroll to the right to complete the following configuration changes:

- **Logging Enable**: Set to True to enable logging of this tag.
- **Logging Alarm Enabled**: Optional. Set to True if you want the RTAC to also display the log entry in the Alarm Summary of the RTAC web interface when it logs the point. Set to False to log only.
- **Logging Alarm State**: Optional if Logging Alarm Enabled is True. Determines if the point goes into alarm when the value is True (button depressed) or when the value is False (button released). A pop-up warning appears on the RTAC web interface to notify the user whenever alarms occur.
- **Logging Category**: Set to any string enclosed in single quote marks ('Internal', for example). You can use this category to sort logs when viewing them.
- **Logging On Message**: Set to any string enclosed in single quote marks ('Lamp button depressed', for example). The RTAC logs this message when the status is True.
- **Logging Off Message**: Set to any string enclosed in single quote marks ('Lamp button released', for example). The RTAC logs this message when the status is False.

Step 4. Save the project and send it to the RTAC.

Step 5. When the transmission is finished, complete the steps under Viewing Logs Via Web Interface to open the web browser and view the RTAC logs.

Step 6. Depress the lamp test button long enough for the RTAC to detect the change (lamp test sampling is one second), and note the updated log message at the bottom of the RTAC webpage screen.

Step 7. Release the button, and note the updated log message reflecting that the button has been released.

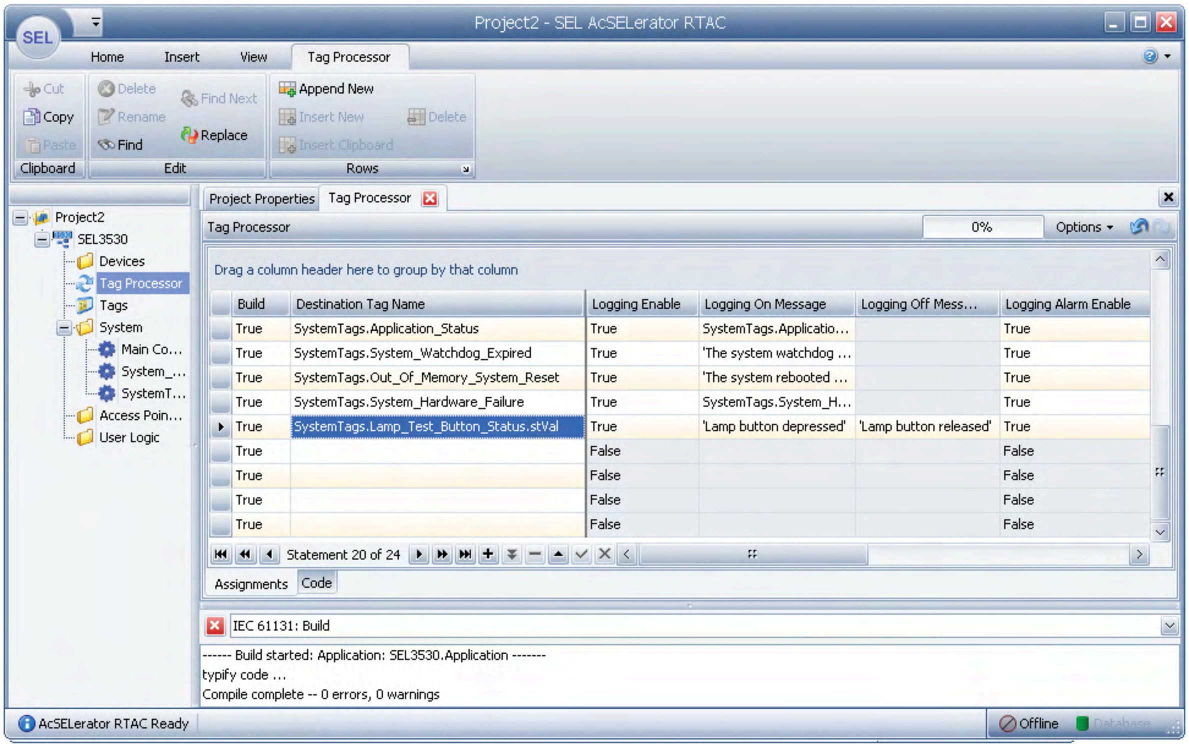


Figure 5.12 Example Log Entry

Example 5.2 Log an INS Value With Time Change Trigger

Normally, an INS value is logged based on the change of the range attributes. The resulting log entry shows the time, tag name, and range value (such as High, High_High, Low, etc.). This example uses the CONCAT function to log the message "My value changed. Value =" along with the INS value each time the time stamp changes.

Step 1. Right-click in the space above headers, and select the **Logging Layout** to show only the columns related to logging.

Step 2. Enter an INS tag name and stVal attribute in the Tag Processor Destination Tag Name column.

Step 3. Configure the following:

Setting	Value
Logging Enable	True
Logging Time Change Trigger	True
Logging Time Change Trigger Message	concat (('My value changed. Value = '), DINT_TO_STRING(test_point.stVal))

Each time the INS time value changes, the RTAC will log the defined message string along with the present INS stVal value.

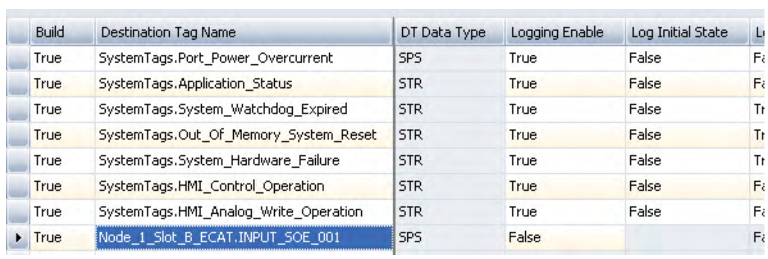
Example 5.3 Log a Digital Input in an SEL-2240 System

Example 5.1 illustrated how to log a tag that originates within the RTAC. The time stamp associated with such a point will be the RTAC system time when the log entry is created. This example will show how to log a tag along with a time stamp that is created independently from the RTAC logic engine. The illustration uses a digital input from the SEL-2240, connected via EtherCAT protocol. The same principles apply to other protocols, such as DNP or C37.118.

Step 1. Create a project that includes an I/O network and at least one SEL-2240 Digital Input Module. Refer to *EtherCAT on page 151* for specific configuration steps.

Step 2. In the Tag Processor, insert the SOE tag for input_001. As described in *DC Mode Processing (DC Control Voltage) on page 165*, this tag includes a time stamp that is accurate within 1 ms, regardless of the logic engine task cycle time. Verify that the debounce settings for this input are appropriate for the application.

As you can see in *Figure 5.13*, enter the tag name without the .stVal attribute. The SPS tag type includes a time attribute that will contain the original time stamp. If you enter the tag name with the .stVal attribute, the logged time stamp will be the RTAC system time when the log is created.



Build	Destination Tag Name	DT Data Type	Logging Enable	Log Initial State	L
True	SystemTags.Port_Power_Overcurrent	SPS	True	False	Fa
True	SystemTags.Application_Status	STR	True	False	Fa
True	SystemTags.System_Watchdog_Expired	STR	True	False	Tr
True	SystemTags.Out_Of_Memory_System_Reset	STR	True	False	Tr
True	SystemTags.System_Hardware_Failure	STR	True	False	Tr
True	SystemTags.HMI_Control_Operation	STR	True	False	Fa
True	SystemTags.HMI_Analog_Write_Operation	STR	True	False	Fa
True	Node_1_Slot_B_ECAT.INPUT_SOE_001	SPS	False		Fa

Figure 5.13 Example 5.3 Tag Entry

Step 3. Scroll to the right to complete the logging configuration changes you want. See *Example 5.1* for details.

Step 4. Save the project, and send it to the RTAC.

Step 5. Assert the contact input, and view the log by using the instructions shown in *Viewing Logs Via Web Interface on page 458*.

Viewing Logs Via Web Interface

You can view logs in either the RTAC's web interface or via ODBC. The web interface log entries contain all the data related to a particular log. Notice the log entry in *Figure 5.14* of several log in attempts. The username is left as **Unknown** until three consecutive attempts, at which time the user is locked out and the user's name is logged. **Feeder1 breaker**, configured to display when in alarm, is red when logged in alarm state and returns to black when the state returns to normal. The **Alarm Summary** is very similar to the SOE report with the exception that it contains only logged items that are configured with Logging Alarm Enabled, and it also automatically updates every 10 seconds to show any new or updated entries. Use the following steps to configure the log entry to show only columns you want:

Step 1. Connect to the RTAC through one of the Ethernet ports or the USB-B interface.

Step 2. Open a web browser.

Step 3. Enter **HTTPS://** followed by the RTAC Ethernet address.

NOTE
The RTAC and PC must be on the same Ethernet subnet to enable web communications.

Step 4. Log in to the RTAC web interface.

Step 5. Note the alarm summary at the bottom of each webpage.

Step 6. To view all logs, events, and alarms, click **Alarm Summary** or **SOE** under **Reports**.

Step 7. Filter the report as needed using the filtering categories located at the top of the page.

Step 8. Apply a check to any items on this list that you want to clear from the list or acknowledge.

Step 9. Select an action from the **Actions** drop-down list to apply to the entries that are checked.

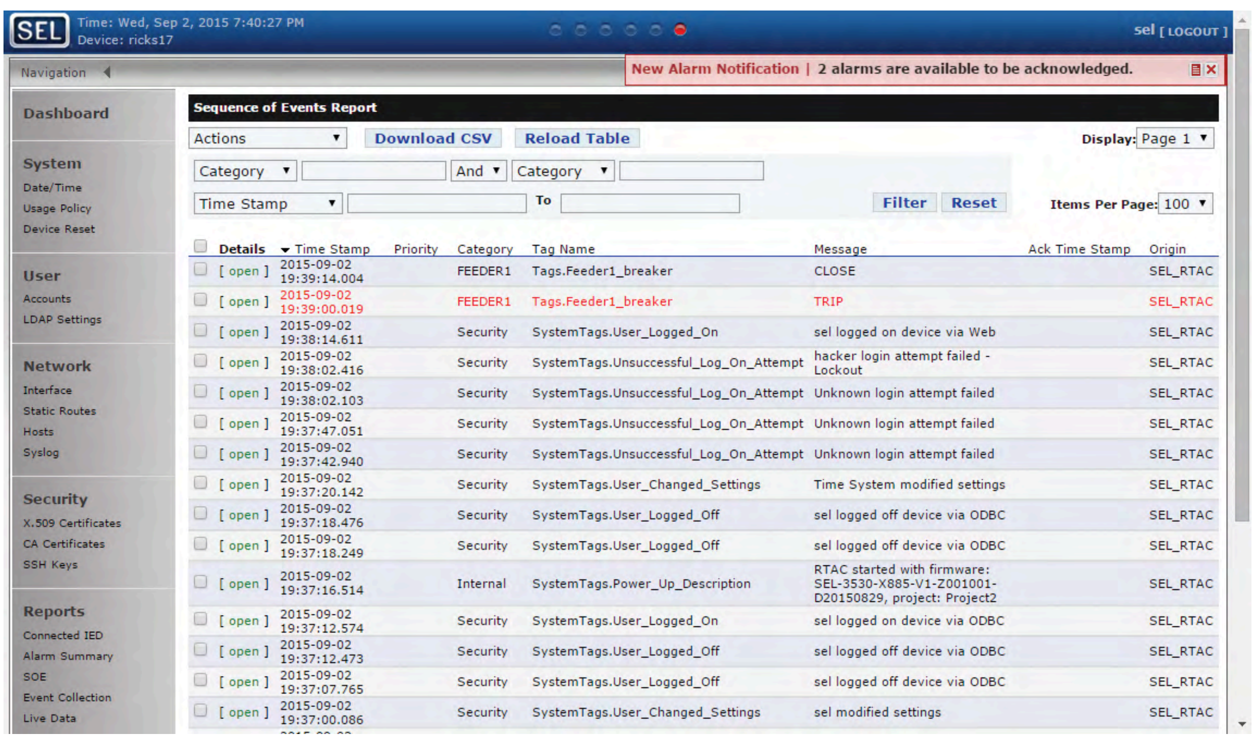


Figure 5.14 View SOE Summary

You can also clear all logs from the web interface in one step. Under the **System** heading, click on **Device Management**. Select the **Clear Logs** check box, and then click **Submit**. This operation cannot be undone.

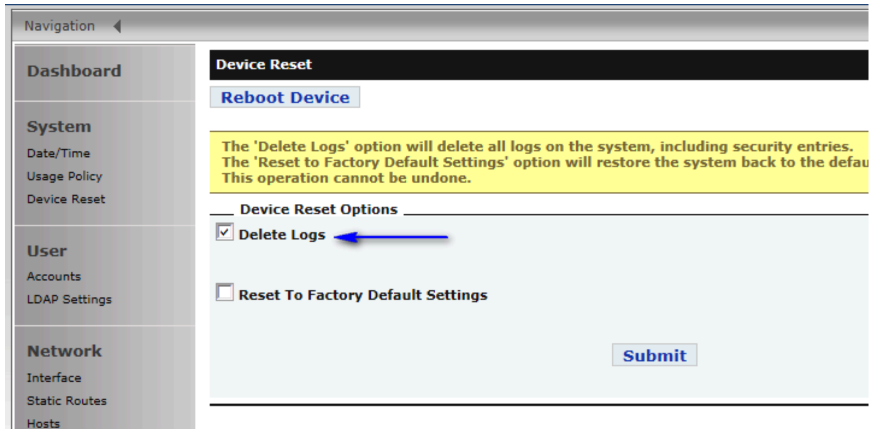


Figure 5.15 Clear Logs

Configure ODBC to View RTAC Database

Use Microsoft Access, Microsoft Excel, OpenOffice.org, or another ODBC-compatible program to view RTAC event logs through ODBC from an Ethernet-connected RTAC. The ODBC connection can also show current settings and other database tables from the connected RTAC.

To make an ODBC connection in Windows 7, do the following:

Step 1. Navigate to **Control Panel** > **Administrative Tools**.

Step 2. Open **Data Sources (ODBC)—32 Bit** to show the ODBC Data Source Administrator. If you have no such option, perform the following steps:

a. Copy and paste the Data Sources (ODBC).

b. Rename the copied Data Sources (ODBC) to Data Sources (ODBC)—32 Bit.

c. Right-click on the new shortcut, and click on **Properties**.

d. Change the information as follows:

Target: C:\Windows\SysWOW64\odbcad32.exe

Start in: C:\Windows\SysWOW64

e. Click **OK** to save the new properties.

f. Use this new shortcut for ODBC configuration RTAC.

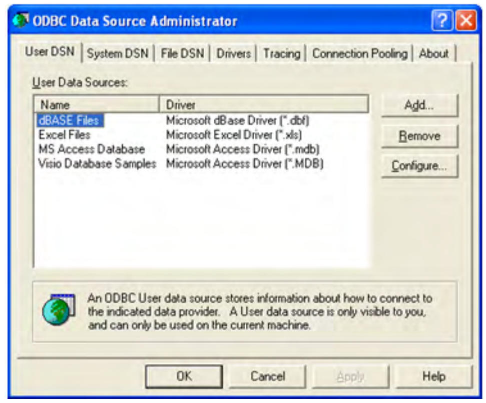


Figure 5.16 ODBC Administrator

Step 3. Select the **System DSN** tab and click **Add**.

Step 4. Scroll down and select **RTACSQL Unicode**.

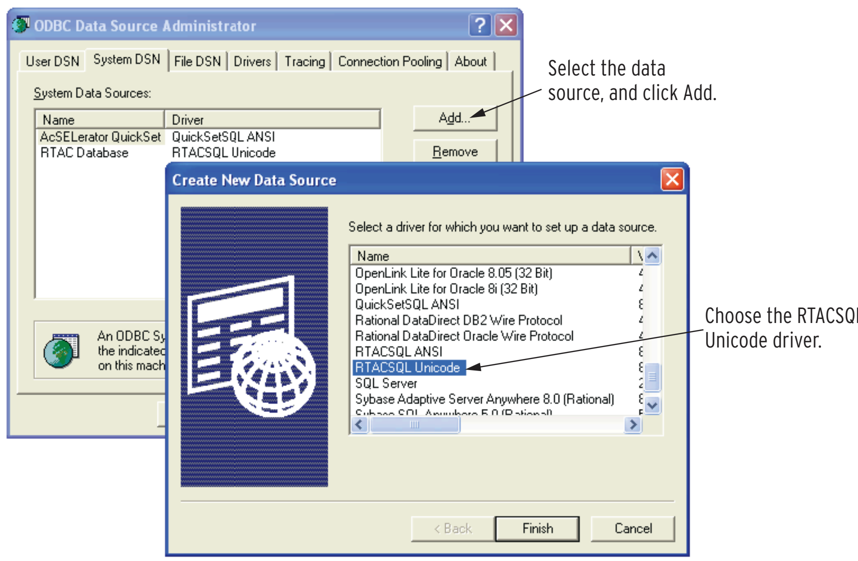


Figure 5.17 Add ODBC Source

Step 5. Configure the pop-up form as shown in *Figure 5.18*, substituting the IP address and login information for that to which your RTAC is configured, and click **Save**.

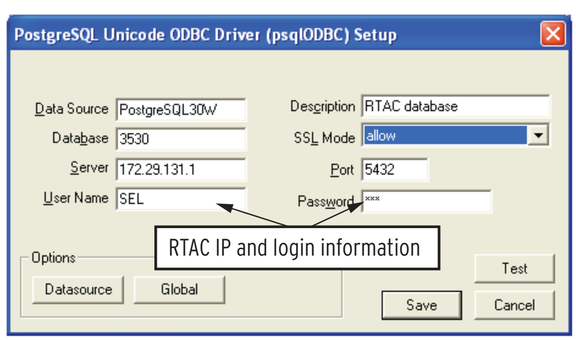


Figure 5.18 ODBC Setup Parameters

Step 6. Open Microsoft Excel, Access, etc., and set up an ODBC connection to postgreSQL30W. The log entries presently on that RTAC will appear in the spreadsheet.

Example 5.4 Use ODBC in Excel 2007

Any program that is ODBC aware can open the RTAC log database. Different programs and versions have somewhat different configurations. To set up the connection in Excel 2007, do the following:

Step 1. Open a new spreadsheet in Excel.

Step 2. Select the **Data** ribbon.

Step 3. Click the **From Other Sources** icon in the ribbon.

Step 4. Select **From Microsoft Query**.

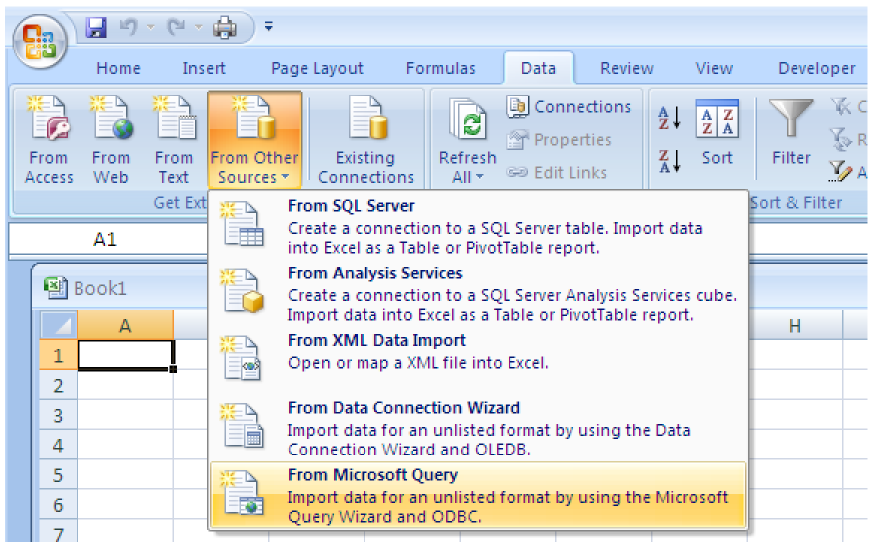


Figure 5.19 Excel 2007 ODBC

Step 5. Select data source and which data columns to obtain the data from the RTAC.

Sources:

- PostgreSQL30W = settings and event tables from the database of a connected RTAC
- RTAC Database = static configuration data from ACSELERATOR RTAC SEL-5033 Software project database residing on the PC

Tables:

- Events = event log table from a connected RTAC
- Other tables = settings tables in the database of the connected RTAC

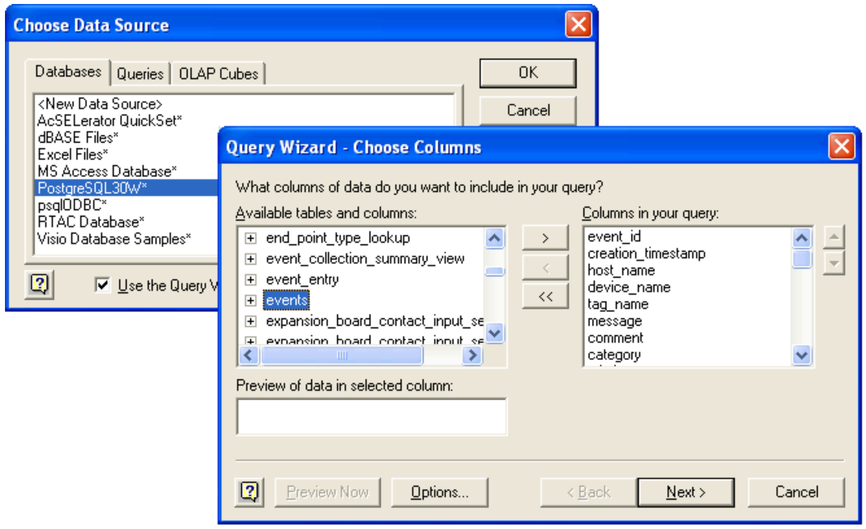


Figure 5.20 Excel 2007 ODBC Table Selection

Step 6. Click on **Next** until you get a screen with the Finish button.

Step 7. Click **Finish**.

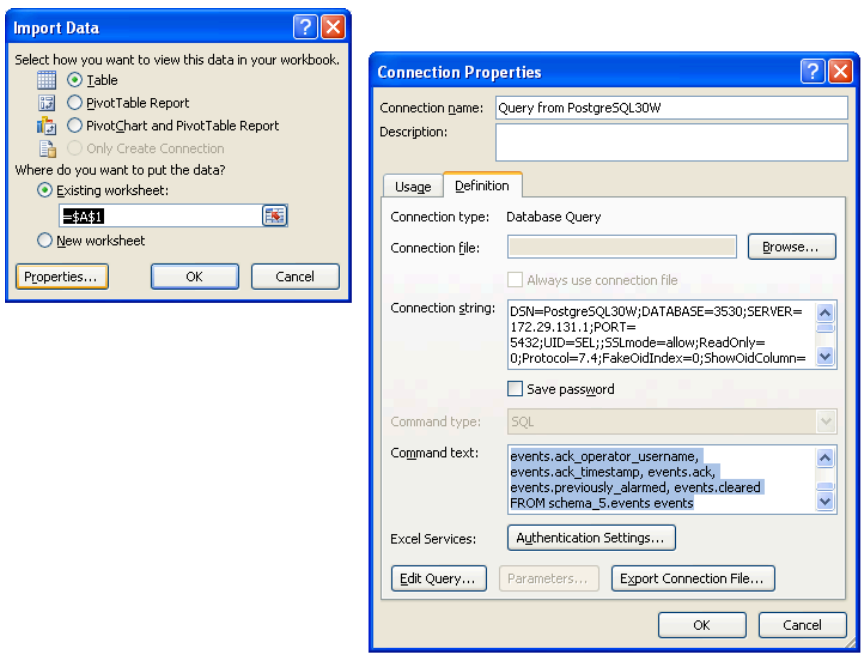


Figure 5.21 Excel 2007 ODBC Connection File

Step 8. On the Import Data screen, press **Properties**.

Excel will remove millisecond time resolution from the data. To prevent it from doing this, you can edit the select statement and cast the time as a character. Find the two time values and replace them using cast statements, then press **OK**.

NOTE

The value for creation_timestamp is when the event was placed into the event log. The value t_value is the actual time of the event that triggered the log.

Replace:

events.creation_timestamp

With:

cast(events.creation_timestamp AS CHAR(30))

Replace:

events.t_value

With:

cast(events.t_value AS CHAR(30))

Step 9. Format the time by right-clicking on the t_value column header, select **Format Column > Custom**, then enter **yyyy/mm/dd hh:mm:ss.000** and press **Ok**.

Step 10. Press **Refresh All** in the **Data** ribbon at any time to refresh the data the spreadsheet displays.

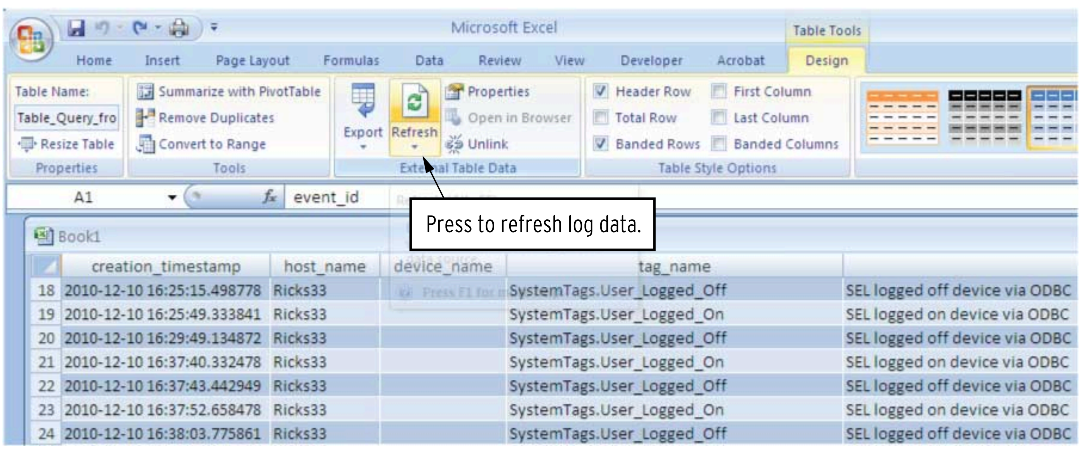


Figure 5.22 Excel 2007 ODBC Event Log

RAID Support and Configuration

RAID—Redundant Array of Independent Disks

The SEL-3555 provides RAID-1 storage management for the RTAC's file system data. RAID-1 manages mirrored copies of its file system data on two disks and enhances the RTAC's data storage resiliency. Only SEL-3555 RTACs that include three or more solid-state drives (SSDs) are compatible with RAID data management.

Setting and Initializing RAID

Configure RAID settings using the RTAC web interface. Select the **Device Management** link from the navigation panel to view and edit the RAID settings. The system disk (designated as "System") includes the RTAC firmware image and is not compatible with RAID. *Figure 5.23* shows the RAID settings for an RTAC with four SSDs.

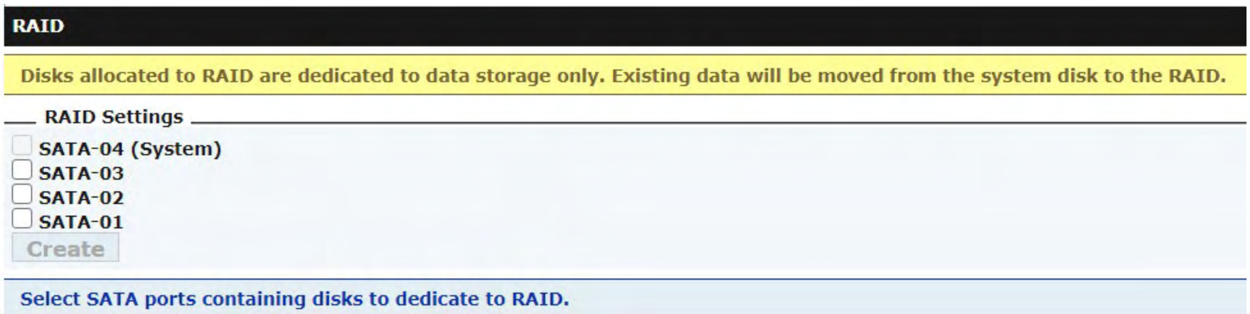


Figure 5.23 RAID Setting View With All SATA Ports Occupied

Activate RAID storage using at least two of the non-System SATA disks. Select the desired SATA ports to dedicate to RAID and select the **Create** button. Once initialized, the RTAC will begin synchronizing all RAID components. The following table lists the possible RAID states based on the number of operational disks in the RAID configuration:

Number of Operational Disks in RAID	Normal Operation	Error State(s)
One	RAID status is *Degraded*, indicating that RAID is operational after completing initialization, but only one disk is present and redundancy is not available.	If a RAID disk fails, the RAID status is *Failed*.
Two	RAID status is *Ready*, indicating that RAID has completed initializing and is fully synchronized.	If both RAID disks fail, the RAID status is *Failed*.
Three	RAID status is *Ready*, indicating that RAID has completed initializing and is fully synchronized. In this state, the RTAC uses a third disk designated as a spare for automatic RAID recovery when one of the active disks fail. RAID activates a *Rebuild* to synchronize data to the spare disk if either of the active disks fail.	If all RAID disks fail, the RAID status is *Failed*.

Identifying a Failed Disk

The following example provides methods for identifying a SATA disk failure. The example system consists of a three-disk RAID where SATA Ports 1 and 2 are synchronized and SATA Port 3 is available as a spare disk. In this example, the disk in SATA-01 fails and RAID rebuilds the spare SATA-03 for mirroring.

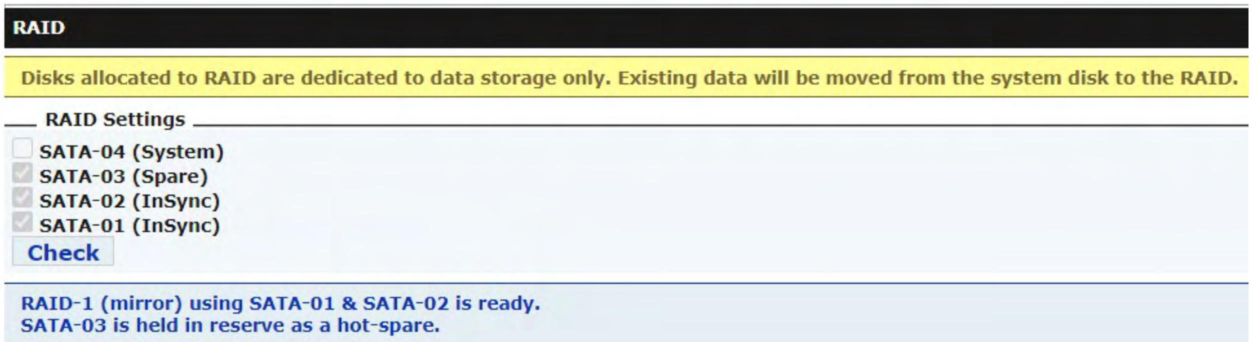


Figure 5.24 Web Page View Normal Operation Three-Disk RAID

The following three methods provides indication that a SATA 1 disk failed:

1. The RTAC's Sequence of Events (SOE) by default logs any changes in the RAID status. In this three-disk RAID example, the RAID status message will be "RAID-1 (mirror) using SATA-02 and SATA-03 is rebuilding". If a RAID status message SOE log appears, follow the next two steps to check SATA port status.
2. SATA port status information is available in the RTAC's Web Interface via the **Device Management** link. The RTAC Web Page provides the individual SATA port status. The failed disk is indicated by "Failed" next to the SATA port. See *RAID and SATA Status Information on page 467* for further details regarding status messages.

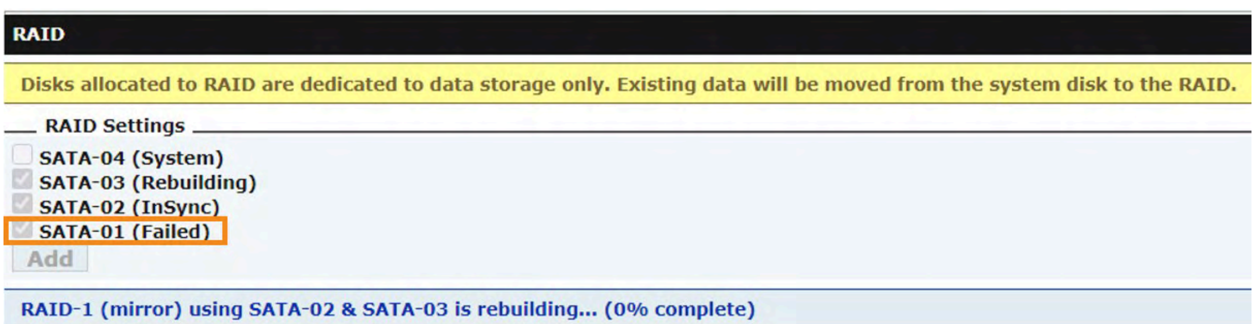


Figure 5.25 Web Interface SATA Port Failure Indication

3. The RTAC's SystemTags also provide SATA port status information. If connected to RTAC via ACSELERATOR RTAC, view the RTAC's SATA RAID Status system tags.

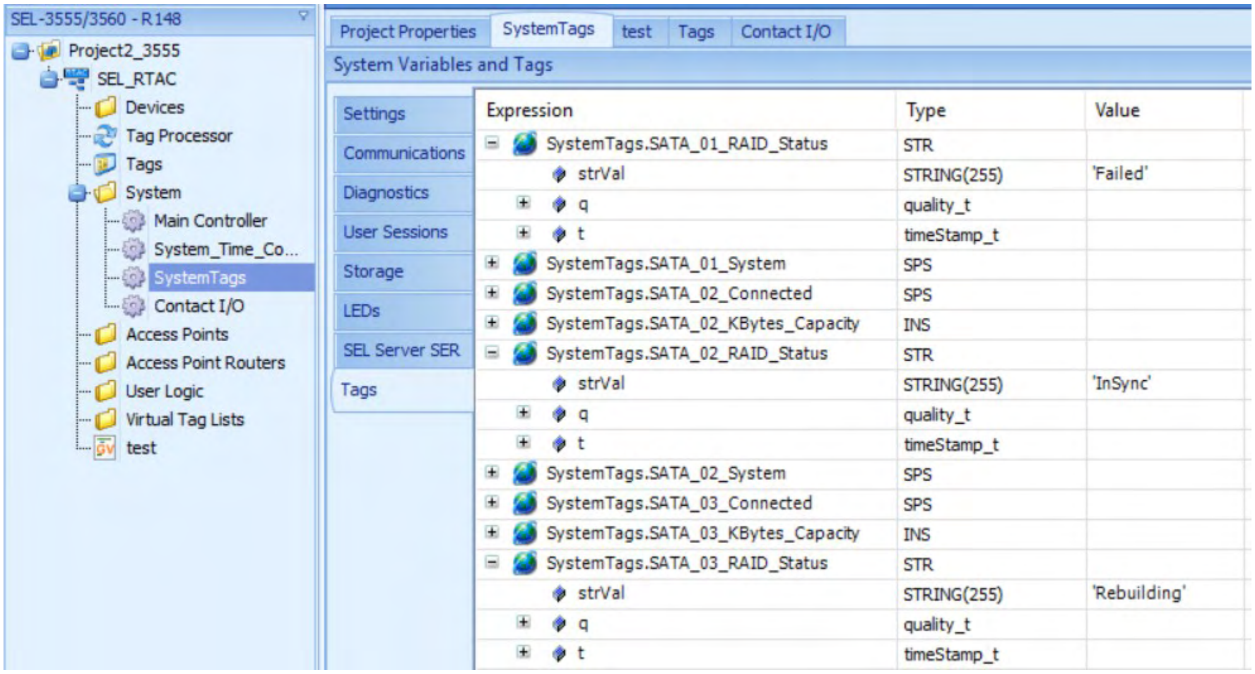


Figure 5.26 ACSELERATOR RTAC SATA Port SystemTags View

After identifying a failed disk, see *SATA Drives* in the *SEL-3555 Instruction Manual* for details on removing and installing a new disk. Once a failed disk has been replaced, RAID will automatically initialize and begin rebuilding a copy of the file system data on the new disk.

RAID and SATA Status Information

The following table describes the possible states that RAID may operate in:

Status	Description
Initializing	RAID is created and synchronizing for the first time.
Ready	RAID has finished initializing/rebuilding and is ready for operation.
Degraded	RAID is operational but is missing disks.
Rebuilding	RAID was previously Degraded and is now re-synchronizing.
Failed	All the disks in the RAID have failed or are missing.
Checking	The RAID is operational, and an integrity check is being performed.

Additionally, the following table describes the possible states that SATA bays may operate in:

Status	Description
InSync	The disk is in the RAID and is fully synchronized.
Rebuilding	The disk is currently rebuilding in the RAID.

Status	Description
Spare	The disk is reserved as a hot-spare for the RAID.
Failed	The disk was in the RAID and failed.

S E C T I O N   6

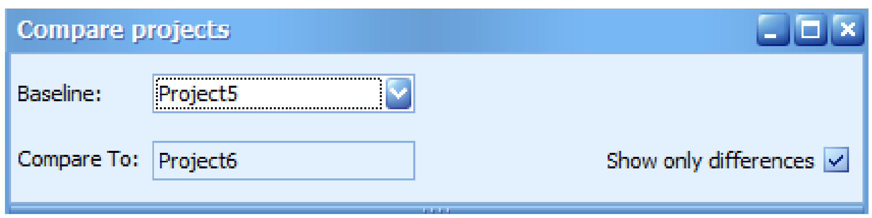
Compare ACSELERATOR RTAC Configurations

Overview

To compare two projects in ACSELERATOR RTAC SEL-5033 Software, select the desired projects to compare, and then right-click and select **Compare** from the menu list.

When the comparison is done, a window appears showing the POUs that registered a difference.

In the top portion of the window the **Baseline** project can be changed. This will change the detection of additional missing POU classifications in the list below. There is also an option to show all POUs in the configuration regardless of whether differences are detected.



The bottom portion of the window shows the list of POUs that have detected differences (if the **Show only differences** check box is selected). For items that are listed as Comparable, double-click or right-click to see a more detailed comparison screen.

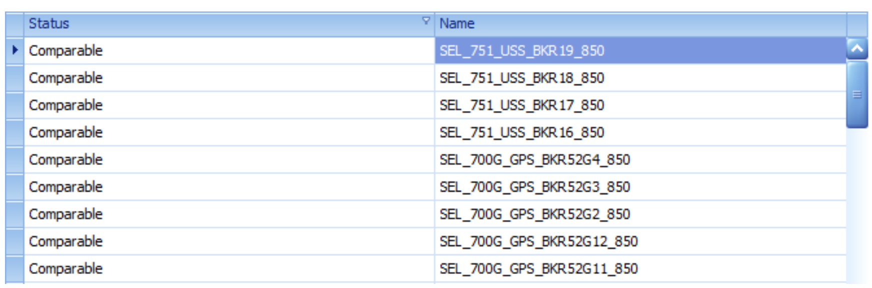


Table 6.1 Status Values

Status Name	Description
Comparable	Differences were detected (double-click for a detailed view).
Added	The baseline configuration contains a POU that the **Compare To** project did not.
Removed	The baseline configuration does not contain a POU that the **Compare To** project did.
Over threshold percentage	This designation indicates that although the POU name is the same, there are so many differences detected that it is likely that the POU is *not* the same type in both configurations. This is designated by a % difference detected, which is configurable under **SEL > Options > Preferences**.

Compare Window Layout

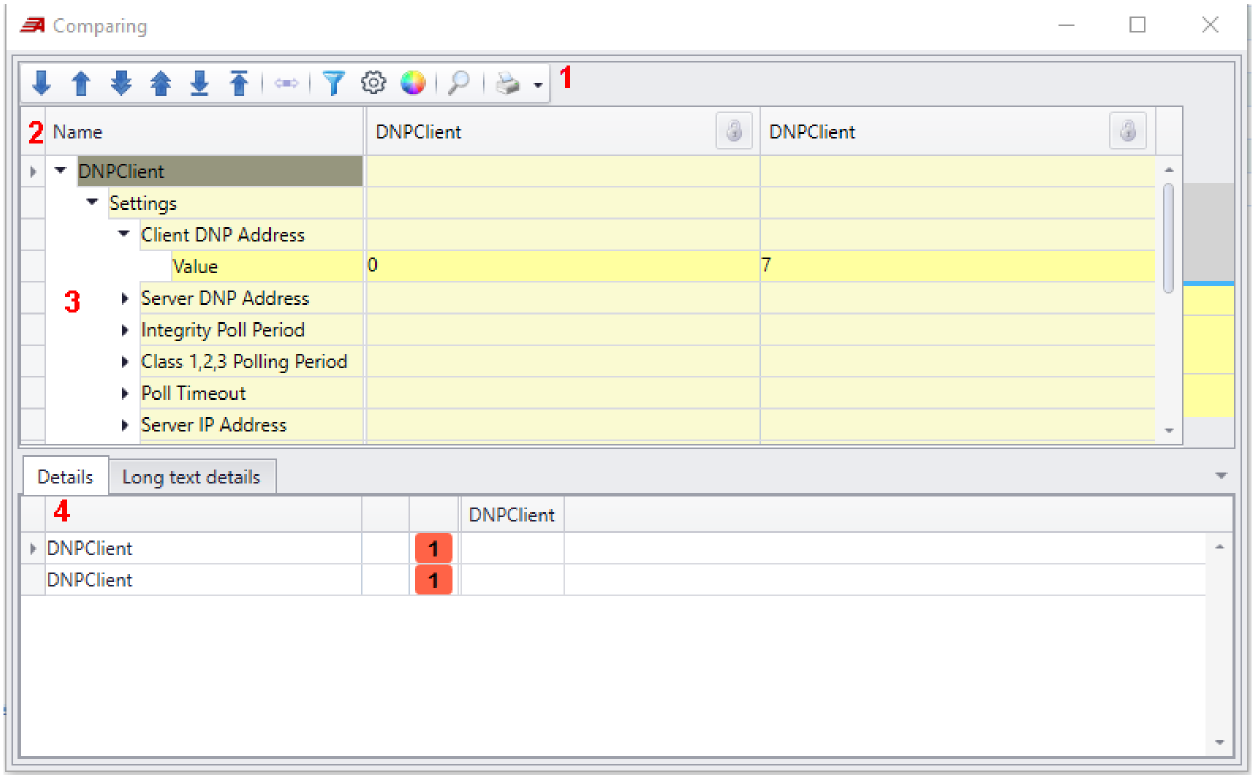


Figure 6.1 Compare Window Components

1. **Navigation and Filter Options**: This section includes the navigation and filter options for the Compare Results window.
2. **Compare Node Names**: This section displays the names of selected nodes and enables the Save icon for those nodes that have changes.
3. **Compare Results Window**: This section displays the node information and attached settings and highlights the differences.
4. **Difference Output**: This section displays the values for and differences, if any, from the selected row in the Compare Results window.

Filter Options

Hide Similar Values

This option filters out any similar values across the rows of settings being compared so that only rows with differences display.

Hide Inactive (Hidden) Rows

This option filters out device node rows rendered inactive and unchangeable because of either a part number or setting rule.

Reset to Default

This option resets the filters to original ACSELERATOR RTAC default values.

Comparison Options

Ignore Case Differences

When the **Ignore case differences** option is enabled, Device Manager treats values that differ only in their use of uppercase or lowercase letters as if they are identical.

For example, "ABC" and "abc" compare identically when the Ignore case differences option is enabled.

Ignore White Spaces Differences

When the **Ignore white spaces** differences option is enabled, Device Manager ignores whitespace characters when performing comparisons of individual items between entities.

For example, "ABC" and "A B C" compare identically when the **Ignore white spaces** differences option is enabled.

Ignore Missing Values

When the **Ignore missing values** option is enabled, missing entity items are ignored.

Compare as Text

When the **Compare as text** option is enabled, all entity items are compared alphanumerically regardless of item data type.

For example, Numeric items "01.1" and "1.10" do not compare equally when you use the **Compare as text** option, but they are equal when you use numeric comparison.

Ignore Differences in Setting Comments

When the **Ignore differences in setting comments** option is enabled, any differences in comments included in the attached settings of the nodes being compared are ignored.

Comparison Window

Best Fit

Fits a selected column to the size that fits all text in the values.

Best Fit (All Columns)

Fits all columns to the size that fits all text in the values.

Freeze Column

Fixes a column at the left so that it does not scroll with the other columns.

Select Reference Column

Selects the column used to generate difference values.

SECTION 7

Security and Account Management

Overview

This section describes security features and procedures in the SEL Real-Time Automation Controller (RTAC) family of products, and in ACSELERATOR RTAC SEL-5033 Software. The RTAC provides physical and cybersecurity measures to block unauthorized access as well as logging and alarming to report access activity.

Ethernet Security

You can configure RTAC Ethernet security settings in the RTAC web interface. To access the RTAC web interface, connect the RTAC to your PC via Ethernet or with a USB B cable and enter **https://** and the IP address or Hostname of the RTAC. If you make the connection with a USB B cable, enter 172.29.131.1 for the IP address. If you use the hostname, as configured in the RTAC web interface, the RTAC will need to be on a network that can resolve the hostname to the IP address. When you connect to the RTAC web interface for the first time, there are no defined user accounts. The RTAC prompts you for a username and also for two password entries. The username defines the first administrator account on the box. The password entry is the password for that username and must be a complex password. A complex password must be from 8 to 72 characters long and contain at least one of each of the following character types: number, lowercase letter, uppercase letter, and special character. The RTAC prompts you to enter the password twice to ensure both entries match exactly. Once you enter the new username and password, the RTAC prompts you to use the new username and password to log in. If you forget the username and password you have created, you cannot log in to the RTAC and must apply internal jumpers to override login authentication. See *Section 3: Factory Reset* in the *SEL-3555*, *SEL-3505*, *SEL-3530*, *SEL-3530-4*, or *SEL-2240 Instruction Manual* for jumper positions.

Figure 7.1 Initial RTAC Login

Once you establish a web session, the RTAC monitors usage and closes the session automatically if it is inactive longer than the timeout you have configured. From the web interface, you can set security settings related to individual Ethernet ports, global Ethernet connectivity, and passwords.

Secure Ethernet Access

The RTAC communicates by using varied standard secure Ethernet methods depending on the connection type. Web access is encrypted through the use of HTTPS over a Transport Layer Security (TLS) socket. HTTPS is a secure HTTP connection that prevents both the chance for man-in-the-middle attacks and clear text transmission of the web session. Remote engineering access over Ethernet is configurable to use Telnet or Secure Shell (SSH) between the RTAC and the remote PC. When using SSH, the RTAC will disconnect the attempted connection after three failed connection attempts. Other connections, such as project downloading and gateway access, are encrypted over ODBC connections via TLS sockets.

Typically, SSH connections require user login authentication. Click on **SSH Keys** under the **Security** section of the RTAC web interface to upload, delete, or edit authorized SSH keys to authenticate incoming SSH connections to allow access without a password. To add an authorized SSH key, click on **Add New SSH Authorized Keys**, then fill in the following information:

- **Name**: Enter a name for the key.
- **SSH Key Mode**: Define SSH type as SSH-RSA or SSH-DSS.
- **SSH Authorized Key**: Paste the key contents into this box.

Press **Submit** to complete the process. You can back up the key contents with other RTAC settings when you read the RTAC project by using ACSELERATOR RTAC software.

Denial-of-Service Monitoring

The RTAC monitors Ethernet traffic for bursts of Ethernet traffic that may be denial of service (DOS) attacks. If the amount of Ethernet traffic addressed to the RTAC causes processing exceeding 80 percent of the configured watchdog time-out, the RTAC will log the message `Impending watchdog failure; adjusting system priorities`. At that time, the RTAC adjusts system priorities, including Ethernet traffic priorities, to prevent a watchdog time-out condition. After five minutes of lowered priorities, if the traffic has diminished, the priorities are reset to normal and the RTAC logs the following message System priorities restored to normal. If the traffic continues at the same or greater volume and rate, the watchdog timer will expire and log the message `System Watchdog has expired`.

Tunneled Serial Encryption

To achieve data encryption in tunneled serial connections, set the Serial Tunneling Mode to SSL/TLS or to SSH. A tunneled protocol or engineering access channel configured as SSH authenticates access using either a username and password combination, or through SSH key exchange. If the SSH Password field is left empty in the Ethernet Tunneled Serial configuration, authentication defaults to SSH key exchange. Manage SSH keys as described in *Secure Ethernet Access on page 474*.

The same mechanisms that are used to establish an HTTPS connection to the RTAC web server are also used to encrypt a tunneled communications session using TLS/SSL. A public key infrastructure is necessary to exchange valid certificates and verify received certificates. This infrastructure includes the active X.509 certificate used by the RTAC when a client initiates a TLS/SSL connection and the certificate authorities (CA) certificates used to verify X.509 certificates received by the RTAC.

Typically, clients validate the server certificate, however, all servers configured in the RTAC that use a TLS/SSL Ethernet tunnel have the option to validate the client X.509 certificate as well. If **Verify SSL Certificate** in an Ethernet Tunneled Serial server is set to **True**, the server validates the client X.509 certificate to verify that the signing chain is recognized. All CAs in the chain of trust must be installed in the CA certificate page on the RTAC web server to successfully authenticate the client X.509 certificate. See *X.509 Certificates on page 482* and *CA Certificates on page 484* for examples of adding X.509 and CA certificates.

Individual Port Settings

Each Ethernet port has settings to customize secure access. The USB-B port, which acts as an Ethernet interface, has similar but non-configurable settings. See *Figure 7.4*.

Table 7.1 IP Security Settings

Ethernet Setting	Description
Enable NIC	Enables/disables the network interface card (NIC).
Enable PING	Enables/disables the ability to ping the NIC.
Enable EtherCAT	Dedicates this port to EtherCAT connections. You must have at least one port dedicated to EtherCAT on an RTAC to use EtherCAT in a project. Axions have dedicated EtherCAT ports and therefore do not need to use this setting.
Enable ODBC Access	Enables open database connectivity (ODBC) connections such as Microsoft Excel.
Enable Web Access	Enables/disables ability to access the web interface on that NIC.

Ethernet Setting	Description
Interface Bonding	Provides configuration of bonding of that NIC with another (not available on USB-B).
Interface Bridging	Provides configuration of bridging of that NIC with other interfaces (not available on USB-B).
Interface PRP Pairing	Provides configuration of PRP of that NIC with other interfaces (not available on USB-B).
Enable DHCP	Allows a DHCP server to assign a dynamic IP address to that NIC (not available on USB-B).
IP Address	If DHCP is not used, provides configuration of IPv4 format IP address, including subnet mask. Subnet mask is designated by Classless Inter-Domain Routing (CIDR) notation. The number of bits of the IP address (starting from the left) used to forma subnet mask. The RTACs network interface will not route any network packets with IP addresses that do not conform to the configured mask.
Default Gateway	Defines the default gateway for this interface. The default gateway is the IP address of a router that provides a path to a network that is not part of the subnet of this interface. You must configure a default gateway if incoming Ethernet traffic to this interface is not on this interface's subnet.
Primary Gateway	Indicates this default gateway is the router the RTAC will use if initiating traffic to an interface on a different subnet domain.

Bridging and Bonding

You can configure RTAC Ethernet ports as standalone network interfaces with individual IP addresses, or configure them to work together using bridging or bonding settings.

⚠ **IMPORTANT**

Aggregate mode on the RTAC only works when used with other aggregate bonding capable devices on the network.

⚠ **IMPORTANT**

Bonding methods do not provide simple switch or routing functionality.

➤ Bridging enables two network interfaces to work as a common unmanaged switch. Bridging ports establish switch-like functionality that you can use to configure multiple devices into a ring network without requiring an external switch. When you add an Ethernet interface to the bridge, it shares the same IP address as other ports on the bridge but maintains its own MAC address. If the bridged Ethernet interface receives a packet intended for itself, it processes that packet normally. If the bridged Ethernet interface receives a packet intended for another address, it forwards that packet to its destination from the port best suited to send it. The bridge interface runs Spanning Tree Protocol (STP) to determine which port forwards packets. On interfaces with bridging enabled, you can optionally disable STP. When STP is not active on bridged interfaces, the RTAC will not participate in network topology changes. The bridge STP configuration parameters are the following:

- ➢ Bridge Priority = 32768
- ➢ Hello Time = 2 s
- ➢ Max Age = 20 s
- ➢ Forward Delay = 15 s

- Bonding in backup mode enables only one of the Ethernet interfaces at one time, and they both share the same IP address. The RTAC checks the link status at the link frequency (in milliseconds) to determine if that interface is still working. If the link is failed on that interface longer than the down delay (the count of the number of link inspections), then that interface is considered failed. The RTAC then activates the second port.
- Bonding in aggregate mode enables both Ethernet interfaces to share the same IP address. Used with the correct external networking equipment, this provides the ability to increase bandwidth while offering a fully redundant network communications path. To use aggregate mode, connect an Ethernet cable to both of the bonded ports, then to the aggregate bonding compliant device, such as a managed switch. If one cable is broken, the RTAC routes all traffic onto the second cable. If both cables are functioning, the RTAC distributes traffic between the two cables, thus increasing available bandwidth.

Parallel Redundancy Protocol

The RTAC platform implements Parallel Redundancy Protocol (PRP) based upon the IEC 62439-3:2012 standard. PRP provides high availability for Ethernet communications by sending the same Ethernet packet over two separate networks with a sequence number and an A or B tag attached to it. The device receiving a PRP packet will accept the first packet it receives from the A or B pair and discard the other packet.

Ethernet interface control from the IEC 61131 logic engine is not supported for interfaces that have PRP enabled. The Ethernet configuration status will not read accurately and changes to the Ethernet interface will not be applied.

To configure PRP, navigate to the Ethernet interface Port Configuration page. To configure an interface to belong in a PRP pair, that interface cannot have DHCP enabled, be bridged, bonded, or have EtherCAT enabled. Starting in firmware version R147, the RTAC supports as many as five pairs of PRP-enabled interfaces. Prior to version R147, only one pair of PRP-enabled interfaces is supported.

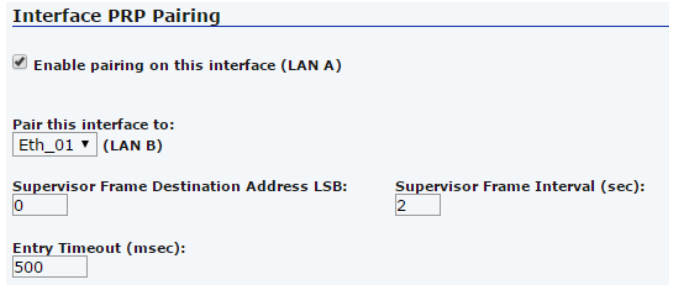


Figure 7.2 Interface PRP Pairing Page

Supervisor Frame Destination Address LSB: This setting allows the user to define the least significant byte in the supervisor frame address that is periodically sent out onto the network. SEL recommends that setting be left at default unless there is a conflict for the supervisor frame multi-cast address. That address is 01-15-4E-00-01-*XX* per the IEC 61439-3 standard, where *XX* is the configurable setting. The range for this setting is 0–255 in decimal and will appear in the MAC address as hexadecimal when transmitted.

Supervisor Frame Interval: The period in which the RTAC will send the supervisor frame out on the PRP network. The purpose of the supervisor frame is for a separate IED to monitor the status of PRP connected devices and provide statistics about the PRP networks.

Entry Timeout: The time that the RTAC will monitor for duplicate sequence numbers received from a single device. For high-bandwidth consumption PRP networks, SEL recommends that this number be set toward the bottom end of the allowed range. For low-bandwidth PRP networks, you can increase this value. This setting might need to be adjusted if the bandwidth of the PRP network is in excess of 70 Mbps.

The bottom of the diagnostics page on the web interface displays statistics about active PRP connections.

```
PRP Diagnostics:

#     Source MAC          Next A Seq  Next B Seq  LAN ID Error
1     00:30:A7:02:80:CD        54786       54786  TRUE

ARP Table:
? (10.203.80.1) at 70:ca:9b:98:4c:cc [ether]  on prp0
? (10.203.80.100) at 00:30:a7:05:9e:44 [ether]  on prp0
? (10.203.85.252) at 00:30:a7:02:80:cd [ether]  on prp0
```

Figure 7.3 Active PRP Connections

Source MAC: Since PRP works at Layer 2, below the IP address layer, all PRP connections will be identified by the MAC address of the IED that the RTAC is communicating with. This MAC address can be correlated with an IP address by looking at the address resolution table (ARP table), displayed below the PRP statistics.

Sequence Numbers: The next predicted sequence number that the RTAC expects to see from a PRP conversation. When this number increments, it indicates the PRP connection is communicating. If this number does not change, then the implication is there is no PRP traffic for that conversation.

LAN ID Error: As seen in the configuration for PRP, each interface is assigned to the A or B network. If the A interface receives a B packet or vice versa then this value will show true. During the configuration of PRP, each interface is labeled as A or B.

NOTE

Do not set unused Ethernet ports to an active subnet.

Table 7.2 CIDR Notation

CIDR Value	Subnet Mask
/32	255.255.255.255
/31	255.255.255.254
/30	255.255.255.252
/29	255.255.255.248
/28	255.255.255.240
/27	255.255.255.224
/26	255.255.255.192
/25	255.255.255.128
/24	255.255.255.000
/23	255.255.254.000
/22	255.255.252.000
/21	255.255.248.000
/20	255.255.240.000
/19	255.255.224.000
/18	255.255.192.000
/17	255.255.128.000
/16	255.255.000.000
/15	255.254.000.000
/14	255.252.000.000
/13	255.248.000.000
/12	255.240.000.000
/11	255.224.000.000
/10	255.192.000.000
/9	255.128.000.000
/8	255.000.000.000
/7	254.000.000.000
/6	252.000.000.000
/5	248.000.000.000
/4	240.000.000.000
/3	224.000.000.000
/2	192.000.000.000
/1	128.000.000.000
/0	000.000.000.000

Subnet Usage

Unless connected physically to the same network, each Ethernet interface should use a different subnet. For example, if two RTAC network interface cards (NICs) have the same subnet, but one is not connected to the local area network (LAN), the RTAC may send some messages erroneously to the disconnected interface. The resulting misdirected messages are lost, because the RTAC has no way of verifying whether the disconnected interface is a valid path on which to send the messages. Use port bridging if you want two network interfaces to work as a common, unmanaged switch.

Example 7.1 Incorrect Subnet Setup

The following example illustrates an incorrect setup that will result in communications problems with the RTAC. Three RTAC NICs have the same subnet. Two NICs physically connect to the same LAN, but one does not.

✓	Eth1 IP = 10.201.10.1/24	Connected to local switch communicating with local IEDs.
✓	Eth2 IP = 10.201.10.4/24	Connected to local switch communicating with remote control center LAN.
×	EthF IP = 10.201.10.5/25	Disconnected (will create communications errors on other two ports.)

The RTAC will direct some network traffic to the disconnected interface because it is on the same subnet as the other two NICs. To resolve the problem, either physically connect EthF to the local switch or reconfigure it to a different subnet. For example, set the IP address to 10.201.*x*.5, where *x* = a number except 10 in the range of 1–255.

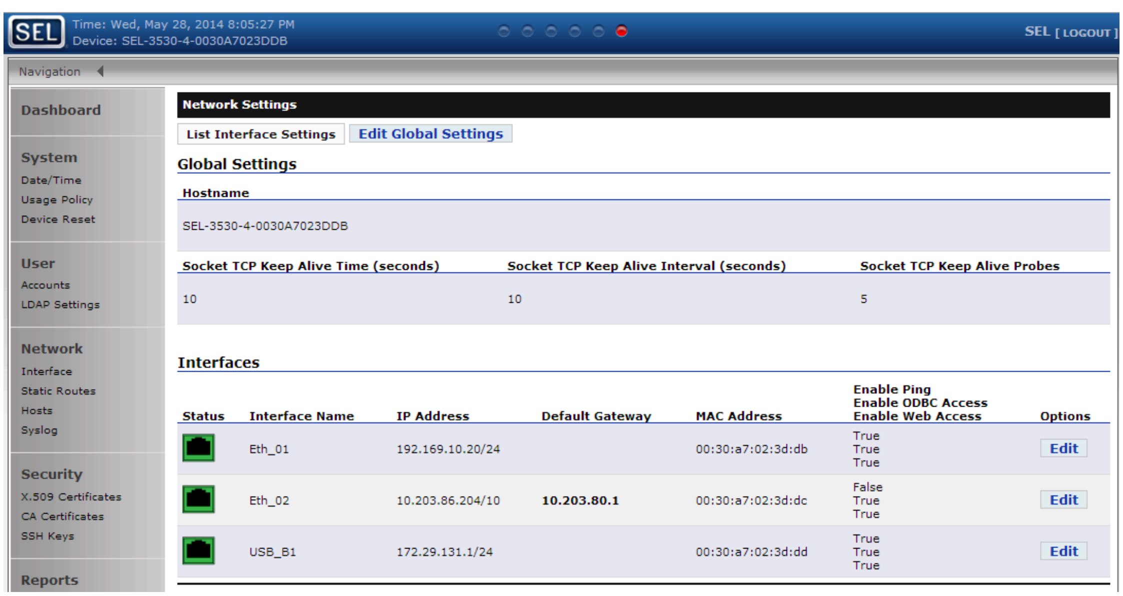


Figure 7.4 Ethernet Settings

Global Ethernet Settings

The RTAC also has Global settings that provide Ethernet security and settings. See *Figure 7.4* and *Table 7.3*.

Table 7.3 Global IP Settings

Global Ethernet Settings	Description
Hostname	The network name of the RTAC.
Static Routes	Adds routes to specific network destinations or hosts by using specific RTAC network interfaces.
Socket TCP Keep Alive Time (Seconds)	The time the RTAC will wait while not receiving any TCP traffic on a socket connection before sending TCP keep alive probes (range: 1–7200).
Socket TCP Keep Alive Probes	The number of probes that will be sent before determining that the TCP socket should be closed if no response is received (range: 1–2000).
Socket TCP Keep Alive Interval (Seconds)	The time that the RTAC will wait between sending TCP keep alive probes (range: 1–7200).
Hosts	Adds host names with their respective IP addresses to the RTAC hosts table. The RTAC needs this information for LDAP configuration.
X.509 Certificate	Generates a new X.509 certificate for web access security.

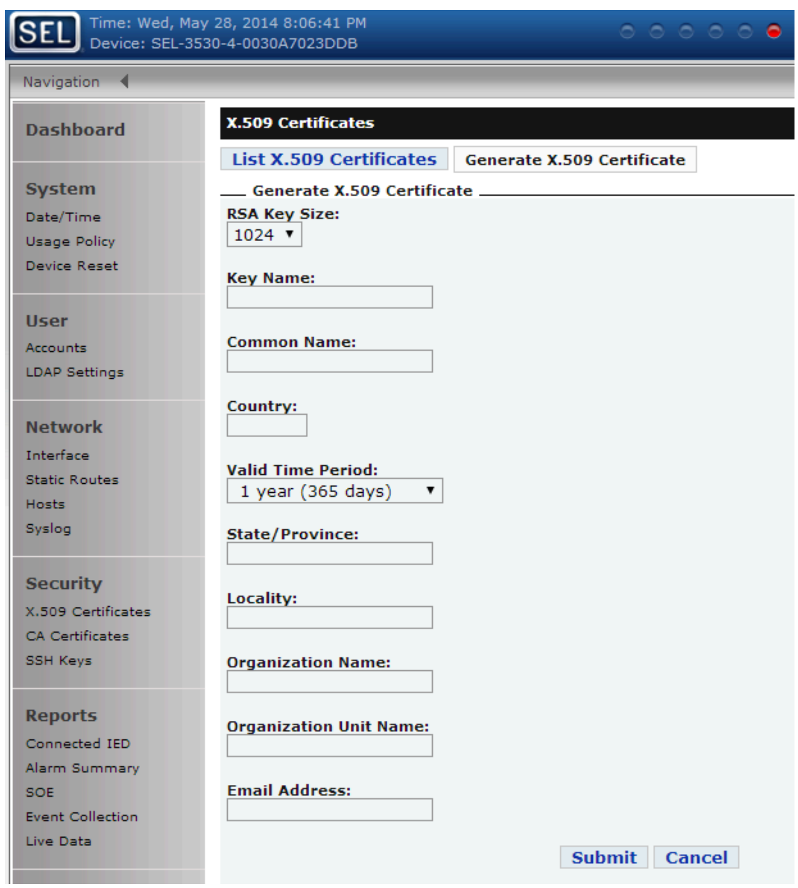


Figure 7.5 Routes and Certificates

Static Routes

Static routes allow a user to specify a path for Ethernet traffic other than the active default gateway IP address. The most common use case of static routes for substation applications is for a network where multiple routers exist and the RTAC needs to communicate with IEDs behind each router.

Table 7.4 Static Route Settings

Static Route Settings	Description
Remote Gateway	The gateway address of the router with which to communicate.
Remote Network	The desired network/IP address with which to communicate. This can be a single IP address, such as 172.29.45.13, or a range of IP addresses specified with a CIDR notation such as 172.29.45.0/24. If specifying a range of IP addresses, the octet, which is part of a range that changes, needs to be 0.
Network Interface	Configure the physical network interface that the RTAC will use to communicate to the router gateway.

X.509 Certificates

The RTAC contains a single unsigned X.509 certificate used for Hypertext Transfer Protocol Secure (HTTPS) and open database connectivity (ODBC) through Transport Layer Security (TLS). The RTAC supports base64 (.cer files) X.509 and CA certificates and does not support any of the binary formats. CA certificates are used by the RTAC for authentication of LDAP and TLS/SSL Ethernet Tunneled Serial connections and are explained more in *CA Certificates on page 484*. You can create, import, and export X.509 certificates through the following steps.

Step 1. Log in to the RTAC web interface.

Step 2. Select **Security**.

Step 3. Choose **X.509 Certificates**.

Step 4. Click **Generate**.

Step 5. Fill out the required information.

Note that the Country name must be two characters, and Common Name must match exactly the IP address or hostname that you are using in the address bar to access this RTAC. In addition, most web browsers also require the IP address or hostname to be populated in the Subject Alternative Name (SAN) fields. Do not leave this field unspecified.

Step 6. Click **Submit**.

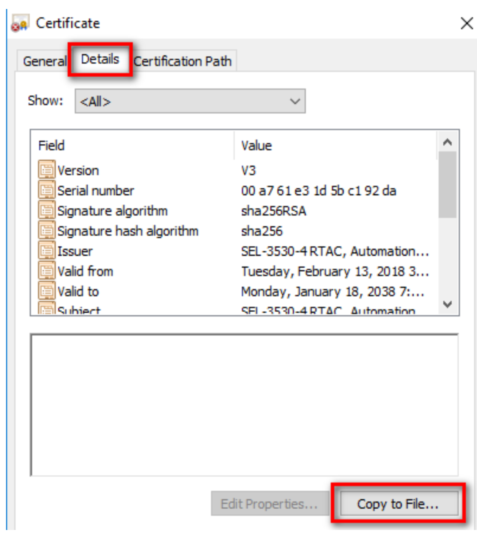
After you have created a certificate, you can either activate it and mark it as trusted for your PC or give the certificate to your information services group to sign. Perform the following steps to install certificates into the Windows certificate store. After the certificate is installed in the trusted root store, all web browsers from that computer should show a trusted connection.

Step 1. On the X.509 Certificates page, click **Activate** to begin using the new certificate. The RTAC reboots to begin using the new certificate. After the RTAC reboots, the web browser shows that there is still a certificate error.

Step 2. Select **Continue to this website (not recommended)** from the certificate error prompt.

Step 3. On the login page, click **Certificate Error** at the top of the page. Many web browsers provide an option to view the certificate. If the web browser does not offer an option to view or inspect the certificate, use another browser, perform a web search on how to view certificates from your specific browser, or download the certificate from the RTAC X.509 Certificate page after logging into the RTAC webpage.

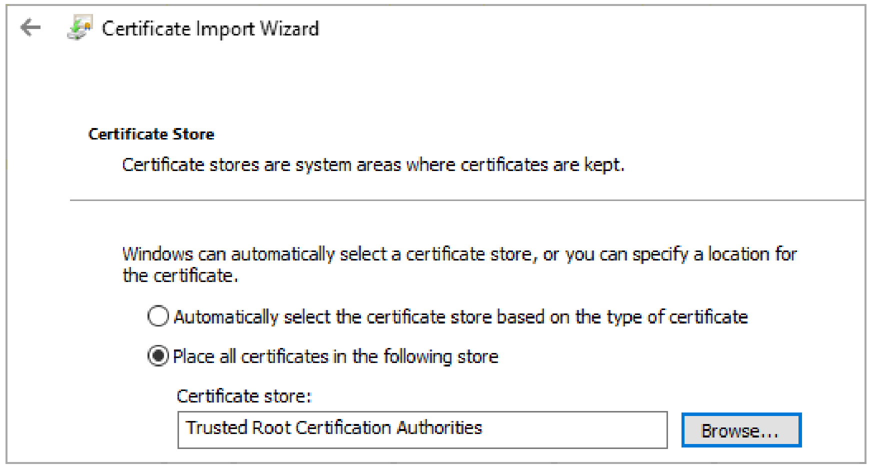
Step 4. Select **View Certificates** from the pop-up window. The certificate should appear in a new window. Click the **Details** tab, which should provide an option to copy the certificate to the local file system of the computer. Copy the file encoded as either DER or Base-64.



Step 5. *(Optional)* Alternatively, the certificate can be downloaded by navigating to the X.509 certificate after logging into the RTAC, copying the entire certificate into a text editor, and saving the file with the extension **.cer**. This will accomplish the same objective as the process completed in the previous step. The certificate will be saved on the computer as Base-64.

Step 6. When the certificate is on the local file system, right-click and select **Install Certificates**.

Step 7. Click **Place all certificates in the following store** and browse to **Trusted Root Certification Authorities**.



Step 8. Finish the Wizard. When the Wizard tells you the import was successful, close and restart the web browser to verify a trusted connection to the RTAC.

If your company has a signing authority CA, perform the following steps to place your company's certificate signature on the new X.509 certificate.

Step 1. After creating the certificate on the RTAC web interface, click the **C.S.R.** button.

Step 2. **Copy** the entire text into a text-editing software such as Notepad. Do not attempt to edit the text and do not place it in an advanced word editor (such as Microsoft Word) that might change the contents.

Step 3. Click **Done** and give the text file to your signing authority.

Step 4. When the signing is complete, click on **Import** for that X.509 entry in the RTAC web interface.

Step 5. **Paste** the entire text contents into the space provided. Ensure that you paste the entire contents and have not inadvertently altered the contents in any way.

Step 6. Click **Submit**.

Step 7. Click **Activate** to begin using the new certificate.

The default self-signed certificate is not editable or viewable. If you reset the RTAC to default settings, all certificates are erased except the default unsigned certificate.

CA Certificates

During the initial stages of establishing a TLS/SSL encrypted communications session, client and server devices exchange public keys. To verify the server's ownership of the public key, CAs digitally sign the certificates used by the server. CA certificates are issued by the CA and contain the public key mathematically related to the private key used to sign an X.509 certificate. The entity verifying the X.509 cross-references the installed CA certificates to verify that the X.509 is signed by a trusted institution(s). An X.509 certificate may be signed by multiple CAs to create a chain of trust. To successfully validate an X.509, all CA certificates in the chain of trust must be installed in the CA Certificate page on the web interface.

Step 1. Click on **CA Certificates** in the RTAC web interface, select **Add New CA Certificate**, then enter the name of the certificate in the **Name** field.

Step 2. Next, open the certificate for the LDAP or other server in Notepad or another word processor that will not reformat the text. It is important that the word processor not reformat the certificate.

Step 3. Copy and paste the contents from the RTAC web interface in the **Certificate** field.

Step 4. Click **Submit**. The RTAC will install the certificate and begin using the certificate for LDAP and for X.509 authentication for other connections.

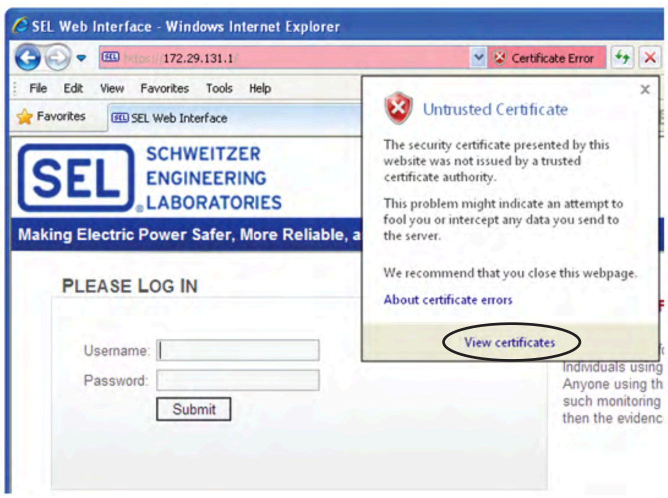


Figure 7.6 Accept New Certificate

URL Whitelist

Firmware versions R143 and later support whitelisting of webserver domains to which a user can navigate by using the RTAC web interface. The URL Whitelist is under the **Security** section of the RTAC web interface. URLs can be added or deleted from this list, as shown in *Figure 7.7*.

Figure 7.7 URL Whitelist

URLs are not case sensitive. The URL must conform to the following rules and formats:

URL Format: [*scheme://*][.]*host*[:*port*][*/path*][@*query*]

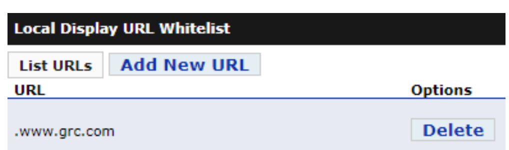
- ➤ *Scheme* can be http, https, ftp, etc. This field is optional and must be followed by *://*. Note that while a file server can be reached via the web interface, no files can be downloaded *onto* the RTAC via this mechanism.
- ➤ An optional dot (.) can prefix the host field to disable subdomain matching.
- ➤ The *host* field is required and is a valid hostname or IP address. It can also end with an asterisk (*) to indicate a prefix match.
- ➤ An optional *port* can come after the host. It must be a valid port value from 1 to 65535.
- ➤ An optional *path* can come after the port. Any string can be used for this value.
- ➤ An optional *query*, which is a set of key-value and key-only tokens delimited by an ampersand (&), can be included at the end of the URL. The key-value tokens are separated by an equals symbol (=). A query token can optionally end with an asterisk (*) to indicate a prefix match. Token order is ignored during matching.

If you prefer that the RTAC not impose any whitelist restrictions, then you can add a single rule such as "http://*" and/or "https://*" to allow access to any URL.

Example 7.2 IP Address

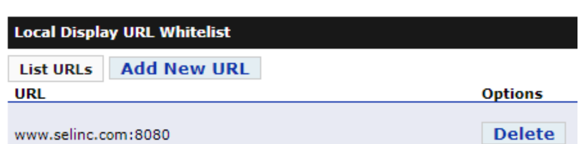
Entering the IP address, as shown above, allows you to navigate to the IP address registered as 192.168.1.12. Because only the hostname is included, you can add any prefix (e.g., *http://*192.168.1.12 or *https://*192.168.1.12).

Example 7.3 No Access to Subdomains



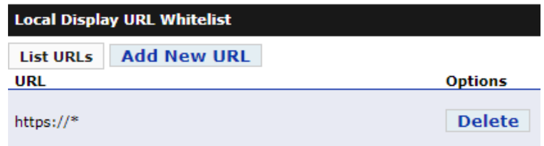
This example shows how to allow access to a domain without access to any of its subdomains. Access to any associated subdomains (for example, www.steve.grc.com) is not allowed.

Example 7.4 Using a Nonstandard Port to Access a Web Server



This example shows how to append a port to the destination URL to use a different port than http (80) or https (443).

Example 7.5 Allowing Access to Any HTTPS Connection



This example shows how to allow for access to any destination that begins with *https://*. This provides similar functionality to the filter of **.443*, which allows access to any domain or IP connection with Port 443. Note that it is not possible to whitelist an IP range; for example, a filter of *192.168.1.** will *not* allow access to any IP in the Class D IP range.

ACSELERATOR RTAC Software Passwords and User Accounts

The RTAC system has two account schemes as well as accounts and passwords associated with communications protocol sessions. The first designates who can use the ACSELERATOR RTAC configuration software and what rights these users have. The second defines users and their privileges in the RTAC. This second scheme applies to the RTAC web interface, open database connections (ODBC), and transparent connections.

Communications protocol session accounts and passwords provide authentication of communications sessions. The ACSELERATOR RTAC encrypts and stores these passwords.

ACSELERATOR RTAC Configuration Software Passwords

Once you are logged in as administrator, set up user accounts in ACSELERATOR RTAC:

Step 1. Click the **SEL action button > Options > User Accounts**.

Step 2. Click on + to add users.

Step 3. Create a name and strong password.

A strong password must meet the following criteria:

- minimum eight characters
- at least one digit
- at least one special character (!, $, etc.)
- at least one capitalized letter
- at least one lowercase letter

Step 4. Assign a user role.

- Administrator
  - Create/delete projects
  - Unlock improperly closed projects

- ➢ Create/delete user accounts
- ➢ Send/read projects to/from RTACs
- ➢ Set project passwords and export encryption pass phrases

➤ Engineer

- ➢ Create/delete projects
- ➢ Send/read projects to/from RTACs
- ➢ Set project passwords and export encryption pass phrases

Step 5. Click **Close**.

Step 6. Click **Ok**.

ACSELERATOR RTAC automatically saves changes, after which these changes are ready for use. See *Figure 7.8*.

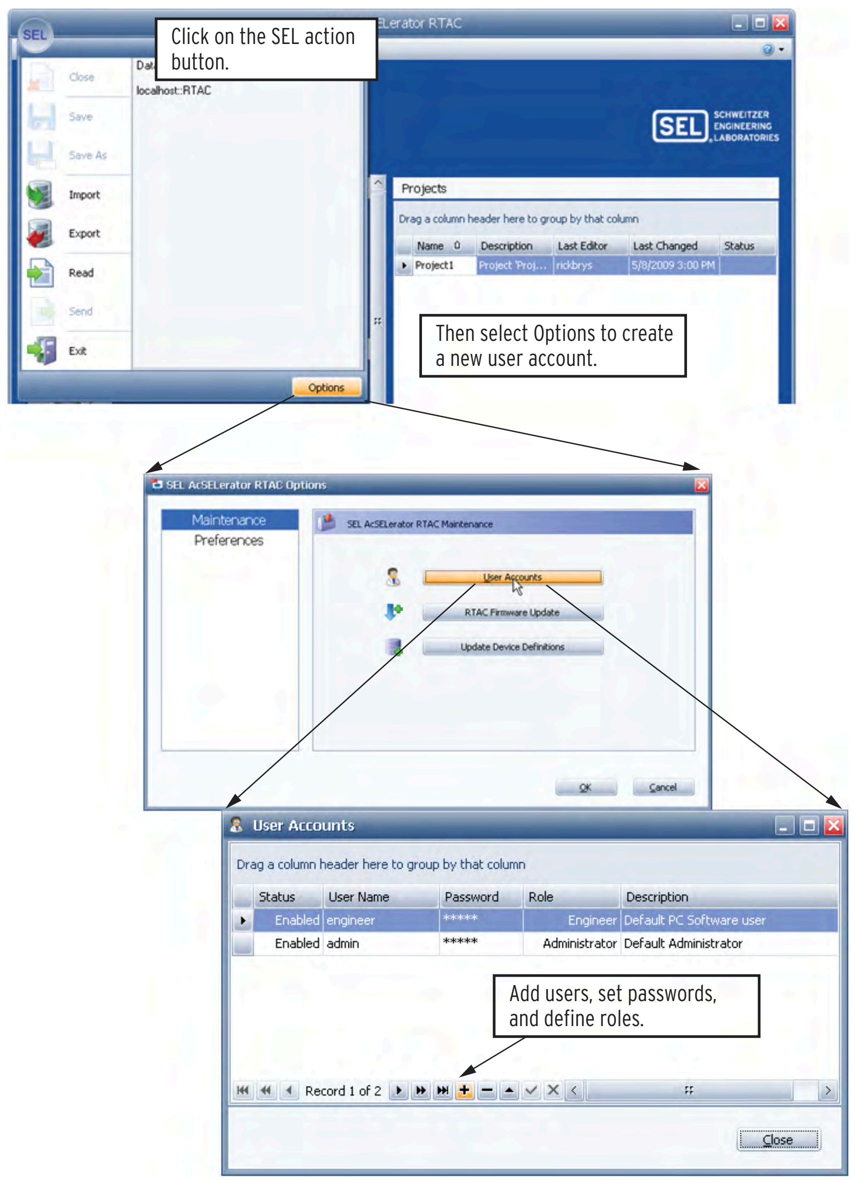
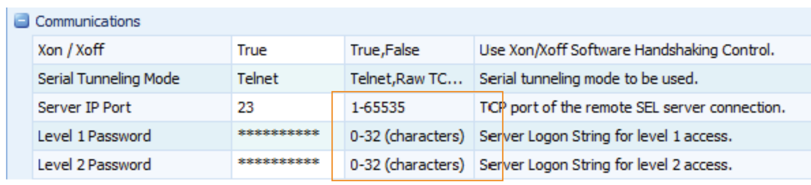


Figure 7.8 ACSELERATOR RTAC User Accounts

Protocol Passwords

Communications protocol passwords provide authentication of authorized communications sessions and are necessary for elevating access levels for restricted services. ACSELERATOR RTAC project versions prior to R136 store protocol passwords in plain text. In projects created for ACSELERATOR RTAC project versions R136 and later, all communications protocol passwords are encrypted, ensuring that sensitive passwords are not disclosed to unauthorized persons. The ACSELERATOR RTAC database stores the password cypher text is with asterisk characters replacing the encrypted cypher text in the ACSELERATOR RTAC password fields. Converting project versions to R136 or later encrypts the passwords in the converted project.



Communications			
Xon / Xoff	True	True,False	Use Xon/Xoff Software Handshaking Control.
Serial Tunneling Mode	Telnet	Telnet,Raw TC...	Serial tunneling mode to be used.
Server IP Port	23	1-65535	TCP port of the remote SEL server connection.
Level 1 Password	**********	0-32 (characters)	Server Logon String for level 1 access.
Level 2 Password	**********	0-32 (characters)	Server Logon String for level 2 access.

Figure 7.9 Hidden Passwords in an SEL Client

Password Report

Starting in firmware revision R136, passwords for IED communications are encrypted and obfuscated in the ACSELERATOR RTAC software. Some users want to see the passwords included in the project configuration, often to verify that the RTAC configuration contains the correct passwords. To accommodate this, starting in firmware revision R141, a report is available on the RTAC web interface with an Administrator account or a custom account with the password report permission included.

The report generates a CSV file that contains the IED name in the ACSELERATOR RTAC configuration, setting name, and the corresponding password.

This report does not contain any passwords for user accounts on the RTAC.

Project Passwords

Add explicit project-level security by setting project passwords from the ACSELERATOR RTAC homepage or from within a project (available in R136 firmware or later). If a project is password-protected, the user must provide a valid password before the project will open with ACSELERATOR RTAC. Project passwords must follow the same password rules defined for an RTAC user account (see *ACSELERATOR RTAC Configuration Software Passwords on page 487*). Once a project password has been created, it can be changed or disabled by an administrative user with the current project password.

Complete the following steps to set a project password:

Step 1. Open ACSELERATOR RTAC software.

Step 2. Log in to the RTAC database.

Step 3. Right-click an unlocked project in the **Projects** window and click **Set Password**.

Step 4. From the **Set Password** pop-up window, enter a new password in both the top and bottom fields. The horizontal bar beneath the confirmation field provides a gauge for password strength. The **Set** button becomes selectable once the software considers the password strong and both lines match.

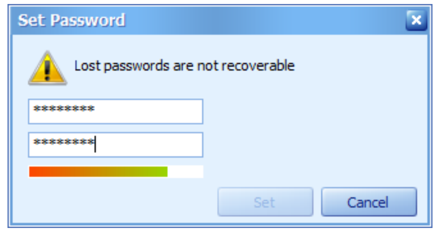
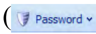


Figure 7.10 ACSELERATOR RTAC Project Password Management

Step 5. Click the **Set** button.

You also can set project passwords from an open project by clicking **Password** () from the toolbar of the **Home** tab and selecting **Set Password** from the drop-down menu. Alternatively, you can set project passwords with the keyboard combination **<Ctrl+L>**.

A password-protected project will display **Password On** in the ACSELERATOR RTAC projects view **Status** column. This field will otherwise be blank. From an open project, the password status is shown in the status bar in the bottom right corner of the window.

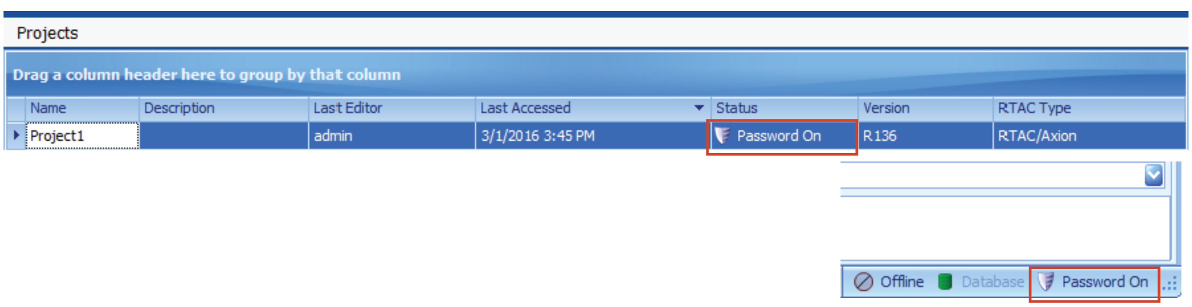


Figure 7.11 Project Password Status

Disable passwords from within the **Projects** view by right-clicking on the desired project and selecting **Disable Password**. You must enter the current password before you can remove password protection from the project. You can also accomplish this from within a project by clicking the **Password** button () and selecting **Disable Password** or typing the keyboard combination **<Ctrl+D>**.

Set or disable a single password for multiple projects by selecting all of the desired projects in the ACSELERATOR RTAC **Projects** view and performing the previous *Step 1* through *Step 4*.

Project Export Encryption

ACSELERATOR RTAC encrypts all exported projects through use of a default hidden pass phrase that generates a pseudo-random number based on the hash-based message authentication code and the secure hash algorithm. The software uses this number to augment the symmetric encryption of the project data with additional pseudo-randomness. You can also enter a custom pass phrase to serve as this number. Once you have enabled this feature, the RTAC prompts you to create a new pass phrase when exporting a project. Similarly, the RTAC prompts you to enter this pass phrase when you import the project into ACSELERATOR RTAC or into the RTAC via the web interface. Once you import the project into the ACSELERATOR RTAC software, the pass phrase is not necessary to open the project.

Once you are logged in to the ACSELERATOR RTAC, enable the export encryption passphrase prompt:

Step 1. Click the **SEL** action button, then **Options > Preferences**.

Step 2. Select the **Prompt for passphrase to encrypt project export** check box.

Step 3. Click **OK**.

Now encrypt an exported project:

Step 1. From the ACSELERATOR RTAC start page, click the desired project to highlight it.

Step 2. Click the **SEL Application** button or right-click the project name and click the **Export** icon ().

Step 3. From the **Set Encryption Passphrase** pop-up window, enter a new password in both the top and bottom fields. The horizontal bar beneath the confirmation field provides a gauge for password strength. When the bar is filled, the password is considered strong, and the **Set** button becomes selectable.

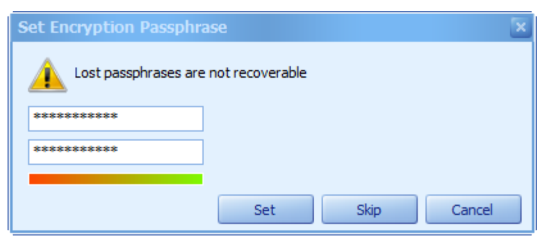


Figure 7.12 ACSELERATOR RTAC Project Export Encryption Management

Step 4. Click **Set**. You also can click **Skip** to use the default encryption seed for this exported project.

RTAC Web Interface User Accounts

User accounts for the RTAC are configured via the web interface. For firmware versions R100–R140, user account information is located in the User Accounts section. For firmware versions R141 and higher, refer to *Customizable User Accounts in R141 and Higher on page 493*. Starting in version R147, the IP address from which a user logs into the RTAC will be included in the login message in the Sequence of Events log.

User Accounts in R100–R140

Set up user accounts in the RTAC web interface. You must configure at least one Administrator account on the RTAC and you should not configure that account to expire. Complete the following steps to configure a new user on the RTAC:

Step 1. Log in to the web interface in an **HTTPS** web session.

Step 2. Click on **User** and then **Accounts**.

Step 3. Click **Add New User**, as shown in *Figure 7.13*.

Step 4. Enter new username.

Step 5. Assign user role.

Table 7.5 Account Access

	Administrator	User Account Manager	Engineer	Monitor	HMI Operator	File Transfer
User Account Settings	Read/Write	Read/Write	Read	No Access	No Access	No Access
Ethernet/System Settings	Read/Write	Read/Write	Read/Write	Read	Read	No Access
Project Settings	Read/Write	No Access	Read/Write	No Access	No Access	No Access
Reports and Diagnostics	Read/Write	Read	Read/Write	Read	Read/Write	No Access
Device Management	Read/Write	Read	Read	Read	Read	No Access
Engineering Access	Read/Write	No Access	Read/Write	No Access	No Access	No Access
Operate HMI Controls	Yes	No	Yes	No	Yes	No
S/FTP Access	Read/Write	Read	Read/Write	Read	Read	Read/Write
Web Communication Captures	Read/Write	No Access	Read/Write	No Access	No Access	No Access
Web Live Data	Read/Write	No Access	Read/Write	Read	Read	No Access

Step 6. If you assign an expiration date, the account will automatically become inactive on the date you assign.

NOTE
You cannot create a user with the name diag. An internal RTAC application already uses this username.

NOTE
Users who log in via LDAP will appear in the accounts list. Deleting these users will not remove their ability to log in to the RTAC as that is managed through the LDAP server.

Step 7. Noncomplex passwords are simple text strings. Complex passwords must contain the following:

- Minimum eight characters
- At least one digit

- At least one special character (!, $, etc.)
- At least one capitalized letter
- At least one lowercase letter

Step 8. Select the **Account Enabled** check box and press **Submit**. The account is now ready for use.

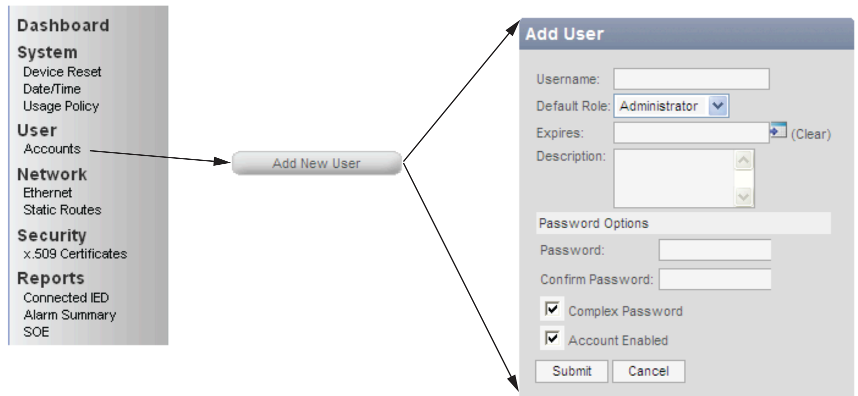


Figure 7.13 RTAC Web Access User Accounts

Customizable User Accounts in R141 and Higher

Starting in firmware revision R141, user accounts are described in terms of permissions. *Table 7.6* shows all permissions in the RTAC and a description for them. Note that the standard user accounts created prior to R141 still exist. Also note that User Manager and the Monitor roles have reduced capabilities starting in R141. Other accounts, such as Administrator, Engineer, HMI Operator, and File Transfer contain the exact same permissions as prior to R141. *Table 7.7* shows the permissions associated with each default user role.

When upgrading to R141 or later from a pre-R141 firmware version, note the following:

> Account names can no longer share the same name as account roles. If an account has the same name as an account role in R141 or later, the account name will have an underscore appended to it. For example, if a user account named Administrator in R140 is used to update the RTAC firmware version to R141 or later, that account will be renamed as Administrator_. A common username is Administrator; however, since this matches one of the account types, any spelling of Administrator is no longer a valid username after R141.

Table 7.6 Available Permissions

RTAC Permission Name	Default Account	Description
Account Write	Administrator, User Manager	Authorized to modify and create User Accounts. LDAP configuration requires the following permissions: Account Write, Network Write, and Security Write. Authorizes ACSELERATOR RTAC to send User Accounts and LDAP Settings when sending a project to the RTAC.
API Login	Administrator, Engineer, User Manager, Monitor, HMI Operator	Authorized to login via the API. Individual API calls may require additional permissions.
Database Login	Administrator, Engineer	Authorized access to Port 5432. This access is needed to download configurations from ACSELERATOR RTAC, collect data with ACSELERATOR TEAM SEL-5045 Software, ODBC queries, and perform firmware upgrades.
Device Password Read	Administrator	Authorized to retrieve passwords from all connections that require a password in the active RTAC configuration.
Device Password Write	Administrator	Authorized to write new passwords for IEDs in the active project via API.
Device Settings	Administrator, Engineer	Authorized to modify Device Information and Date/Time on the web interface. Authorizes ACSELERATOR RTAC to send Website settings when sending a project to the RTAC.
Diagnostic Settings	Administrator, Engineer, File Transfer	Authorized to read diagnostic information.
Engineering Access	Administrator, Engineer	Authorized to access the SEL Server and other access points.
Factory Settings	Administrator	Authorized to restore the device to the factory-default settings.
File Read	Administrator, Engineer, User Manager, Monitor, HMI Operator, File Transfer	Authorized to read files via File Manager and the file system via protocols that support file transfer mechanisms.
File Write	Administrator, Engineer, File Transfer	Authorized to write files via File Manager and the file system via protocols that support file transfer mechanisms. This permission inherits File Read.
Firmware Upgrade	Administrator, Engineer	Authorized to apply Firmware to the device.
Force Data	Administrator, Engineer	Authorized to force data through Live Data and ACSELERATOR RTAC software. This permission inherits Live Data.
HMI Local Write	Administrator, Engineer, HMI Operator	Authorized to issue controls on the HMI via the local display.
HMI Remote Write	Administrator, Engineer, HMI Operator	Authorized to issue controls on the HMI via a remote web browser connection.
IED Connection Information	Administrator, Engineer	Authorized to read the Connected IED page.
Live Data	Administrator, Engineer, User Manager, Monitor, HMI Operator	Authorized to read the Live Data webpage.
Logic Engine API Read	Administrator	Authorized to read tags from the RTAC logic engine via the API.

RTAC Permission Name	Default Account	Description
Logic Engine API Write	Administrator	Authorized to write tags to the RTAC logic engine via the API.
Manage Audits	Administrator	Authorized to create or delete audit reports.
Network Read	Administrator, Engineer, User Manager, Monitor, HMI Operator	Authorized to read network settings.
Network Utilities	Administrator, Engineer	Authorized to run Comm Monitor and network utilities. This permission inherits Network Read.
Network Write	Administrator, Engineer	Authorized to modify network settings. Authorizes ACSELERATOR RTAC to send Ethernet Settings and Hosts when sending a project to the RTAC. This permission inherits Network Utilities.
Project Read	Administrator, Engineer	Authorized to read projects from the web interface and ACSELERATOR RTAC software via PROJECTS directory for file transfer protocols.
Project Write	Administrator, Engineer	Authorized to send projects from the web interface and ACSELERATOR RTAC software via PROJECTS directory for file transfer protocols. This permission inherits Project Read.
Reboot Device	Administrator, Engineer, User Manager, Monitor, HMI Operator	Authorized to reboot the device.
Relay ACCess	Administrator, Engineer	Authorized to enter Access Level 1 on the SEL server.
Relay 2ACcess	Administrator	Authorized to enter Access Level 2 on the SEL server.
Report Delete	Administrator, Engineer, HMI Operator	Authorized to delete SOE events, either individually on the SOE page, or all events via Device Management page. Authorized to delete events from the Event Collection webpage. This permission inherits Report Write.
Report Read	Administrator, Engineer, User Manager, Monitor, HMI Operator, File Transfer	Authorized to read SOE, Alarm Summary, and Event Collection webpages. Authorizes access to the EVENTS directory for protocols that support file transfer mechanisms. Authorizes ACSELERATOR RTAC to read Event Collection and Event Logs when reading a project from the RTAC.
Report Write	Administrator, Engineer	Authorized to acknowledge SOE, Alarms, and Events. Authorizes ACSELERATOR RTAC to send Event Collection and SOE logs when sending a project to the RTAC. This permission inherits Report Read.
Security Read	Administrator, Engineer, User Manager.	Authorized to read CA Certificates, X.509 Certificates, SSH Keys, and the Usage Policy. Authorizes ACSELERATOR RTAC to read CA certificates, SSH Authorized Keys, SSH Host Keys, X.509 Certificates, and Usage Policy when reading a project from the RTAC.
Security Write	Administrator, Engineer, User Manager.	Authorized to edit CA Certificates, X.509 Certificates, SSH Keys, and the Usage Policy. Authorizes ACSELERATOR RTAC to send CA certificates, SSH Authorized Keys, SSH Host Keys, X.509 Certificates, and Usage Policy when sending a project to the RTAC. This permission inherits Security Read.

RTAC Permission Name	Default Account	Description
SFTP Login	Administrator, Engineer, User Manager, Monitor, HMI Operator, File Transfer	Authorized to log in to the file system via SFTP or FTP.
Web Login	Administrator, Engineer, User Manager, Monitor, HMI Operator	Authorized to log in via the Web interface.

Table 7.7 Default Users and Associated Permissions

Default Roles	Permissions
Administrator	Account Write, Database Login, Device Password Read, Device Settings, Diagnostic Settings, Engineering Access, Factory Settings, File Read, File Write, Firmware Upgrade, Force Data, HMI Local Write, HMI Remote Write, IED Connection Information, Live Data, Network Read, Network Utilities, Network Write, Project Read, Project Write, Reboot Device, Report Delete, Report Read, Report Write, Security Read, Security Write, SFTP Login, Web Login
User Account Manager	Account Write, File Read, Live Data, Network Read, Reboot Device, Report Read, Security Read, Security Write, SFTP Login, Web Login
Engineer	Database Login, Device Settings, Diagnostic Settings, Engineering Access, File Read, File Write, Firmware Upgrade, Force Data, HMI Local Write, HMI Remote Write, IED Connection Information, Live Data, Network Read, Network Utilities, Network Write, Project Read, Project Write, Reboot Device, Report Delete, Report Read, Report Write, Security Read, Security Write, SFTP Login, Web Login
Monitor	File Read, Live Data, Network Read, Reboot Device, Report Read, SFTP Login, Web Login
HMI Operator	File Read, HMI Local Write, HMI Remote Write, Live Data, Network Read, Reboot Device, Report Delete, Report Read, SFTP Login, Web Login
File Transfer	Diagnostic Settings, File Read, File Write, Report Read, SFTP Login

To use an account with customizable permissions, first create a new user role. Navigate to the **User Roles** page to create new user roles. This page allows for user role management, including editing, deleting, and adding new roles. This page lists all existing user roles on the RTAC. Default user roles cannot be removed or modified.

Table 7.8 User Roles Columns

Column Name	Description
Role Name	Contains the name for the user role.
Description	Contains notes about the user role. Non-default user accounts may include user-created descriptions.
Members	This column includes information about the account that may have inherited permissions from other user roles. Direct Members: User roles that were directly selected when creating the user role. Eventual Members: User roles that Direct members user roles referenced.
Permissions	Contains a list of permissions for the user roles. If all permissions do not fit on the screen, a tooltip appears when hovering over the permission text to list all permissions.
Options	Contains the options for editing, copying, and deleting a user role. Default user roles cannot be edited or deleted.

When you add a new role, you can include a description. To resize the **Role** description box, click and drag on the bottom right corner. This allows for all text to be easily viewed. When creating a new user role, you can select existing roles on the unit to provide a starting set of permissions for the new role. Many users want custom roles based on default accounts. Selecting an existing role will populate all permissions associated with that role. Additional permissions can be added but removing roles that were inherited from another role is not possible. In this case, deselect the role. All permissions associated with the previously selected role remain selected but can now be removed.

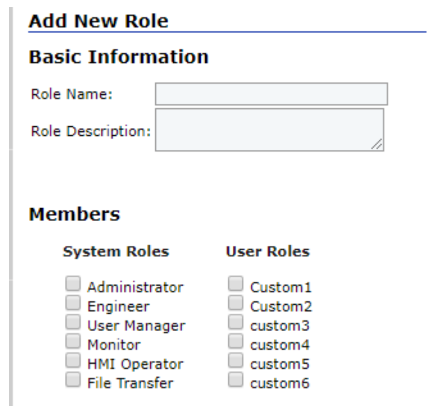


Figure 7.14 Add New Role Page

This page also includes each individual permission, as shown in *Table 7.9*.

Table 7.9 Individual Permissions

Column Name	Description
Name	Contains the name of the permission.
Inherited From	Shows information about why the permission is being included if the user did not specifically select the permission. Often selecting an existing permission will show the user role from which the permission was included. Also, some permissions automatically inherit other permissions. Typically, write-oriented permissions inherit associated read permissions.
Description	Contains information about each permission.

Each permission is focused on access to a particular function on the RTAC platform. To access some functionality, users may need to enable multiple permissions. *Table 7.10* shows a few examples of functionality that requires multiple permissions.

Table 7.10 Example Functionality Requiring Multiple Permissions

Functionality	Necessary Permissions
Configuring LDAP	Account Write, Network Write, and Security Write
Sending a project to the RTAC via RTAC software	Database Login, Project Write (if advanced settings are to be sent, additional permissions are required)
Sending a project to the RTAC via the web interface	Web Login, Project Write
HMI operator (includes acknowledging alarms and sending controls)	Web Login, HMI Local Write, HMI Remote Write, and Report Write

To create a new user account, navigate to the **Accounts** page. Select **Add New User**. When adding a new account, multiple user roles can be selected.

Usernames can contain any 7-bit ASCII printable character with the following exceptions: @ # $ % ^ & + = { } [ ] : ; " ' , < > ? \. Spaces are also not allowed in usernames.

Passwords can contain any 7-bit ASCII printable character with no exceptions.

The 7-bit ASCII printable characters include the entire English alphabet, upper and lower case, numbers, and some special characters included on keyboards.

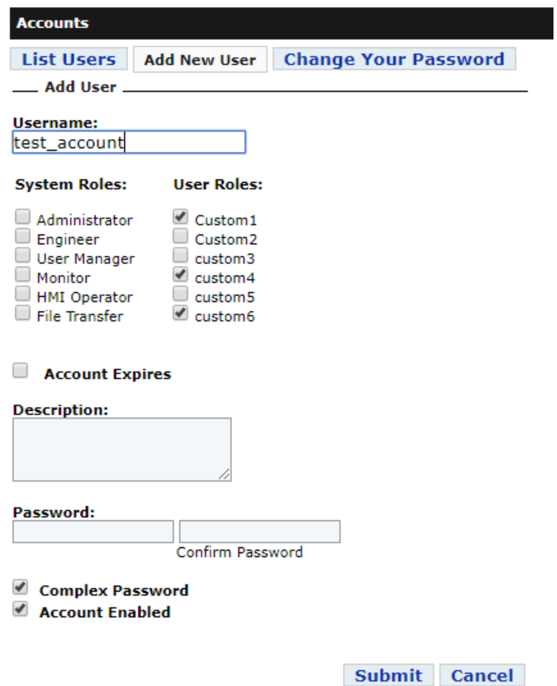


Figure 7.15 Add New User Page

In addition, accounts can be configured to expire on a specific date and time or be disabled while still existing on the unit.

Centralized User Accounts With LDAP

Many IT departments use Lightweight Directory Access Protocol (LDAP) to manage the users and devices on their corporate networks. LDAP is a powerful and flexible protocol that allows for fast information lookups from servers that are optimized for read access. The information stored on LDAP servers can be any type of record-based information that is stored in a directory structure. Examples include user and device lists, phone books, and recipes.

The RTAC includes LDAP as a mechanism for centralized user management. With LDAP, you can manage users at a central server. When a user who does not have a local account requests access to the RTAC, the RTAC will poll the central directory to verify that the user is authorized to access the unit (see *Figure 7.16*).

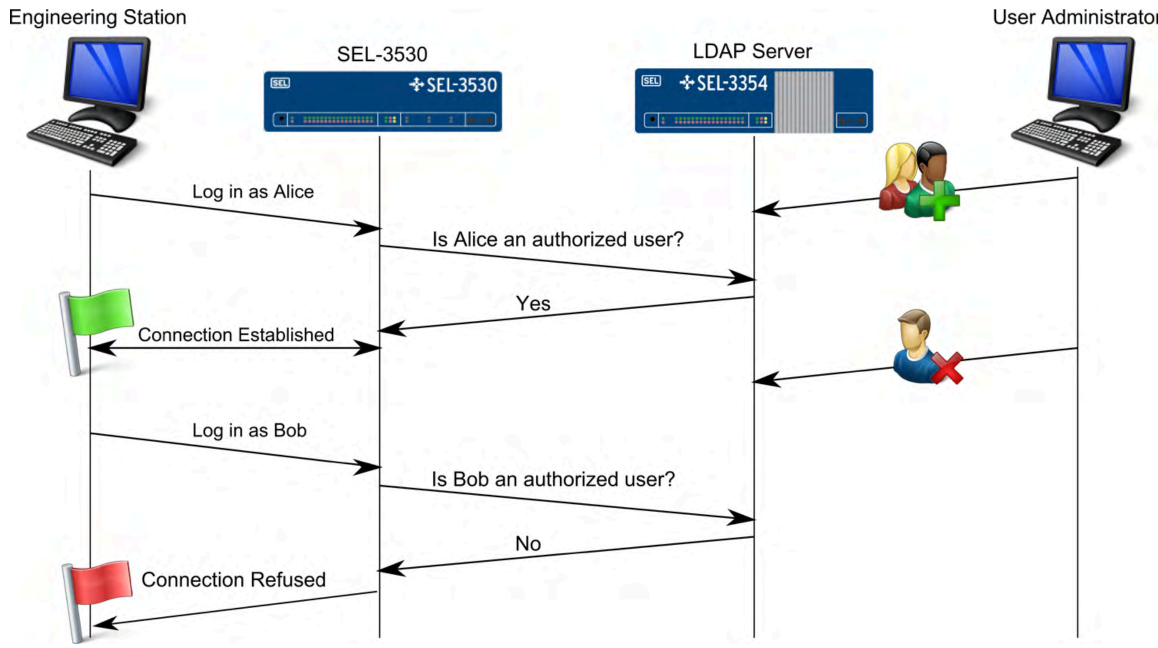


Figure 7.16 LDAP Login Process

For the RTAC to support this behavior, you must configure certain parameters in the RTAC to allow it to communicate with your LDAP server. All of these parameters are configurable through the RTAC web interface. To configure LDAP on your RTAC, access the RTAC web interface and use an account with administrative privileges to log in.

To begin, the RTAC needs to know the name and IP address of your LDAP server. Select the Host link for the navigation panel on your webpage to view and edit the Host settings (see *Figure 7.17*).

Figure 7.17 Host Settings

The RTAC needs the name and IP address of your LDAP server to determine how to contact it. The Host Settings page provides a method to statically map IP addresses with hostnames for such devices as your LDAP servers. To map an IP address to a hostname, select the Add Host button. This will show the Add Host form (see *Figure 7.18*).

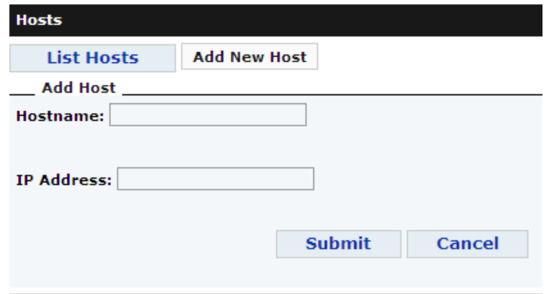


Figure 7.18 Add Host

Populate the Add Host form with the correct hostname and IP address of an LDAP server. The RTAC supports as many as 64 hosts. For your convenience, we have included in *Table 7.11* a form for your LDAP administrators to complete. Ask your LDAP Administrators to complete the form, and then map the information they provide into the RTACs Host Settings.

LDAP Certificates

LDAP requires X.509 authentication to create binds between the server and client. This ensures that attackers do not spoof the authentication server to gain unauthorized access. The RTAC supports base64 (.cer files) X.509 and CA certificates and does not support any of the binary formats. The RTAC requires that the root certificate of the LDAP servers certificate chain be stored locally. Follow the instruction in *CA Certificates on page 484* to install the required CA certificates for LDAP X.509 authentication.

LDAP Settings

Now that your RTAC knows the names and locations of your LDAP servers, we can configure the RTAC to access those servers. Select the **LDAP Settings** link from the navigation panel on your webpage to view the LDAP parameters (see *Figure 7.19*).

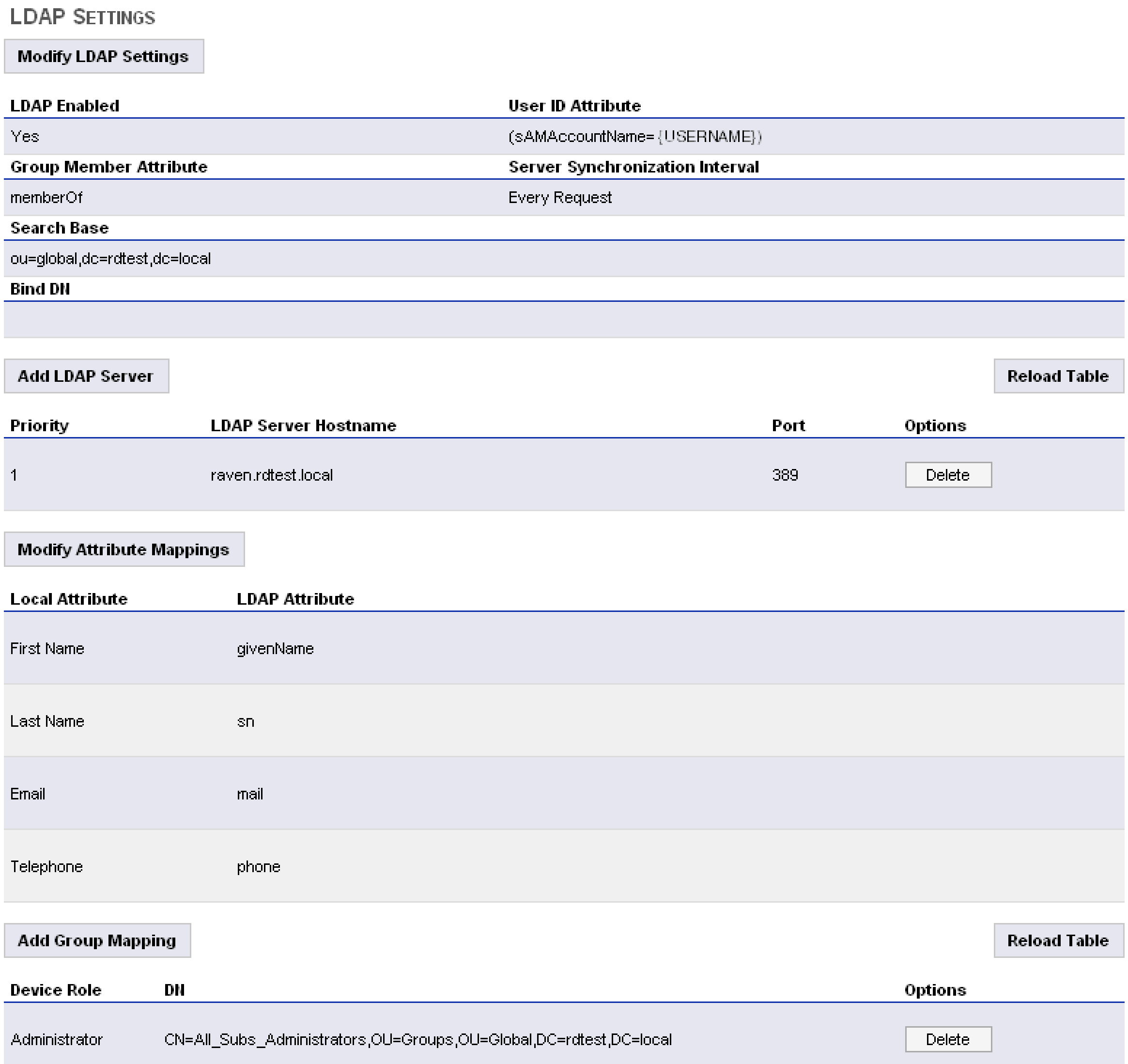


Figure 7.19 LDAP Settings

Figure 7.19 shows the **LDAP Settings** page and all the options for communicating with your LDAP servers. To simplify configuration, we have included in *Table 7.11* a form for your LDAP administrators to complete. You can then use the information from this form to populate all of the LDAP fields.

To modify the LDAP settings, select the **Modify LDAP Settings** button located at the top of the main page. This will show the Edit LDAP Settings form on the right side of the page (see *Figure 7.20*).

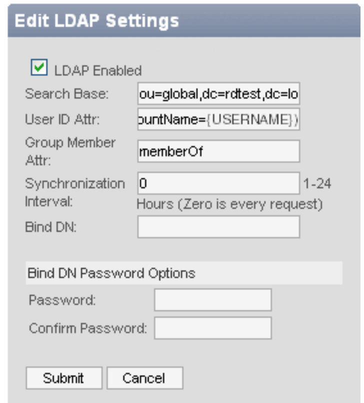


Figure 7.20 Edit LDAP Settings

You can consider the LDAP Search Base as the root directory from which to begin your user search. Form this by listing all of the components of the search base separated by commas and going from the most specific component to the broadest component. In the figure above, the Search Base configuration is ou=global,dc=rdtest,dc=local. In this search base, dc refers to domain component. The domain components combine with "." to create the search domain. In this case, the search domain is rdtest.local. The "ou" component is the organizational unit, or directory, from which to begin the search. You can interpret this search base as, "Start the search from the Global directory residing on an LDAP server in the rdtest.local domain."

One other common LDAP component is cn. The abbreviation "cn" is short for common name. It is a name that refers to a specific object that may or may not be unique. Examples of common names are groups and usernames.

NOTE
The broader your search base, the more users/groups can access the RTAC. Be aware that broader search bases can take significantly more time to search than search bases that specify organizational units or groups.

The User ID and Group Member attributes are the LDAP labels that identify the usernames and groups of system users. If you enter these incorrectly, the RTAC will be unable to determine which LDAP fields to search for usernames or privileges.

The synchronization interval setting exists to reduce the overhead associated with pulling account information, such as attributes and role mappings, from an LDAP server. The RTAC caches this information locally for centralized users, improving their access times. You can then determine an interval, from 0 to 24 hours, at which the RTAC synchronizes this information. If you set the synchronization interval to 0, then the RTAC synchronizes every time someone logs on. When you use centralized accounts, a successful LDAP bind is necessary every time someone logs on, even if the account information is cached locally.

The RTAC supports both authenticated and anonymous binds to your LDAP servers. Authenticated binds use a service account to access the LDAP server. If the LDAP service account is revoked, or if the password expires, the RTAC will be unable to access the LDAP server, and centralized users will be unable to access the RTAC. Anonymous binds forgo the use of service accounts. Learn from your LDAP administrators which method they prefer for your system.

Obtain the necessary configuration information from your LDAP administrators, and enter the settings. For your convenience, we have included a form in *Table 7.11* for your LDAP administrators to complete. Ask your LDAP administrators to complete the form, and then map the information they provide into the LDAP settings for the RTAC.

LDAP Server

To improve availability when one or more LDAP servers may be inaccessible, the RTAC supports access to multiple LDAP servers. To add an LDAP server, select the **Add LDAP Server** button. This will show the Add LDAP Server form on the right side of the page (see *Figure 7.21*).

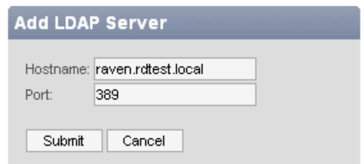


Figure 7.21 Add LDAP Server

You can identify LDAP servers by their hostnames and port numbers. Use port 389, unless your LDAP administrator specifies a different port number. Use the form in *Table 7.11* to obtain this information from your LDAP Administrators.

The RTAC allows for two LDAP servers for redundancy and increased availability. The RTAC assigns a priority to each LDAP server and queries the servers in their order of priority until it identifies the user attempting to access the RTAC, or until it exhausts the list.

Attribute Mappings

The RTAC can pull user attributes from your LDAP server and store those attributes on the local machine. To map your LDAP attributes to RTAC attributes, select the **Modify Attribute Mappings** button. This selection causes the RTAC to display the Edit Attribute Mappings form on the right side of the page (see *Figure 7.22*).

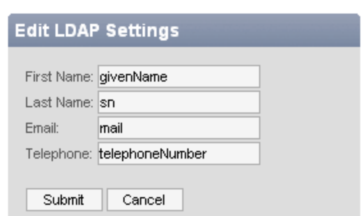


Figure 7.22 Edit Attribute Mappings

The field labels in the Edit Attribute Mappings form are the titles for the RTAC attributes. To map LDAP attributes to these local attributes, enter the appropriate LDAP attributes into the text boxes. These settings are optional. You can use the form in *Table 7.11* to obtain this information from your LDAP Administrators.

Group Mappings

The RTAC has specific device roles that you can map to LDAP group memberships. Select the **Add Group Mapping** button to configure a new group mapping. This will expand a table that shows all of the LDAP groups the RTAC can access—based upon your search base (see *Figure 7.23*).

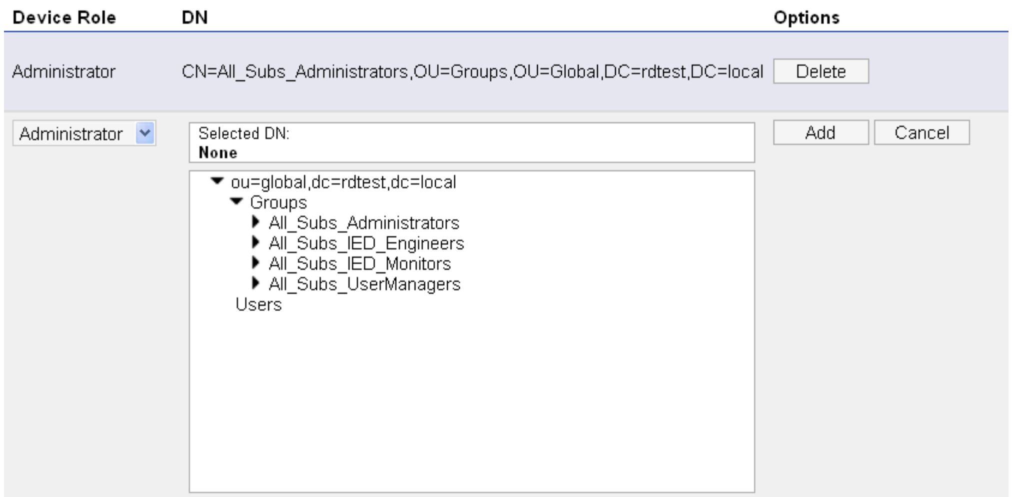


Figure 7.23 Group Mappings

To configure a group mapping, select the RTAC user type from the drop-down box on the left, and navigate through the tree to the user or group that should be given the roles privileges. Select the user or group, and select the **Add** button to create the mapping. Your server administrator may need to create new groups and assign members appropriate for these mappings. Work with your LDAP administrator, and use the form in *Table 7.11* to determine group mappings.

Table 7.11 LDAP Settings Form

LDAP Hosts (Input these settings on the Hosts page, need at least one):	
Hostname:	IP Address:
Hostname:	IP Address:
LDAP Settings (Input these settings on the LDAP Settings page):	
Search Base:	
User ID Attribute:	
Group Member Attribute:	
Bind DN (Optional, if left blank will use anonymous binds):	
Bind DN Password (Optional, required only if not using anonymous binds):	
LDAP Servers (Input these settings on the LDAP Settings page, need at least one):	
Hostname:	Port Number:
Hostname:	Port Number:
Hostname:	Port Number:

Attribute Mappings (Optional, input these settings on the LDAP Settings page):

First Name Attribute:

Last Name Attribute:

Email Attribute:

Telephone Attribute:

Device Roles (Required to map user privileges, input these settings on the LDAP settings page):

Administrator Group/User DN:

Engineer Group/User DN:

User Manager Group/User DN:

Monitor Group/User DN:

Centralized User Accounts With RADIUS

The SEL RTAC supports the basic NAS client authentication functionality of the RADIUS protocol. By configuring the RADIUS settings, you can log in using credentials not stored on the RTAC.

SEL cannot guarantee that the device will be compatible with all possible RADIUS server architectures and implementations. Configure communications with a RADIUS server on the RADIUS Settings page in the RTAC web interface.

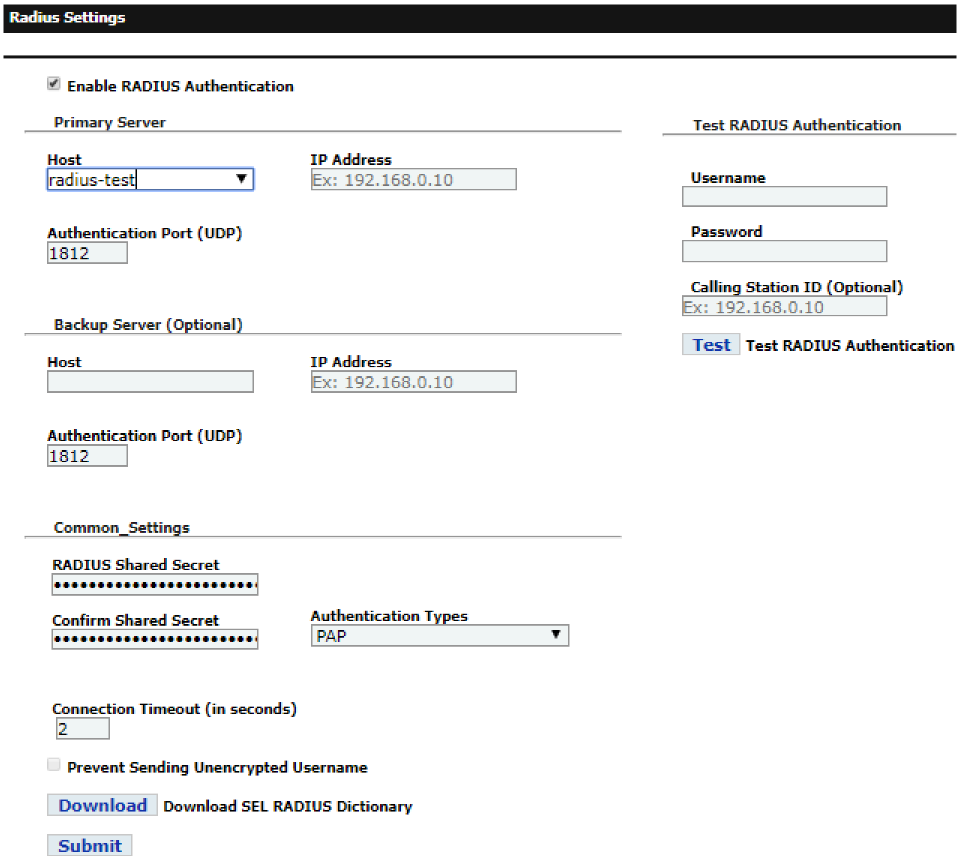


Figure 7.24 RADIUS Settings

Table 7.12 Radius Settings in the RTAC

Setting Name	Description
Enable Radius Authentication	Select this check box to enable RADIUS authentication on the RTAC
Primary Server Host and Ip Address	The host setting allows you to select any host that is already configured on the Hosts page. When selecting a host, the IP Address field will automatically populate. If the user enters a new host name with a corresponding IP address, the pair will be added to the Hosts page after submitting the RADIUS settings. In addition, if a host is selected and then the IP address is changed on this page, the change will be reflected in the Hosts page after submitting settings.
Primary Authentication Port	The UDP port on which the RTAC will attempt to contact the Primary RADIUS server. The port range is 1024–65534.
Backup Server Host and IP Address	These fields are optional. If populated, when the RADIUS client is unable to exchange information with Primary server, the Backup server will be contacted.
Backup Authentication Port	The UDP port on which the RTAC will attempt to contact the Backup RADIUS server. The port range is 1024–65534.
Shared Secret	The Shared Secret is a string that is determined by the RADIUS server and must match in order for authentication requests to be successful.
Authentication Types	The RTAC supports, PAP, EAP-PEAPv0/MSCHAPv2, EAP-TTLS/PAP. This is the method in which the RTAC will communicate with the RADIUS server. When selecting an EAP method type, the RTAC will require the full CA certificate chain for the RADIUS server. This certificate will be imported on the CA certificates page.

Setting Name	Description
Connection Timeout	This is the time for which the RTAC will wait for the RADIUS server to respond to an authentication request. The RTAC will attempt to connect to the Primary server three times before waiting for the connection time-out period. If the RTAC does not receive a response, then the RTAC will attempt to authenticate and repeat this process with the backup server if configured. The range is 1–10 seconds.
Prevent Sending Unencrypted Username	Select this check box to hide usernames by using "anonymous" as a username to send requests to the RADIUS server. When this is done, the actual username is sent in the encrypted portion of the message.

The RTAC SEL RADIUS Dictionary can be downloaded from the RADIUS Settings page. The dictionary lists all current account types configured on the RTAC including both default roles and custom roles. This file defines the SEL vendor-specific attributes that must be defined on your RADIUS server and which are used by your device to appropriately grant or restrict privileges for users.

Communications with the RADIUS server can also be tested via the RADIUS Settings page. Enter the username and password configured on the RADIUS server and the RTAC will send the credentials to confirm or deny authentication.

Security Logging

Logging provides a method for you to audit authorized and unauthorized access and changes in the system. You can log changes in the RTAC by placing a variable in the Tag Processor and enabling logging of that variable. The RTAC has ten specific User Sessions tags that can log user activity on the RTAC (see *Figure 7.25*). See also *Section 5: Web HMI and Reports* for details on logging.

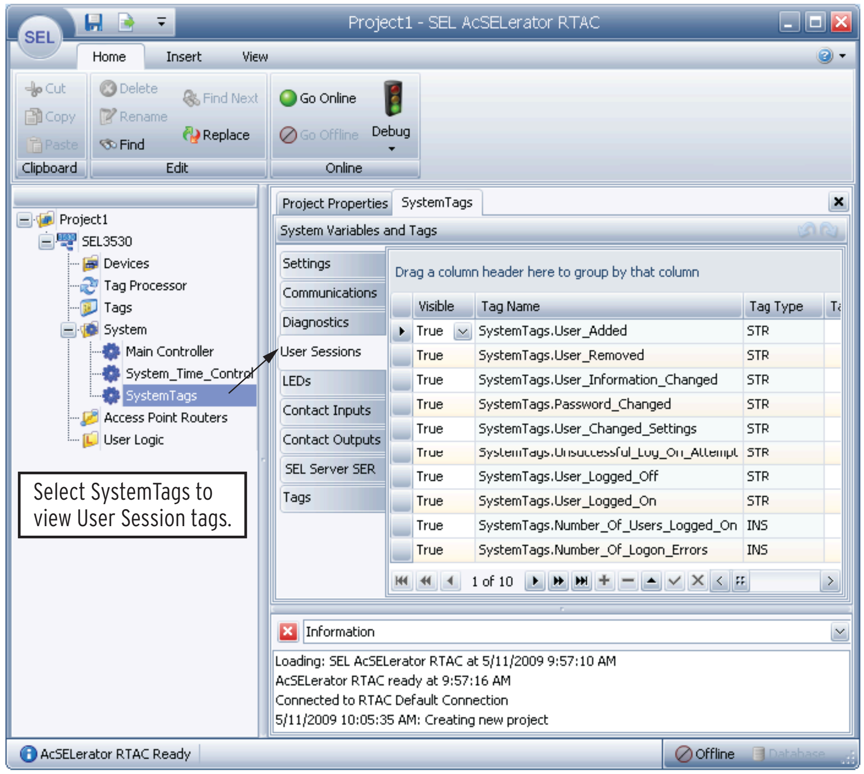


Figure 7.25 System User Tags

Example 7.6 Toggle Alarm Contact Upon User Login

By default, ACSELERATOR RTAC enables logging of User_Logged_On and User_Logged_- Off. ACSELERATOR RTAC also creates a log status bit automatically for every log entry. Each time ACSELERATOR RTAC logs the tag, this status bit toggles True/False to indicate the logging action. You can map this status bit in the Tag Processor grid in the same way you can map any other digital indication point.

This example illustrates mapping this bit to the RTAC alarm contact. When someone logs on, the alarm contact (OUT101) pulses closed, and the front-panel LED illuminates briefly.

Step 1. Open a blank project in ACSELERATOR RTAC. This example assumes that the default logging of User_Logged_On is in the Tag Processor grid.

Step 2. Press the save icon or **<Ctrl+S>** to compile the blank project and generate the Tag Processor code.

Step 3. Click on the **Code** tab at the bottom of the Tag Processor to reveal the code that the Tag Processor generated (see *Figure 7.26*).

Step 4. Notice that the code is in a program called TagProcessing. Data within that program are global; you can map these data within ACSELERATOR RTAC.

Step 5. Notice that each log entry is a variable structure called Logger1, Logger2, etc. The logged bit is in that variable structure.

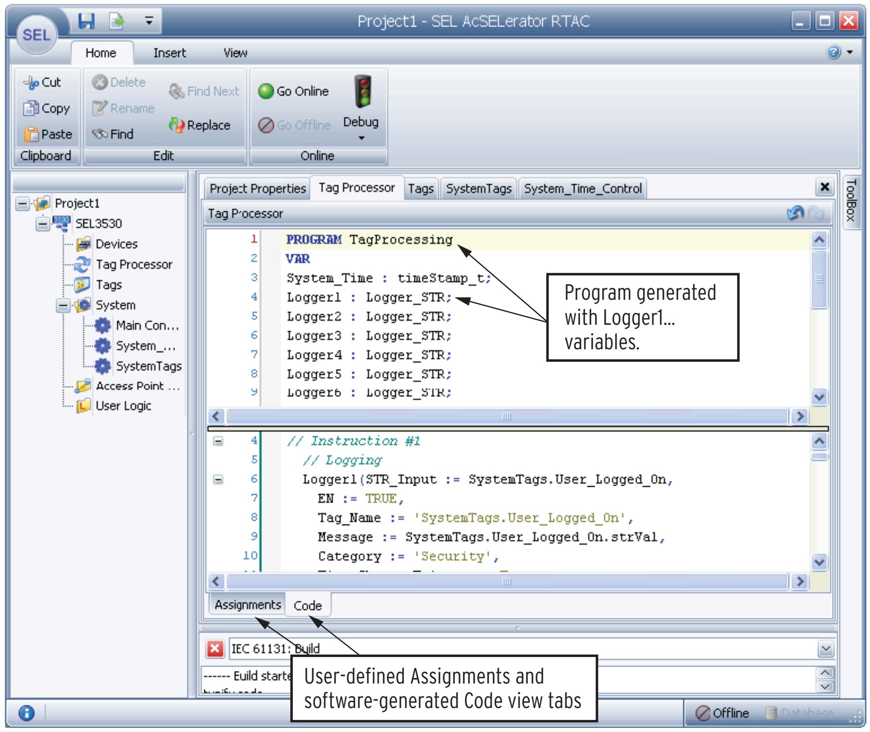


Figure 7.26 Tag Processor Code View

Step 6. Click on the **Assignments** tab on the bottom of the Tag Processor to return to the grid view. We will assign the log bit to the alarm contact in the grid.

Step 7. Press the + icon to create a new line, if necessary.

Step 8. On the first blank line in the **Destination Tag Name** column, enter the operSet attribute for the alarm contact tag. The operSet attribute, when True, causes the alarm contact to close.

SystemTags.OUT101.operSet.ctlVal

Step 9. On the same line in the **Source Expression** column, enter the full tag name of the bit that indicates that RTAC has logged a login action. This tag name contains the program name, the variable in the program, and the structure entity from that variable:

TagProcessing.Logger1.event_logged

Build	Destination Tag Name	Source Expression
True	SystemTags.OUT101.operSet.ctlVal	TagProcessing.Logger1.event_logged

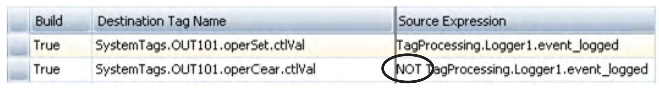
Figure 7.27 Event Logged Status

Step 10. On the next blank line under the **Destination Tag Name** column, enter the operClear attribute for the alarm contact tag. The operClear attribute appears as follows:

SystemTags.OUT101.operClear.ctlVal

Step 11. On the same line in the Source Expression column, enter the full tag name of the bit that indicates that RTAC has logged a login action, and negate that action by adding a NOT before the rest of the tag name. This will cause RTAC to toggle the set and clear attributes when the event_logged value toggles.

NOT TagProcessing.Logger1.event_logged



Build	Destination Tag Name	Source Expression
True	SystemTags.OUT101.operSet.ctlVal	TagProcessing.Logger1.event_logged
True	SystemTags.OUT101.operCear.ctlVal	NOT TagProcessing.Logger1.event_logged

Figure 7.28 Inverted Event Logged Status

Step 12. Save and download the project into the RTAC.

Step 13. Open a web browser; enter https:// and the IP address of the RTAC.

Step 14. Log in at the prompt and notice that the RTAC alarm relay clicks closed and then open and that the front-panel alarm LED flashes on and then off at the same time. We did not map the alarm LED, because the RTAC has mapped the LED internally to the alarm contact.

Syslog

The Syslog protocol, defined in RFC 3164, defines how a device can send system event notification messages across IP networks to remote syslog servers. Syslog is commonly used to send system logs such as security events, system events, and status messages. For example, a printer can send a message that it is running low on ink. The syslog packet size is limited to 1024 bytes and is formatted into three parts: PRI, HEADER, and MSG.

1. **PRI**: A number enclosed in angle brackets that represents both the Facility and Severity of the message. This number is derived as:

   PRI = Facility code • 8 + Severity

   The Facility code in the RTAC is fixed at 23. The severity is derived from the user-configurable Logging Priority string in the Tag Processor found in the logging layout. *Table 7.13* shows the numeric equivalent of various case-sensitive strings. Any strings that do not precisely match those in *Table 7.13* will have a numeric code of 6. For example, an item configured in the Tag Processor with a Logging Priority of Critical would have a <186> in the PRI field of the syslog message.

Table 7.13 Syslog Message Severities Supported in the RTAC

Numeric Code	Logging Priority String
0	Emergency
1	Alert
2	Critical
3	Error
4	Warning

Numeric Code	Logging Priority String
5	Notice
6	Informational
7	Debug

2. **HEADER**: The message origination time stamp and source. Time stamps are based on the time of the originating host, so it is critical to have all devices time synchronized to accurately correlate events. The source of the message is the RTAC hostname, as defined on the RTAC web interface dashboard.
3. **MSG**: The human readable body of the message. Messages are constructed from fields in the Tag Processor logging layout and associated tag information as shown in *Table 7.14*.

Table 7.14 Syslog MSG Configuration

Message Component	Description
Logging Device	Fixed name of RTAC Logger
Event Time Stamp	Time stamp of the associated tag
Log Severity	Logging Priority specified in Tag Processor
Log Category	Logging Category specified in Tag Processor
Tag Name	Destination Tag specified in Tag Processor
Log Message	Logging Message specified in Tag Processor
Logging Comment	Logging Comment specified in Tag Processor

Note the following Syslog message generated when user SEL logged in to the RTAC web interface on March 21, 2015 at 12:09:58.593:

<190>Mar 21 2015 12:10:08 SEL-3530-xxxxxx RTAC Logger: 2015-03-21 12:09:58.593, Informational, Security, SystemTags. User_Logged_On, 'SEL logged on device via Web'.

Configure the RTAC to send Syslog messages by performing the following steps:

Step 1. Configure logging as defined the in *Security Logging on page 507*, using the logging priorities defined in *Figure 7.29*.

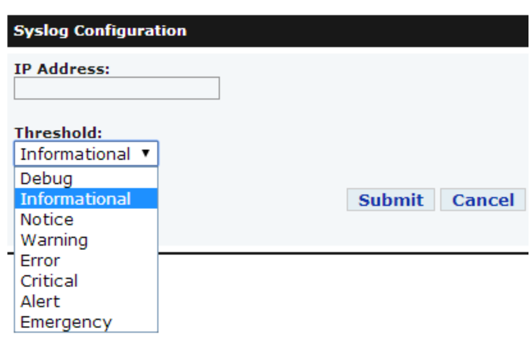


Figure 7.29 Configure Syslog

Step 2. In the RTAC web interface, click on **Syslog** under the **Network** heading.

Step 3. Click on **Add Syslog Destination**.

Step 4. Enter the IP Address of the Syslog server. Ensure the RTAC and Syslog server are on the same subnet or you have sufficient routing configured such that the RTAC can reach the Syslog server.

Step 5. Select the threshold that correlates to the logging priority you configured in the Tag Processor logging settings.

The threshold is rated from top to bottom, with Debug as the lowest priority and Emergency as the highest. All logged items that are of the threshold priority or above are sent as Syslog messages to the Syslog server. For example, configure the threshold as Critical. All logs with a priority of Error, Warning, Notice, Informational, or Debug are not sent via Syslog. All logs that are Critical, Alert, or Emergency are sent via Syslog. The RTAC will store as many as 10,000 Syslog messages before writing over the oldest entry.

SECTION 8

Time Synchronization

Overview

There are two major aspects to date and time management in the SEL Real-Time Automation Controller (RTAC) family of products. The first is the synchronization and sourcing of date and time. For example, the RTAC can accept IRIG-B input, make adjustments for geographical location, and then source DNP time to an IED. The second major aspect is the time stamp associated with each piece of data. Data that enter or exit with or without time stamps need to be processed appropriately.

Precision Time Protocol (PTP)

RTAC supports PTP slave-clock functionality as per the IEEE 1588-2008 standard. PTP is an industry standard protocol for time synchronization of distributed nodes in packet-based networks like Ethernet.

In firmware versions R143 and later, PTP is supported via bonding in backup mode. PTP is also supported when PRP is enabled.

The RTAC supports listening to a single grandmaster clock. The network may contain multiple master clocks, which will communicate amongst themselves to elect a grandmaster clock. The RTAC will then listen to the grandmaster clock messages and synchronize to that signal.

Definitions

PTP Domain: A PTP domain is a logical grouping of clocks that are synchronized to each other.

PTP Master Clock: A PTP master clock is a clock capable of providing a time source to which other clocks on that path synchronize.

PTP Grandmaster Clock: A PTP grandmaster clock is a master clock that has been elected as the primary time source for the PTP domain via the best master clock algorithm.

PTP Slave Clock: A PTP slave clock is a clock capable of synchronizing to a PTP master clock. The RTAC is a slave clock.

PTP V1 vs. V2

PTP V2 signaling is a significant departure from V1 and is not backwards compatible. The RTAC supports PTP V2 signaling.

Accuracy

When the RTAC is synchronized to a PTP time source and used to distribute time to additional IEDs, the outgoing signal will be accurate to within 500 µs of the RTAC system time. Certain applications, such as IEEE C37.118 synchrophasor measurement, rely on accuracies of one microsecond or better.

Profiles

The RTAC supports the delay request response and peer-to-peer default PTP profiles according to the IEEE 1588-2008 standard. A PTP profile is a set of required and prohibited PTP options as well as the ranges and defaults of configurable attributes that is used to meet the requirements of a particular application.

In firmware versions R144 and later, PTP Profile is supported.

IEC/IEEE 61850-9-3 Power Profile

The IEC/IEEE 61850-9-3 profile allows for compliance with the highest synchronization classes of IEC 61850-5 and IEC 61869-9.

IEEE C37.238-2011 Power Profile

The IEEE C37.238-2011 power profile is an extension of IEC/IEEE 61850-9-3 that provides additional information about real-time clock performance and optional local time zone information. These additions were meant to position this profile as a replacement for IRIG-B and to support local display applications.

Table 8.1 lists the attributes of each profile and the corresponding compliance level of the RTAC.

Table 8.1 Profile Comparison and RTAC Support

	IEEE C37.238-2011	IEC 61850-9-3	Peer-to-Peer	Delay-Request	RTAC Implementation
General PTP Attributes					
Number of PTP Ports	1	1 or more	NA	NA	1
Log Announcement Interval	1 second	Variable	Range (0 to 4) Default: 1	Range (0 to 4) Default: 1	1 second
Log Sync Interval	1 second	Variable	Range (–1 to 1) Default: 0	Range (–1 to 1) Default: 0	1 second
Announce Receipt Timeout	3	3	Range (2 to 10) Default: 3	Range (2 to 10) Default: 3	3 seconds
Log Min delay request interval	1 second	Variable	Range (0 to 5) Default: 0	Range (0 to 5) Default: 0	1 second
Priority 1	255 (slave-only)	255 (slave-only)	128	128	128
Priority 2	255 (slave-only)	255 (slave-only)	128	128	128

	IEEE C37.238-2011	IEC 61850-9-3	Peer-to-Peer	Delay-Request	RTAC Implementation
Domain Number	238	Default: 0	Default: 0	Default: 0	Settable 0–127, 238
Path Delay Mechanism					
Delay Mechanism	Peer-to-Peer (P2P)	Peer-to-Peer (P2P)	Default: P2P	Default: E2E	Settable (P2P, E2E)
Best Master Clock Determination	Best master clock (BMC) algorithm	Best master clock (BMC) algorithm	Best master clock (BMC) algorithm	Best master clock (BMC) algorithm	Best master clock (BMC) algorithm
Clock Accuracy	254 (slave-only)	35 (optional) (slave-only)	NA	NA	254
Clock Class	255 (slave-only)	NA	NA	NA	255
Transport Mechanism					
Transport	IEEE802.3	IEEE802.3	NA	NA	Settable (IEEE802.3, UDP)
VLAN Tagging	Required: VLAN Priority = 4, VLAN ID = 0	Not required	NA	NA	Supported
Management Mechanism					
SNMP Support	Optional	Optional	NA	NA	Not supported
Clock Types					
One-Step	Required	Required	NA	NA	Supported
Two-Step	Required	Required	NA	NA	Supported
clockIdentity	EUI-64 constructed based on EUI-48	NA	NA	NA	EUI-64 constructed based on EUI-48
TLV					
Alternate Time Offset Indicator	Required	NA	NA	NA	Supported
Management	Required	NA	Management message	Management message	Supported
Management Error Status	Required	NA	NA	NA	Supported
Organization Extension	IEEE C37.238	NA	NA	NA	IEEE C37.238
Order of TLVs	Order requirement: ORGANIZATION_EXTENSION... ALTERNATE_TIME_OFFSET... Required: organizationID organizationSubType Grandmaster ID GrandmasterTimeInaccuracy NetworkTimeInaccuracy	NA	NA	NA	Supported

PTP Settings

The ACSELERATOR RTAC software provides settings for configuring PTP on the **System_Time_Control** page.

Setting	Description
Set_PTP_Transport (UDP, IEEE 802.3)	Set the transport scheme to UDP for routable synchronization over Layer 3 communications. Use this setting when the RTAC must synchronize to a master/grandmaster clock over a wide area network. Set the transport scheme to IEEE 802.3 for Layer 2 communications. Use this setting for time synchronization among devices within a local network.
Set_PTP_Domain (0–127)	Set the Set_PTP_Domain setting in accordance with the master-clock configuration of your system.
Set_PTP_Path_Delay_Mechanism (P2P, E2E)	Set the path delay mechanism to determine the manner in which path delay is calculated to provide an accurate offset at the slave clock. Peer delay request is the peer-to-peer (P2P) calculation method by which each device in the network contributes to the calculation of the overall path delay. This occurs through calculation of the delay to each devices peer and adding it to the synchronization message. This method is highly accurate but usually requires a specially engineered network because all intermediate switches and routers must be IEEE 1588 compliant. The end-to-end (E2E) path-delay mechanism calculates the overall path delay by using only timing information from the grandmaster and slave clocks. This method allows for greater versatility than does P2P because it permits non-IEEE-1588 compliant switches and routers within the communications network.
Enable_PTP_Power_Profile	Enables the RTAC to process PTP Power Profile.
Set_PTP_Interface_1	Select an Ethernet interface that will participate in PTP time synchronization.
Set_PTP_Interface_2	Select a second Ethernet interface that will participate in PTP time synchronization simultaneously with PTP Interface 1.

Time Source Settings

The RTAC can receive time synchronization from IRIG-B, precision time protocol (PTP), network time protocol (NTP), DNP, and i870. If one time source fails, the RTAC begins using the next highest priority time source automatically. You can configure the priority of each time source in ACSELERATOR RTAC SEL-5033 Software. Default order is IRIG-B, PTP, NTP, i870, and DNP. DNP and i870 have configurable timeouts to determine if they have failed as a time source, whereas predetermined criteria determine whether IRIG-B, PTP, and NTP are failed or good.

System time-synchronization parameters are contained in the **System_Time_Control** tab. The RTAC also has time-related settings in the protocols that support time synchronization, but these settings are used to adjust time-stamped data and not for synchronization of the RTAC system time. IRIG-B is the default primary system time source. The RTAC adjusts selectively the valid time-synchronization messages that it receives from IRIG-B to local time from universal time (UTC) and for Daylight-Saving Time (DST) according to the configuration in the System_Time_Controller. The system time is set to this adjusted time. If the IRIG-B time message fails, the RTAC uses secondary time source messages for time synchronization. The default secondary time source is PTP. If the PTP time source fails, the RTAC uses the tertiary source for time. If the RTAC determines that all time sources have failed, the free running clock will provide the system time. See *Figure 8.1*.

The RTAC has a unique time-synchronization algorithm that provides a high-accuracy clock even when all time sources fail. Rather than setting the system clock with every time-synchronization message it receives from a time source and not setting the clock at all if there is no valid time source, the RTAC uses time-synchronization messages to tune the system time. This implies that the system time will become gradually closer to the time-synchronization message time until it is within the specified accuracy. An exception is when the RTAC system time is outside a reasonable range (one second) from the time-synchronization message. When the RTAC detects this situation, it sets the Time_Sync_Outside_Slew_Threshold pin and sets the system clock immediately to match the time-synchronization message.

If the system loses all time sources, the system retains the tuning information and adjusts the free-running clock constantly to account for inherent drift. The constant tuning adjustments reduce clock drift greatly, and the system maintains accuracy of time measurements even in the absence of a valid time source.

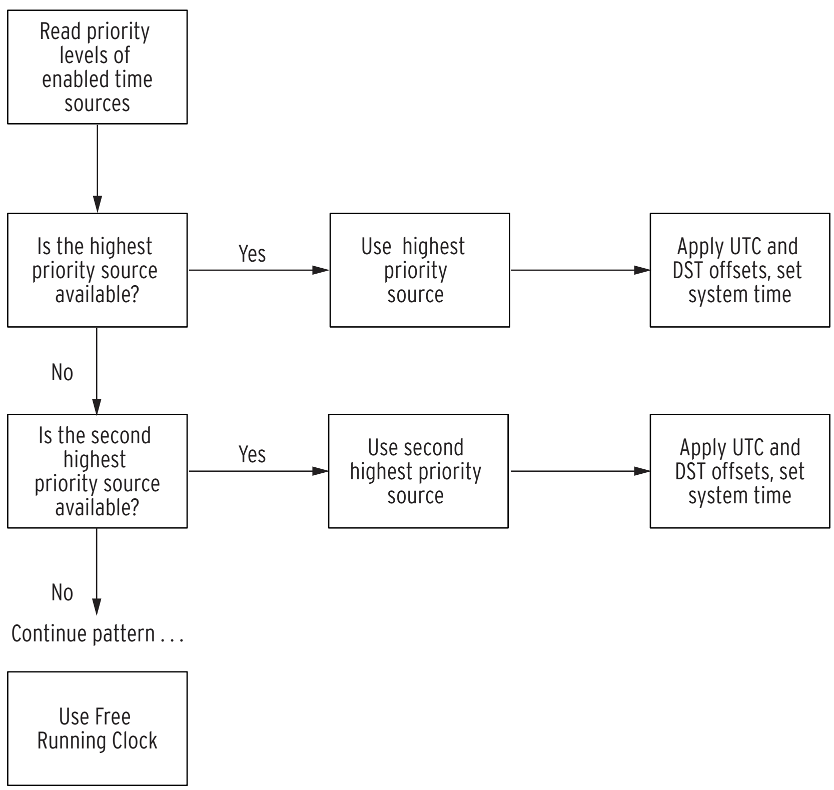


Figure 8.1 Time-Synchronization Message Selection

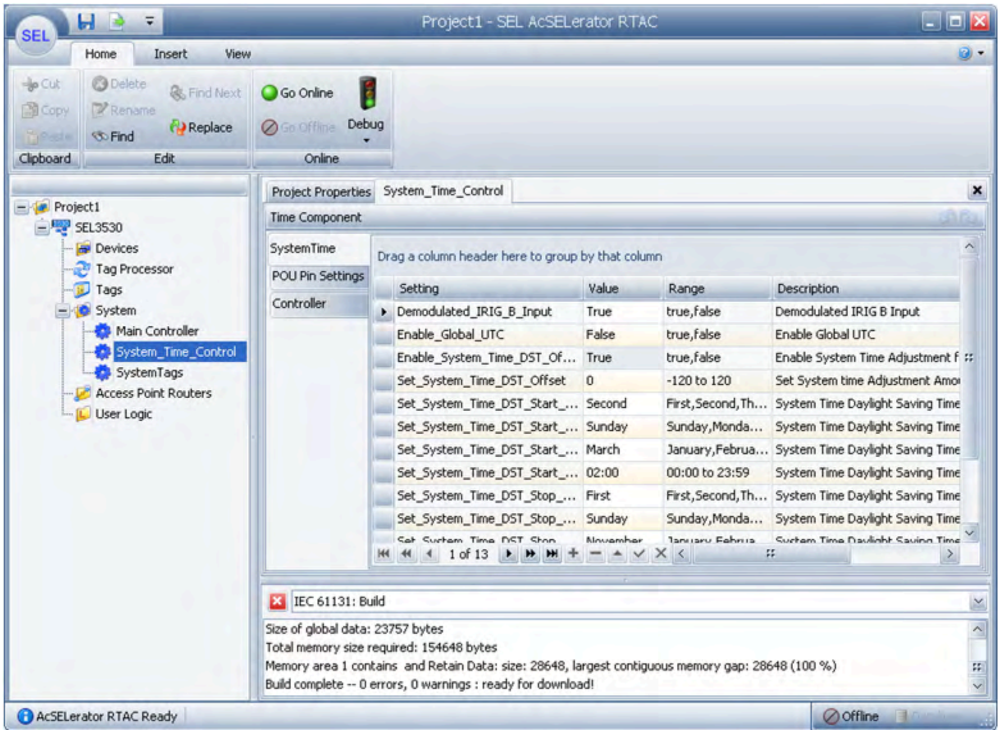


Figure 8.2 System Time Settings

Enable DST offset by setting Enable_System_Time_DST_Offset to True in the **SystemTime** settings tab. Configure related DST parameters to determine start/stop date and time of DST, offset, etc. Other parameters in the SystemTime tab are as follows.

Table 8.2 SystemTime Parameters

Parameter	Description	Default
Demodulated_IRIG_B_INPUT	Set True to receive demodulated IRIG-B, False for modulated.	True
Enable_Global_UTC	Set True to revert all system time offsets to 0 and disable all DST settings.	False
Enable_System_Time_DST_Offset	Set True to apply daylight-saving time offset configuration to the system time.	True
Set_DNP_Failure_Timeout	The maximum number of seconds between DNP time-synchronization messages from a DNP client. Excursion of this timeout marks the DNP time source as failed.	10800 seconds
Set_i870_Failure_Timeout	The maximum number of seconds between IEC 60870-5 time synchronization messages from an IEC 60870-5 client. Excursion of this timeout marks the IEC 60870-5 time source as failed.	10800 seconds
Set_PTP_Transport	Select the appropriate PTP network transport scheme (UDP, IEC 802.3)	UDP
Set_PTP_Domain	Select the appropriate PTP domain (0–127).	0
Set_PTP_Path_Delay_Mechanism	Select the appropriate path delay mechanism (P2P, E2E).	P2P
Set_PTP_Interface	Select the interface on which to listen to PTP traffic.	None
Apply_System_Time_UTC_Offset_Globally	Set to True to copy the user configured Set_System_Time_UTC_Offset POU pin value to all POU time settings and automatically determine if the time needs to use the value or not. This setting is mutually exclusive with Enable_Global_UTC and, once set, will disable manual configuration of UTC offsets in each POU.	True

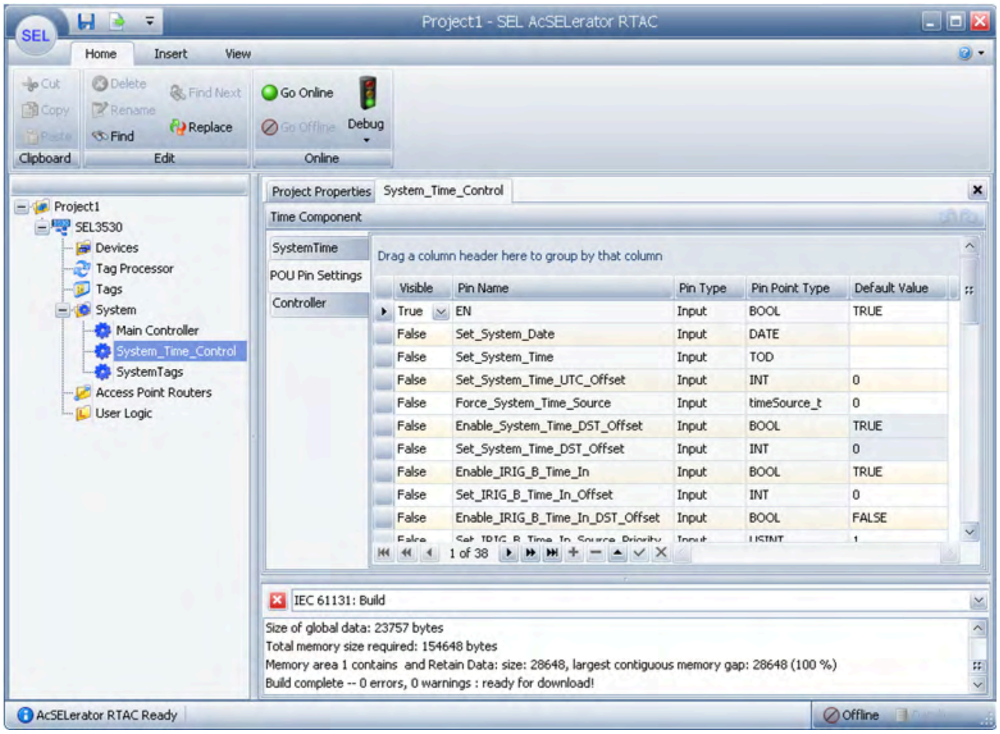


Figure 8.3 SystemTime POU Pin Settings

Set time source priorities, DST and UTC offsets per time source, and NTP server information in the system time **POU Pin Settings** tab.

Table 8.3 Pertinent System Time POU Pin Settings

Parameter	Description	Default
Set_System_Date	Manually set system date (online only). Equivalent to setting on the web interface and ignored if a better time source is available.	
Set_System_Time	Manually set system time (online only). Equivalent to setting on the web interface and ignored if a better time source is available.	
Set_System_Time_UTC_Offset	Offset in minutes from UTC. Used typically to adjust to local time.	0
Enable_IRIG_B_Time_In	Enable IRIG-B as a time source.	True
Set_IRIG_B_Time_In_Offset	Configurable offset if necessary.	0
Enable_IRIG_B_Time_In_DST_Offset	Apply DST offset to IRIG-B time.	False
Set_IRIG_B_Time_In_Source_Priority	Priority of IRIG-B as a time source. Highest priority is 1.	1
Enable_IRIG_B_Time_Out	Enable IRIG-B time output time synchronization.	True
Set_IRIG_B_Time_Out_Offset	Configurable offset if necessary.	0
Enable_IRIG_B_Time_Out_DST_Offset	Apply a DST offset to IRIG-B output time messages. The DST settings are in the **SystemTime** settings tab.	True

Parameter	Description	Default
Force_System_Time_Source	Force which time source the system will use. This is an enumerated value for the time sources, defined as follows: 0 - NULL 1 - IRIG-B 2 - NTP 3 - DNP 4 - Free Running 5 - IED_Protocol 6 - I6870 7 - PTP	0
Enable DNP_Time_In	Enable DNP as a time source.	True
Set_DNP_Time_In_Offset	Configurable offset if necessary.	0
Enable_DNP_Time_In_DST_Offset	Apply DST offset to IRIG-B time.	False
Set_DNP_Time_In_Source_Priority	Priority of DNP as a time source. Highest priority is 1.	3
Enable_DNP_Time_Out	Enable DNP clients to output time synchronization messages.	True
Set_DNP_Time_Out_Offset	Configurable offset if necessary.	0
Enable_DNP_Time_Out_DST_Offset	Apply a DST offset to DNP output time messages. The DST settings are in the **SystemTime** settings tab.	False
Enable_NTP_Time_In	Enable NTP as a time source.	True
Set_NTP_Time_Server_1_IP Set_NTP_Time_Server_2_IP Set_NTP_Time_Server_3_IP	IP address of NTP servers. As many as three are configurable.	192.0.0.1
Set_NTP_Time_In_Source_Priority	Priority of NTP as a time source. Highest priority is 1.	2
Enable_NTP_Time_Out	Enable NTP clients to output time-synchronization messages.	True
Enable_PTP_Time_In	Enable PTP as a time source.	True
Set_PTP_Time_In_Source_Priority	Priority of PTP as a time source. Highest priority is 1.	2
Enable_i870_Time_In	Enable i870 as a time source.	True
Enable_i870_Time_In_DST_Offset	Apply DST offset to i870 time.	True
Enable_i870_Time_Out	Enable IEC 60870-5 clients to output time synchronization messages.	True
Enable_i870_Time_Out_DST_Offset	Apply a DST offset to i870 output time messages. The DST settings are in the **SystemTime** settings tab.	True
Set_i870_Time_In_Offset	Configurable offset if necessary.	0
Set_i870_Time_Out_Offset	Configurable offset if necessary.	0
Set_i870_Time_In_Source_Priority	Priority of i870 as a time source. Highest priority is 1.	5

Source Time Stamps

The logic engine uses the RTAC system time to time-stamp local data (including system tags), contact I/O, and non-time-stamped data it receives from client protocols. Typically, DNP and SEL IEDs communicating with the RTAC obtain time synchronization with the RTAC through IRIG-B or the DNP3 protocol. In those cases, the RTAC logic engine stores these time stamps without change.

The RTAC may receive other time-stamped data from IEDs that are either not time synchronized by the RTAC or that perhaps have different UTC or DST offsets. These external data time stamps will not be in reference to the RTAC system time. Configuration parameters in protocols that support time synchronization provide UTC and DST offset parameters per device. Using these parameters, the RTAC adjusts time stamps on data it reads from each IED to bring all time into reference with RTAC time. The RTAC database stores tag values with adjusted time stamps.

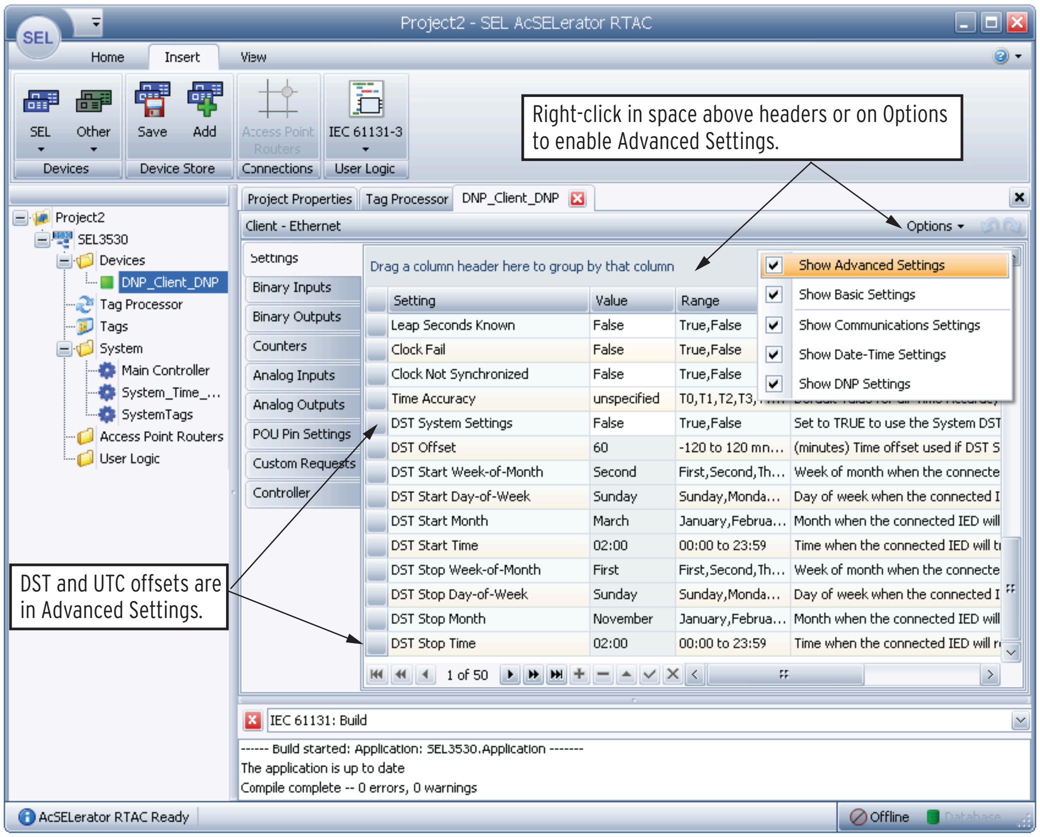


Figure 8.4 Client Time Settings

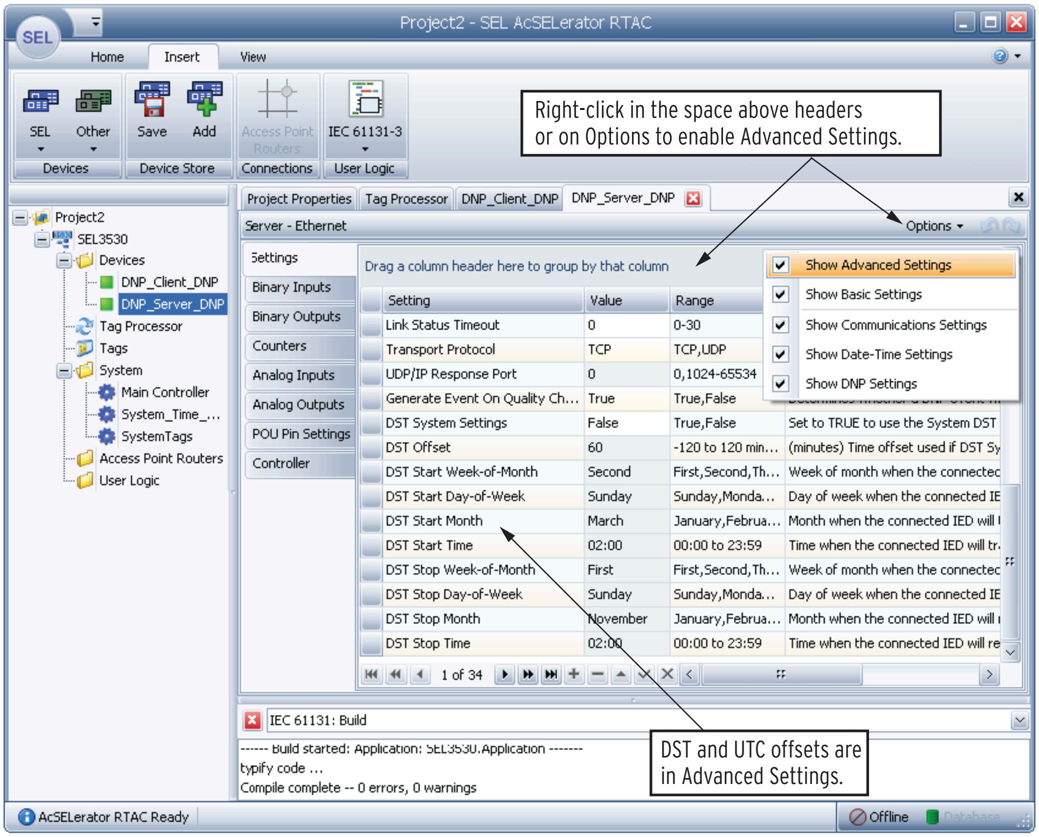


Figure 8.5 Server Time Settings

Destination Time Stamps

A DNP server device will return time-stamped event data to a polling RTAC client. The RTAC adjusts these data for any configured UTC and DST offsets before storing these data in the RTAC database. The DNP server also has DST and UTC offsets, which you can configure to alter further the time stamps of the event data. As an example, you could configure the RTAC to store and display all data in local time, but the polling client is expecting all returned data time stamps to be in UTC time. By your applying the correct offsets in the DNP server configuration, the RTAC will adjust all data time stamps to UTC time.

SECTION 9

Custom Logic

Overview

This section contains description of three specific tasks where you can take advantage of the powerful and easy-to-use IEC 61131-3 programming environment in ACSELERATOR RTAC SEL-5033 Software.

1. Creation of a Source Expression in the Tag Processor. Use IEC 61131-3 Structured Text to manipulate a combination of source tags, and then transfer the result to the Destination Tag.
2. Creation of programs, functions, and function blocks in the IEC 61131-3 User Logic area. Refer to *Appendix B: IEC 61131-3 Programming Reference* for function and data type reference.
3. Creation of custom logic in the IEC 61131-3 Continuous Function Chart (CFC) under a device **Controller** tab. This area contains all the controlling elements (PINs) related to that device. Use this feature for advanced applications.

Logic in the Tag Processor

You can enter structured text statements that use any global tags within the ACSELERATOR RTAC database in the Source Expression column in the Tag Processor. Use this functionality to scale values, combine values, or logically manipulate data for transfer to the Destination tag. See *Section 3: Tag Processor* for details and examples of writing logic in the Tag Processor grid.

IEC 61131-3 Programming Overview

Create custom programs, functions, and function blocks by inserting an IEC 61131-3 User Logic block. The programming language is easy to learn and powerful. Insert a block, and select which of three types of logic blocks you want to create.

Program

- A program is the most basic logic block. You cannot use functions and function blocks without first inserting a program.
- Programs can be standalone units. They contain logic to perform a single task, perform multiple tasks, or invoke functions and function blocks.
- A program must have a unique name in a project.

- Programs can access global system variables and act directly on them. In the examples this section provides, programs toggle light-emitting diodes (LEDs).
- Many programs can be in a single project.
- Programs act independently of each other.
- Local scratch pad variable (VAR) data from one program are not accessible by any other program.
- Any actions one program performs on global variables such as LEDs, etc. are available to other programs.
- Programs can call any number of functions or function blocks.
- Programs can contain any number of variables.

Function

- A function is a routine that a program or calling logic block can call to perform repetitive tasks. SQRT() is a good example of a standard function.
- Multiple programs can access any given function.
- A function must have a unique name in a project.
- It is not necessary to declare or set up functions in the calling logic block.
- Generally, functions do not read global system variables directly unless the calling logic block passes the values of those variables to the function.
- Generally, functions do not write directly to any global system variables.
- The logic block calls functions in the following two ways:
  - By assignment: SomeVariable := FunctionName (InValue1, InValue2);

    InValue1 and InValue2 are VAR_INPUTs in the function.

    SomeVariable is a locally defined VAR the same type as the function.

    The output of the function is assigned to SomeVariable.
  - By call using VAR_OUPUTs: FunctionName (real, out1 => LocVar1, out2 => LocaVar2)

    out1 and out2 are VAR_OUTPUTs in the function.

    LocVar1 and LocVar2 are locally declared VARs.

    The data type is real.
- The function call assignment must have the same number of arguments in parentheses as the function contains input variables.
- The function must assign return value(s) to the function name or OUTPUT_VARs before the function returns control to the logic block (see *Figure 9.6*).
- The design of a function is to always return a certain output for the same inputs. For example, SQRT(3) always returns the same value.
- A function does not retain values from one call to the next unless the function uses VAR_STAT (static variables).

Function Block

- A function block is a routine for which a program or calling logic block can have definitions for multiple instances to perform specific tasks. Each instance is unique and separate.
- Function block instances retain any values they use. This makes function blocks unique from functions. For each use of a function block instance, the instance updates values according to inputs and previous results.
- A function block must have a unique name in a project.
- The VAR section of a program or calling logic block contains the definition for a function block instance.
- A logic block can contain definitions for multiple instances of a function block (each with a unique name).
- Function blocks typically do not directly access and act on global system variables.
- To activate or cause function blocks to run, reference their data elements: e.g., FB_ instanceName (FB_INvarName := someValue).
- Use the calling logic block at any time after function block activation to access the values of each function block instance. Do the following to accomplish this:
  - Declare the data elements you want in the function block as output variables (VAR_OUTPUT)
  - Access data elements in the calling logic block by referencing the function block name and output variable: e.g., Value_I_Want := FB_instanceName.FB_ OUTvarName;

NOTE

You must reference the function block each processing interval to keep the function block running. For example, the following code will not allow a TON timer to execute correctly:

```
if a then

my_ton(IN:=tempBool, PT:=T#3S);

a:=FALSE;

end_if
```

If a is True the first run of the program, the TON function block instance, my_ ton, will start timing. Because a is set to FALSE, subsequent iterations of the program will not execute the function block reference. This results in the function block my_ton no longer executing and therefore never counting up to the value PT.

General Rules

See *Appendix B: IEC 61131-3 Programming Reference* for programming syntax and definitions. Following are some general rules for custom user logic development:

- Ensure values used in a logic statement are of the same type or of a type that can be converted to the same type. Mismatched value types will result in compilation errors. Refer to *Variables on page 680* for data type conversions.
- Take care when programming to not perform illegal logical operations. Illegal operations, such as corrupting memory by writing outside the bounds of an array or misusing pointers, can cause a memory exception error. Never-ending FOR and WHILE loops can exhaust the Watchdog timeout period. Such issues may cause undesirable effects, including replacing a halted project with the default project.
- Press the Save icon or **<Ctrl+S>** to save the project and compile the changes. The compiler will check the logic statements for errors and warnings and report these errors and warnings in the **Output** window.
- Always check the **Output** window for errors and warnings. Double-click on the error message to find the line with that error.
- Use parentheses to ensure maintenance of the correct order of calculation. For example, (4 + 2) * 6 is not the same as 4 + (2 * 6).
- When using global tags, use the appropriate value in the data tag structure. For example: Instantaneous analog input value: Relay1_DNP.AI_0001.instMag
- In CFC or LD functions such as ADD, SUB, etc., you can add input pins by dragging the pin tool from the ToolBox on to the function box.
- In CFC or LD functions, you can add or remove the EN and ENO pins by right-clicking on the function box.
- In CFC or LD functions, right-click on the function box or individual input and output pins to see available options.
- Observe operator precedence when writing logic (see *Table 9.1*).

Table 9.1 IEC 61131-3 Operation Order of Precedence (Highest to Lowest)[a]

Operation	Symbol
Parentheses	(expression)
Function evaluation Examples:	identifier (argument list) MAX(X, Y, Z), COUNT (A)
Exponentiation	EXPT
Negation	-
Complement	NOT
Multiply	*
Divide	/
Modulo	MOD
Add	+
Subtract	-

Operation	Symbol
Comparison	<, >, <=, >= LT, GT, LE, GE
Equality	=
Inequality	<>
Boolean AND	& AND
Boolean Exclusive OR	XOR
Boolean OR	OR

[a] Adapted from table at http://www.plcopen.org.

Variables

A variable is a tag with a unique name and a specific data type that allows you to reference data in the logic blocks as well as elsewhere in the SEL-5033.

- A program, function, or function block uses variables to store data values.
- A VAR is a definition statement you can use to create local variables. A local variable is only accessible to the logic block in which it is defined. You cannot pass or share these variables among logic blocks directly. The behavior of VARs differs according to the type of logic block in which they are created.
- Programs retain VAR values while a program is running.
- A function retains values as long as the function is executing. Once the function terminates (returns to the program), all VAR values are lost.
- Function blocks retain values as long as the program is running. Each time ACSELERATOR RTAC activates a function block, the function block uses all previous results in its VAR variables to perform logic and calculations, similarly to a program.

The following variable declarations are available in user logic:

- VAR_INPUT is a definition statement that functions or function blocks use to identify variables whose values will come from a logic block. Functions and function blocks must contain VAR_INPUT variable definitions before a calling logic block can pass data to a function or function block.
- VAR_OUTPUT is a definition statement that a function or function block uses to identify variables whose values the calling logic block can access. You can create any number of VAR_OUTPUTs to allow sharing of data. The function or function block will retain values of VAR_OUTPUTs for as long as the logic block exists. Note that VAR_OUTPUTs will have the same names for each function block instance, but they each have individual values and memory space. A calling logic block accesses function block data through a VAR_OUTPUT or VAR_IN_OUT variable.
- VAR_STAT is a definition statement that a function or function block uses to identify tags that maintain their value after the execution is complete. You can access static variables only within the scope in which you declare these variables. An example would be a counter within a function or function block that retains its value between calls to that function.

- VAR_TEMP is a definition statement available in programs and function blocks. Temp vars are only available within the program or function block in which they are defined and are automatically reinitialized every time the function block or program is called.
- VAR CONSTANT is a definition statement that can apply to a local or global list of identifiers. Constants have the same appearance as variables but are fixed values, set at declaration time, and are used in the same way as literals, such as 1, 9, or any other explicitly typed-out number. You can use constants in places throughout a program where fixed values would normally occur. The advantage of using constants is that you can change the values in the declaration section rather than searching through a program to find every instance of the value. You can also use a constant in the declaration of an array. A constant list has the following format:

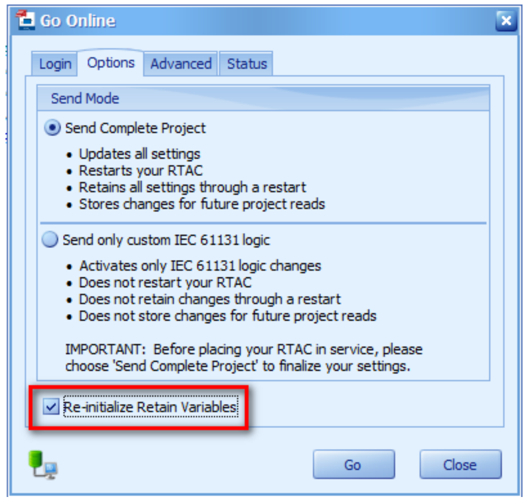
  VAR CONSTANT

  the list of constants in this format:

  <identifier> : <Type> := <value>;

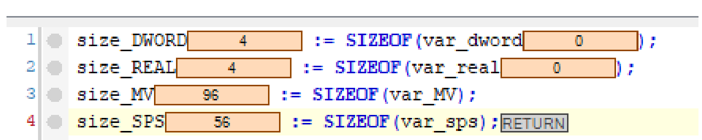
  END_VAR
- VAR_IN_OUT is a definition statement that a function block uses to identify tags that act as VAR_IN and VAR_OUT at the same time. The calling logic block and function block can read and write to VAR_IN_OUT variables.
- VAR_GLOBAL is a definition statement that global variable lists (GVLs) use to identify tags that you can access globally throughout the RTAC.
- VAR_GLOBAL RETAIN is a definition statement that you can use on GVLs to identify tags that retain their values after a reboot or project download. For example, click on **IEC-61131-3** under the **Insert** ribbon, then select **GVL**. You can rename the GVL here. Then, change the keywords VAR_GLOBAL to VAR_GLOBAL RETAIN and configure the global variables just as you would any variable list in the RTAC. The variables will retain their values through a power cycle, reboot, or after the project is resent to the RTAC.

Note that attempting to initialize RETAIN variables in the declaration is ignored. The values in existing memory will be used when a project is sent or the unit is power-cycled. To initialize the retained variables to a specific value, you must do this in a program. Variables in a retained list can be initialized to their default values by selecting the **Re-initialize Retain Variables** check box in the **Options** tab when sending a project to the RTAC.



An important characteristic of retained memory is that the order of the retained variable list is important to keeping values available through power cycles and project sends. If a retained list is created and then used in service and a configuration change requires adding variables to the retained list, the best practice is to add the new variables to the end of the retained list so as not to interfere with the present order. If the order of the retained list is changed, a recommended practice is to re-initialize the variables upon the project being sent, as shown in the previous figure. It is also recommended that only one retained list be used in a single RTAC project.

The memory allocated for retained variables is 28.5 KB. To calculate how many variables can be stored in retained memory, use **SIZEOF(<variable name>)** to see how many bytes of memory that variable consumes, and then multiply that by the variables declared in the retained list. The following figure shows several examples of determining how many bytes a variable uses.



Cross References

Cross references show all locations of a tag or variable within a project and indicate how the tag or variable is being used in each location. Examples of usage include declaration of the tag or variable, write operations, and read operations. You can find all locations of any tag or variable in your project by performing the following steps:

Step 1. Within a structured text program, function, or function block, right-click on the tag or variable you want to cross reference.

Step 2. Select **Browse > Browse Cross References**.

The software will find all instances of the tag or variable within the project and display the results in an output window. Double-click any of the references to navigate to the location of that tag or variable usage. By default, the scope of this search is limited to the active program. To find references elsewhere in the project, you must change the scope to include the entire project.

Creating User Logic

Write custom programs, functions, and function blocks by inserting an IEC 61131-3 user logic block. Write custom logic to perform tasks from simple analog scaling and data conversions to advanced closed loop control. The RTAC supports the following three IEC 61131-3 languages:

- Structured Text (ST): This is a simple, non-graphical programming language very similar to Pascal.
- Continuous Function Chart (CFC): This is a graphical function block diagram interface that is unconstrained to position in determining logic flow. You can reconfigure the execution order of each logic statement in ST, CFC, or LD.
- Ladder Logic Diagram (LD): This is the industry-standard relay ladder logic that uses logical ladder rungs to define logic flow.

Insert a Program

NOTE

You can perform column editing in Structured Text by pressing **<Ctrl+Alt+right-click>** and dragging over the column. The editor replicates anything you type in a highlighted column in every row of that column.

Step 1. Right-click on SEL_RTAC in the device tree, or click on the **Insert** ribbon and select an IEC 61131-3 user logic block. Recall that there must be at least one program, because functions and function blocks cannot operate independently.

Step 2. Identify the type of programming you want to do (ST, CFC, or LD).

Step 3. Create variables in the top window. Enter user logic in the bottom window.

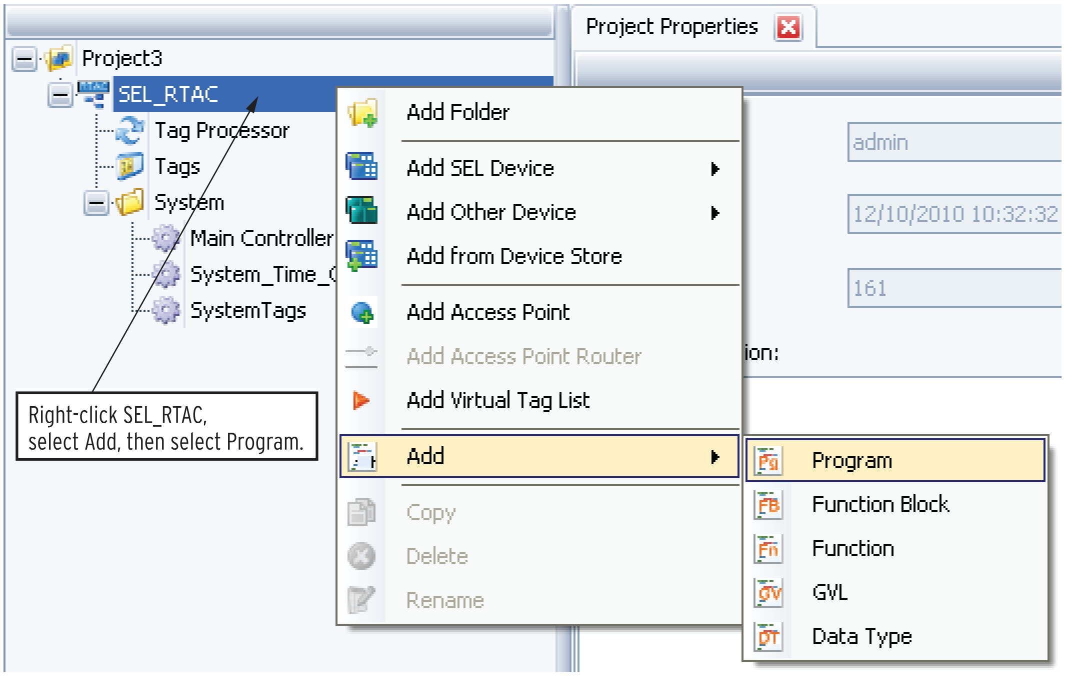


Figure 9.1 Inserting a Program

If you select ST, enter your structured text logic statements in the bottom window.

If you select CFC or LD, use the ToolBox on the right side of the screen to drag and drop logic elements onto the bottom window. The ToolBox contains graphical CFC or LD elements including input boxes, output boxes, and generic logic blocks. You can undock and move the ToolBox by clicking on the stickpin icon and then dragging the ToolBox by the header. You can re-dock the ToolBox by dragging your mouse to one of the dock points that appear when you are moving the ToolBox. You can also hide the ToolBox by clicking on the x in the ToolBox window. Un-hide the ToolBox by selecting Show ToolBox in the View ribbon menu.

In CFC, you can define a generic object box by dragging one onto the programming window, right-clicking on **???**, and selecting **Input Assistant**. Select the applicable function from the Input Assistant function list.

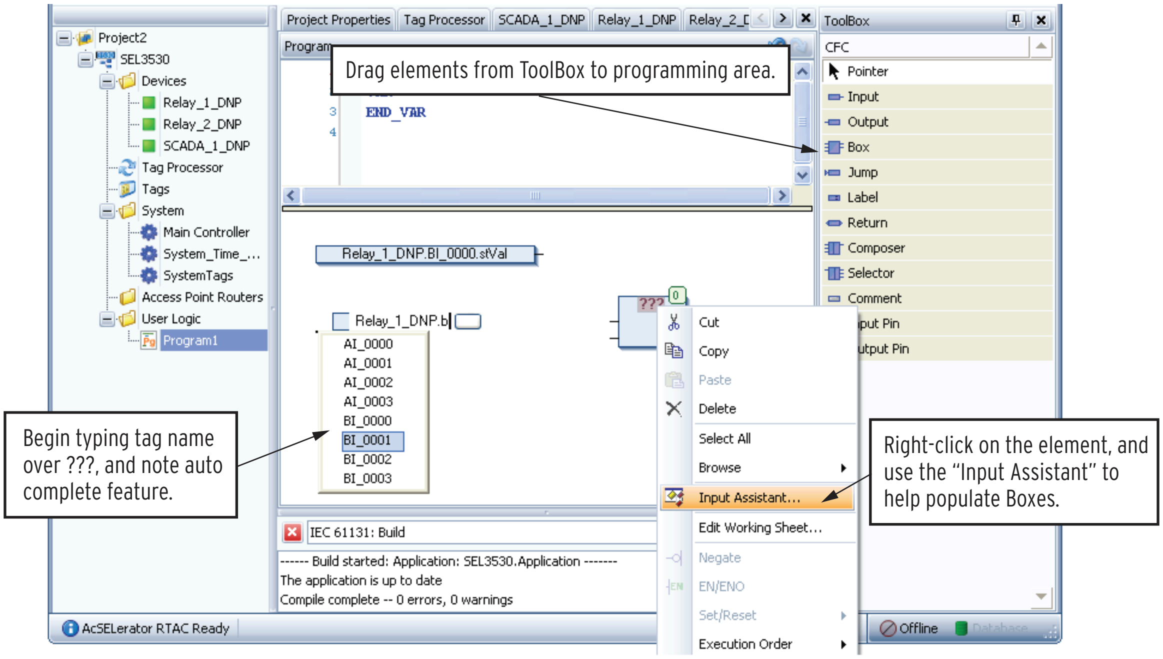


Figure 9.2 Input Assistant

Examples

The following programming examples demonstrate use of a program, a function, and a function block. In a Boolean test, we OR together binary input points that we configured previously in two client device connections (Relay_1_DNP and Relay_2_DNP). The four examples do essentially the same thing, but they look quite different. The examples demonstrate the different tools available as well as different programming styles that have essentially the same result.

1. The first program is in CFC. It compares two binary input values and sets Auxiliary LED 1 on the front panel to ON if either status becomes true and toggles it to OFF if both points are false. Notice on this program the use of negate to reverse the values of the inputs before they enter the AND gate.
2. The second program is in LD. Its function is exactly the same as the previous CFC program.
3. The third program is in ST. It uses an IF THEN ELSE statement to perform the same functionality on LED02.
4. The fourth program uses a function call as well as a function block to perform the same end result, toggling LEDs 3 and 4, respectively.

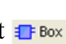
Example 9.1 Continuous Function Chart

Step 1. Select **Insert IEC 61131-3** from the ribbon.

Step 2. Select **Program**.

Step 3. In the pop-up window, give the program a unique name and select **CFC** from the drop-down menu.

Step 4. Click **Insert**.

Step 5. Drag one **Box** element  from the ToolBox on the right to the bottom window worksheet.

Step 6. Drag two **Input** elements and place these elements to the left of the Box element.

Step 7. Drag one **Output** element and place this element to the right of the Box element.

Step 8. Change the **???** in the Box element to OR. Either type in OR, or use the Input Assistant.

Step 9. Change the **???** in the **Input** and **Output** elements as the following example shows. Note that the software helps you complete point names.

Step 10. Click on the pin on the **Input** element and, while depressing the mouse button, drag the line over to the input pin of the Box element. This action connects the logic blocks with lines.

Step 11. As the example shows, repeat *Step 10* for the second **Input** box.

Step 12. In this example, there are two similar sets of logic. To save time, copy and paste the logic blocks to create duplicates. To select the elements of the program, do one of the following:

a. Right-click on the blank part of the page, and then choose **Select All**.

b. While holding down the **<Ctrl>** key, click the edge of each element box with the mouse.

c. Click on the upper left-hand corner of the programming screen. With the left mouse button depressed, drag a selection box over all the elements to mark them all as selected.

Step 13. Press **<Ctrl+C>** to copy, **<Ctrl+V>** to paste (or use the **Copy** and **Paste** menu buttons).

NOTE
In the lower right-hand corner of the program worksheet area is a zoom tool. Use this to increase/decrease the visible workspace in the worksheet.

Step 14. While holding down the **<Ctrl>** key, click on the edge of one of the duplicate boxes and drag the duplicated boxes below the original set of boxes. All duplicate boxes should move together and maintain their spatial order. If you do not hold down the **<Ctrl>** key, the boxes may become deselected and may fail to move together. If this happens, drag the boxes separately or reselect each box while holding down the **<Ctrl>** key, and attempt to move them again as a group.

Step 15. Change the names/logic in the duplicate boxes to match *Figure 9.3*. You can copy and paste any item in a CFC in a similar manner.

Continuous Function Chart Solve Order

Figure 9.3 shows *Example 9.1* and also small green squares containing numbers in them. Because this program is written in CFC, there are no hard spatial relationships dictating the solve order of the logic. The numbers in the green squares indicate the default solve order. Pay special attention to solve order when testing your programs to ensure that the results are what you expect.

Testing your program may reveal that it is necessary to change the solve order of your program. For example, your program may be comparing results of a multi-step calculation before that calculation completes. This situation could cause unexpected and unwanted results. Keep in mind that watching the values change is not a good indication that the solve order is correct. Changes could occur and affect logic much faster than you can observe these changes on the screen.

Perform the following steps to change the default solve order.

Step 1. Right-click on the **Logic** box to change its solve order.

Step 2. From the pop-up menu, select **Execution Order**.

Step 3. Select one of the options to change the execution order of that box.

Step 4. In the following example, the execution order appears backwards. Perform the following steps to correct the order:

a. Right-click any block, and then select **Execution Order**.

b. Select **Order** by data flow.

The numbers in the green boxes will change to 0, 1, 2, 3 in order.

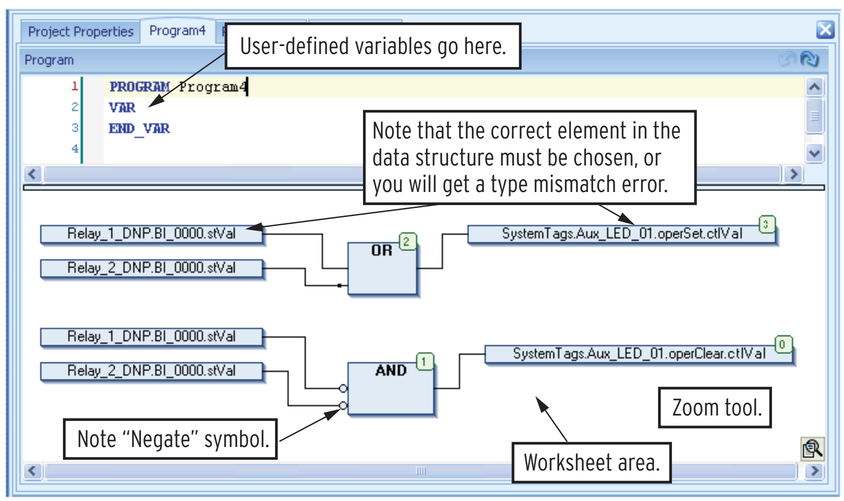


Figure 9.3 Example CFC Program

Example 9.2 Ladder Logic Diagram (LD)

Step 1. Select **Insert IEC 61131-3** from the ribbon.

Step 2. Select **Program**.

Step 3. In the pop-up window, give the program a unique name and select program language **LD** from the drop-down menu.

Step 4. Click **Insert**.

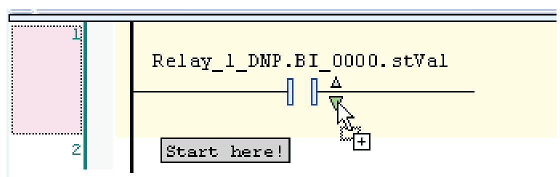
Step 5. By default, the ToolBox covers the left side of the workspace. Move the ToolBox by first clicking on the tack icon, then grab the ToolBox header and move it to the right of the screen.

Step 6. In the ToolBox, select the **Ladder elements** tab.

Step 7. Click on **Contact**, and drag it onto the workspace. Place it on the **Start here!** icon, and release the mouse button.

Step 8. Replace the **???** above the newly placed contact with Relay_1_DNP.BI_ 0000.stVal.

Step 9. In the ToolBox, click on **Parallel contact** and drag it onto the workspace. Place it on the down arrow that appears on the ladder (also refer to *Figure 9.4*).



Step 10. In the ToolBox, click on **Coil** and drag it onto the workspace. Place it on the **Add output or jump here!** icon to complete the circuit.

Step 11. Replace the **???** above the contacts and coil with the tag names shown in *Figure 9.4*. This ladder is a logical OR of the two status tags, with the output controlling the coil at the end of the ladder.

Step 12. In the ToolBox, click on **Network** and drag it onto the workspace. Place it on the down icon to insert another network ladder.

Step 13. Drag a **Contact** from the toolbox onto the new network.

Step 14. Drag another **Contact** from the toolbox and place it on the icon beside the contact from *Step 13*.

Step 15. Drag a **Coil**, and place it at the end of that ladder.

Step 16. Replace the **???** above the contacts and coil with the tag names shown in *Figure 9.4*. This ladder is a logical AND of the two status tags with the output controlling the coil at the end of the ladder.

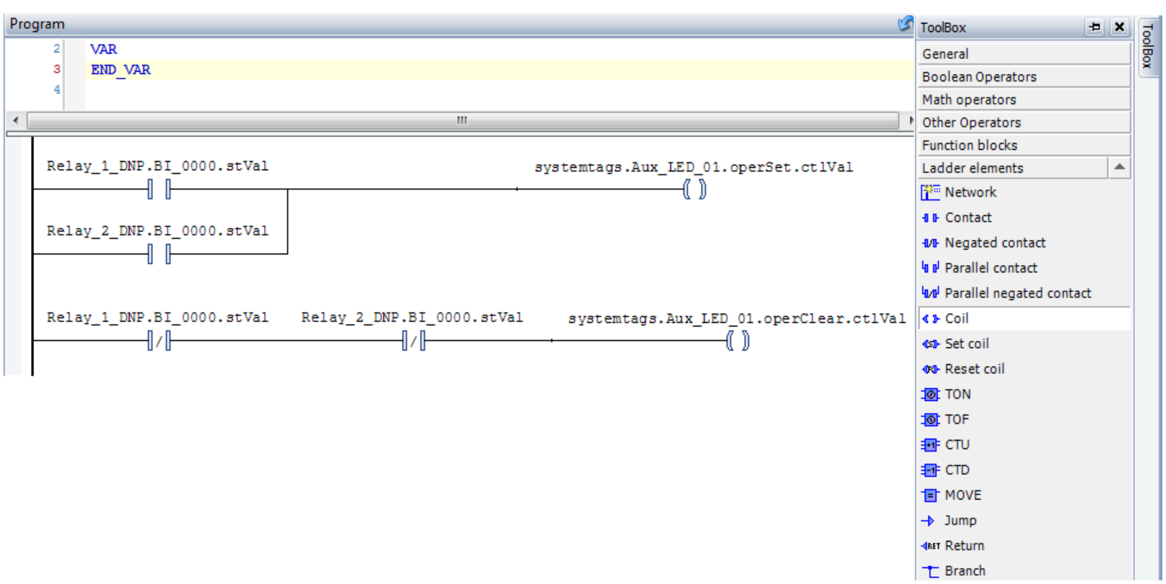


Figure 9.4 Example LD Program

Example 9.3 Structured Text

Step 1. Select **Insert IEC 61131-3** from the ribbon.

Step 2. Select **Program**.

Step 3. In the pop-up window, give the program a unique name and select **ST** from the drop-down menu.

Step 4. Click **Insert**.

In this example, there are no user-defined variables. If there were, these variables would go in the user-defined variable section. All variables in use here are system tags.

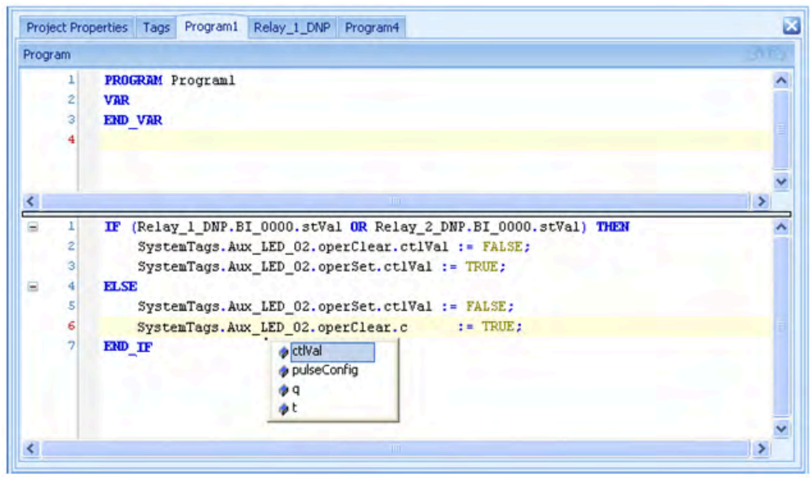


Figure 9.5 Example ST Program

Example 9.4 Structured Text Program Using a Function and a Function Block

This example is actually two examples in one. The first is ST using a function call. The second is identical to the first, except that it uses a function block. This example demonstrates that programs can use both functions and function blocks, and it shows the differences between functions and function blocks.

Step 1. Insert a program, select **ST**, and provide a unique name for the program. We named it PGM_Using_Function_and_FunctionBlock in the example.

Step 2. Insert a function, select **ST**, and give the function a unique name. We named it OR_Pnts_FN in the example.

Step 3. Insert a function block, select **ST**, and give the function block a unique name. We named it OR_Two_Points_FB in the example.

Step 4. Create an instance of the function block in the program, by creating a unique name and defining the instance as type OR_Two_Points_FB.

You can define any number of uniquely named instances as that same type in any number of separate programs. In this example, the instance is named my_OR_FB.

Note that we can activate the function block, or cause it to run, by referring to its name and assigning values to its input variables.

Step 5. The calling program can access output variables from the function block at any time after activation of the function block. Access these variables by using the following format: function block instance name.output variable name. This is illustrated in the following function block variable accessed by the sample program:

my_OR_FB.LED_off;

Step 6. You can access the results of the function after an assignment statement calls the function. An assignment statement calls the function and places the resulting value in a variable of the same type as the function. This is illustrated in the following function call invoked by the sample program:

LED_State := OR_2_Pnts_FN(DI_Value1, DI_Value2);

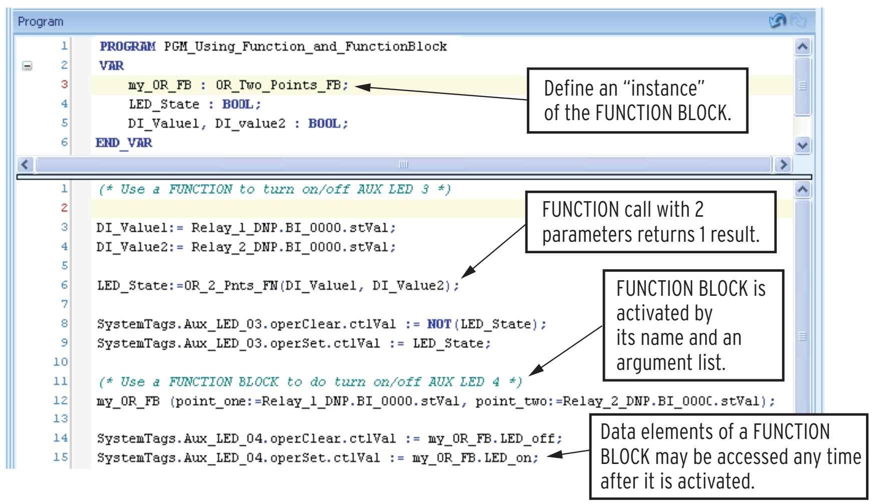


Figure 9.6 Using a Function and Function Block in a Program

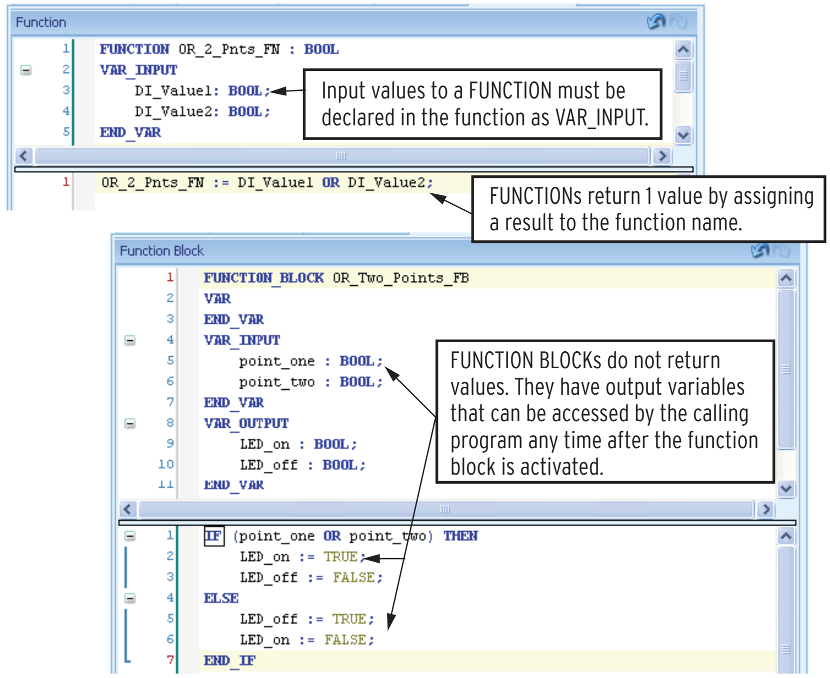


Figure 9.7 Example Function and Function Block

SECTION 10

Extensions

Dynamic Disturbance Recorder

Overview

The Dynamic Disturbance Recorder extension is designed to simplify data archiving from the IEC 61131 logic engine to the sequestered file system of an SEL Real-Time Automation Controller (RTAC). Generated files are then managed based upon parameters supplied by the user. The maximum rate at which a Dynamic Disturbance Recorder can record data is directly related to the main task cycle time configured in an ACSELERATOR RTAC® SEL-5033 Software project. Data cannot be recorded at a rate faster than the main task cycle, and the effective data recording rate is a multiple of the main task cycle. See *Section 1: Getting Started* for more details on configuring the task cycle.

Whether the Dynamic Disturbance Recorder is enabled is determined by the MOT string (or part number) stored in the RTAC memory. When you order a new RTAC including the Dynamic Disturbance Recorder library extension, the MOT is configured correctly at the factory. If you want to add this feature to an existing RTAC, contact your SEL sales representative to obtain an easily installed upgrade package.

Settings Tab

The **Settings** tab contains all configurable items for data recording and file management of generated log files. See the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description. Type any applicable comments in the blank column on the right side.

Monitored Data Tab

Data tags that are monitored by a Dynamic Disturbance Recorder are added to the **Monitored Data** tab.

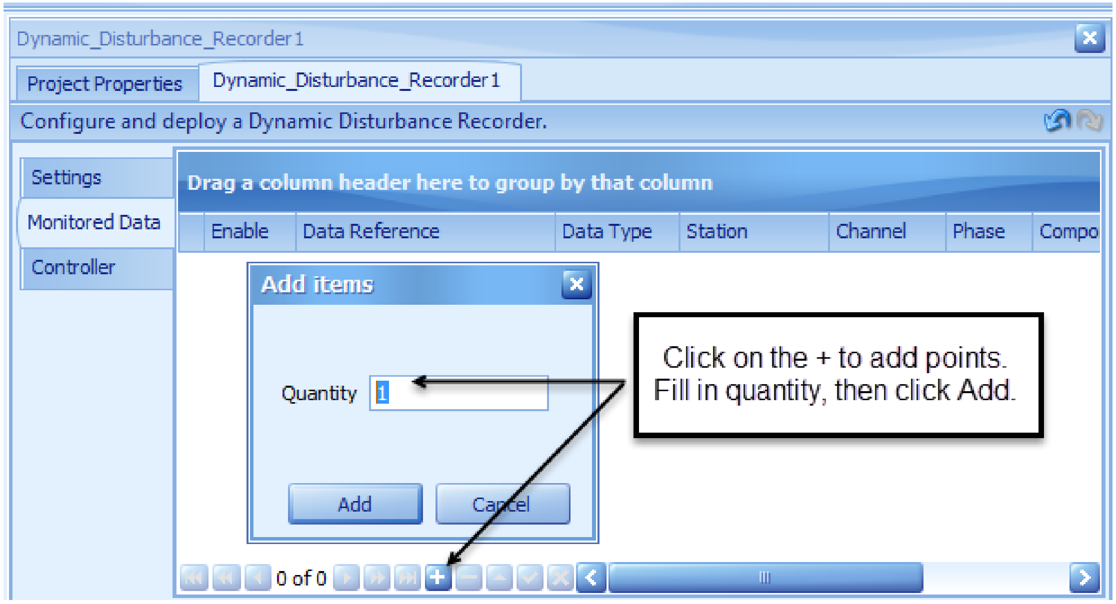


Figure 10.1 Add Data Tags for Monitoring and Recording

Monitored Tag Configuration Parameters

Each monitored tag type has parameters that control whether the tag is recorded and what information is associated with recorded tag data.

Table 10.1 Monitored Tag Type

Column Name	Description
Enable	The point is monitored by the recorder when True.
Data Reference	Source tag that is monitored by the recorder.
Data Type	The auto-detected data type of the monitored tag.
Station	User designation for data grouping.
Channel	User designation for data source.
Phase	Phase designation for the data. Only available for COMTRADE file formats.
Component	Component designation for the data. Only available for COMTRADE file formats.
Unit	The engineering units assigned to the quantity. Only available for analog quantities in COMTRADE file formats.
Default Value	Default state of the Boolean status.
Comment	Optional user-entered comment field.

Controller Tab

Use the function blocks pins to view and modify the state of a Dynamic Disturbance Recorder. See *Table 10.2* for the settings descriptions.

Table 10.2 Dynamic Disturbance Recorder POU Pin Settings

Pin Name	Pin Type	Description	Default
EN	Input: BOOL	The EN input enables or disables this specific function block instance. Other inputs have no effect while EN is False.	True
ActiveFileName	Output: String(255)	Name of the file to which logs are actively being written.	""
ConsumedDirectorySize	Output: ULINT	Size of all files in the monitored directory.	0
MonitoredPoints	Output: DINT	The total number of tags monitored by the recorder. Applicable to CSV formats.	0
AnalogPoints	Output: DINT	The number of analog tags monitored by the recorder. Applicable to COMTRADE formats.	0
DigitalPoints	Output: DINT	The number of digital tags monitored by the recorder. Applicable to COMTRADE formats.	0
Error	Output: BOOL	A potential error condition was detected.	False
ErrorMessage	Output: String(255)	Message describing the source of the Error flag.	"

Supported File Formats

Three file formats are available which fall into two general categories: discrete event and time-aligned log entries. The three formats are listed and described in this section.

Discrete Event Log Entries

SOE CSV

An ASCII CSV file in which each log represents the status value and the date and time value of a tag. Additional information in the log includes the time code (offset from UTC), station name, and channel name. This format is categorized as a discrete event log, i.e., a log is generated when the status value of the tag changes state. Supported tag data types include BCR, CMV, DPS, INS, MV, SPS, and STR. A sample file name and record are shown in *Figure 10.2* and *Figure 10.3*, respectively. The sample file shown in *Figure 10.3* is configured to monitor two SPS tags (representing breaker positions).

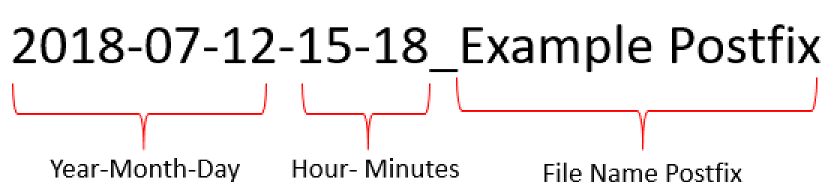


Figure 10.2 Example SOE and Time-Aligned CSV File Name

The time stamp of the file reflects the time of file creation, which may not be the time of the first log entry. The 1 and 0 in the sample record represent a TRUE and FALSE, respectively.

```
Date,Time,Time Code,Station,Device,Value
07/12/2018,15:16:05.441,-6,SubstationXYZ,Breaker1,1
07/12/2018,15:16:05.481,-6,SubstationXYZ,Breaker2,0
```

Time-Aligned Log Entries

The following two file formats are categorized as time-aligned logs in which all tags represented in a log entry are associated with a single time stamp. This category of files has three triggering mechanisms (detailed in *Triggering Conditions on page 543*).

Time-Aligned CSV

An ASCII CSV file in which each log entry is a single line with all included tag values with a single time value. The order in which the data points are listed in the **Monitored Data** tab determines the position of the data in the log. A null entry, representing missing data for appropriate triggers, is represented as a blank entry (no characters) delimited by two commas. Supported tag data types include BCR, CMV, DPS, INS, MV, SPS, and STR. The created file name follows the format shown in *Figure 10.2*. A sample log is shown in *Figure 10.4*. This recorder is configured to monitor one of each of the following data types: CMV, INS, and SPS. Note in *Figure 10.3* that a phasor quantity, represented by a CMV data type, shows two entries: an entry for the magnitude and the angle of the phasor.

```
Timestamp,StationXYZ:PhA:Magnitude,StationXYZ:PhA:Angle,StationXYZ:BattVoltage,
StationXYZ:BattVoltage,StationXYZ:51A
2018/07/12 15:16:06.684,100.026,120.0,120,1
2018/07/12 15:16:07.684,100.026,120.0,120,1
```

COMTRADE Float32

Two COMTRADE files are generated that adhere to IEC 60255-24:2013 and IEEE C37.111-2013. The first, a CFG file, is an ASCII file that includes information pertinent to the interpretation of a corresponding DAT file. The maximum size of the CFG file cannot exceed 1 MB. The size of the CFG file is related to the number of tags and the parameters given for those tags in the **Monitored Data** tab.

The DAT file contains the log entries and is a FLOAT32 binary file. Supported tag data types include CMV, MV, INS, and SPS. The order of the points in the CFG and DAT file is the order in which the supported tag data types are listed above, followed by the order the points are listed in the **Monitored Data** tab. For the COMTRADE format, missing data are filled with the hex value 0xFF7FFFFF in the binary DAT file for analog points. For binary points, missing data are represented as a zero. See IEC 60255-24:2013 and IEEE C37.111-2013 for more details pertaining to the COMTRADE file format. The generated files are named following the IEEE Standard for Common Format for Naming Time Sequence Data Files (COMNAME) C37.232-2011A. A sample file name and CFG file is shown in *Figure 10.3* and the screen that follows, respectively. The recorder is configured to monitor one of each of the following data types: CMV, INS, and SPS.

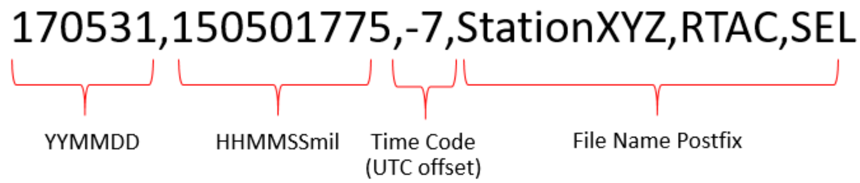


Figure 10.3 Example COMTRADE Float32 File Name

```
StationXYZ,RTAC,SEL,RTAC_Archive,2013
4,3A,1D
1,SubstationXYZ:Bay1_m,Am,CT,kA,1,0,0,-3.4028265E38,3.4028235E38,1,1,P
2,SubstationXYZ:Bay1_a,Aa,CT,kA,1,0,0,-3.4028265E38,3.4028235E38,1,1,P
3,SubstationXYZ:DC Supply,,Battery,V,1,0,0,-3.4028235E38,3.4028235E38,1,1,P
1,Substation:51 Element,A,SEL_421,F
60
0
0,173
12/07/2018,15:16:06.684394 12/07/2018,15:16:06.684394 float32 1000
-6,-6
```

Triggering Conditions

For both the Time-Aligned CSV and COMTRADE Float32 formats, there are three types of triggers that generate logs. Only one trigger type can be configured per recorder. These triggers are listed as follows:

➤ Time Change

This trigger method monitors the tags listed in the **Monitored Data** tab for changes in the dateTime_t data structure. For more information on tag data structures, see *Section B: IEC 61131-3 Programming Reference*. A detected change in any dateTime_t data structure within a monitored tag creates a log with a time value. Each log entry contains all the points for which a time change was detected. By default, the first dateTime_t data structure for which a time change is detected is the time reference of the log entry. A user can specify a timeStamp_t data structure that is used as the time reference to better control when log entries are created and what time stamp is associated with the log. This is the optional setting, Time Reference, in the **Setting** tab. Any detected change in the dateTime_t data structure of the Time Reference setting will generate a log, and the time value associated with that log will be derived from the dateTime_t data structure of the Time Reference setting. Time Variance is an optional setting that allows users to set a window around the time reference. If a time change is detected during the scan, but the time values of those data points do not fall within the time window (defined by the setting Time Variance) relative to time reference, it will not be included in the log. Points that do not change or are outside the time window will have a null entry. This entry maintains the columnar position in the log entry as determined by the file format.

- Periodic

  This trigger method periodically samples all points at the specified interval set by the Logging Interval in the **Settings** tab, regardless of whether the time value or the data status value of the monitored tags change. The time value for the log entry is the system time of the RTAC at the time of the periodic interval. No null entries are present because all data are sampled. This is a snapshot of the tag status values as they are in the RTAC logic engine at the time of the periodic interval.

- Event

  This trigger method monitors a Boolean value set by the Trigger Signal in the **Settings** tab. When the Boolean trigger condition is evaluated as TRUE, a file is generated that contains a configurable number of pre-trigger and post-trigger log entries in addition to a log entry for the time the trigger signal asserted. New log entries are created at each main task cycle. The time period encompassed by the log is equal to the main task cycle time multiplied by the number of cycles set by the settings Pre-trigger task cycles and Post-trigger cycles plus a trigger sample for the asserted Trigger Signal. If a trigger condition is detected before the minimum configured Pre-trigger task cycles are met, a log will be created that contains the available pre-trigger log entries plus the trigger and the post-trigger log entries. If a trigger condition is detected before a previous trigger event is processed, it will be ignored. Like the Periodic trigger, a value is populated for each monitored tag, regardless of change in data status or time stamp. Triggered file records cannot exceed 1 MB in total file size.

File Management

A Dynamic Disturbance Recorder automatically manages the rate at which files are generated and deleted. These behaviors are adjustable via three settings in the **Settings** tab: Max File Size, Max Folder Size, and Maximum Number of Days.

The rate at which files are generated is controlled by the Max File Size. This setting enables a compromise between the number of files generated and the maximum usable file size that analytic applications can successfully interpret. This will also determine how easy it is to move or transfer files. For optimal file system performance, each recorder must be configured such that the rate of file generation does not exceed the ability of the file system to process the created files. When configuring your recorder, ensure that files are not created faster than once every 100 processing cycles as a general rule. To verify if the recorder is creating files at an unsustainable rate, monitor the ActiveFileName output POU Pin. If this name is changing faster than 100 processing cycles, the recorder may encounter issues when trying to create or delete files.

Generated files persist in the file system until one of two conditions are met: (1) the files age out of Maximum Number of Days for file retention or (2) the configured Max Directory Size is exceeded. This setting is used to set an upper bounds on the amount of storage a recorder is allowed to consume in the sequestered file system. It also provides a means to provision storage between multiple recorders such that the overall available storage is not exceeded. Ensure an appropriate buffer so that maximum system storage is not exceeded.

The mechanisms for managing files differ between CSV and COMTRADE file format recorders. The CSV file format recorders assess the Maximum Number of Days threshold against current RTAC system time and deletes files that exceed the threshold. If the Max Directory Size is exceeded, files are deleted one at a time until the total directory size is less than the Max Directory Size. COMTRADE file format recorders assess the Maximum Number of Days threshold against the latest file time stamp (as shown in the file name) in the directory. Files that exceed that threshold are deleted. If the Max Directory Size is exceeded, files are deleted in units of one day until the total directory size is less than the Max Directory Size.

Recording Triggers

Overview

The Recording Triggers Extension is designed to configure simple analog and digital trigger event conditions for Axion Recording Groups and other event recording devices. When a trigger event condition occurs, the Recording Trigger Extension asserts its output and sends this output signal to all the associated recording devices to initiate oscillography data capture. *Figure 10.4* shows the interaction between the Recording Triggers Extension and the Axion Recording Groups. See *Configuring Axion Recording Groups on page 221* for more information on Axion Recording Groups.

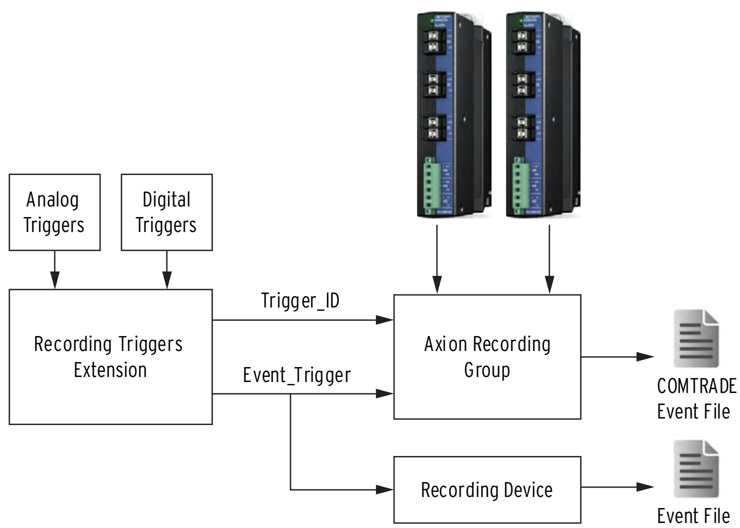


Figure 10.4 Interaction Between Recording Triggers Extension and Recording Devices

Use the **Insert** ribbon selection, as shown in *Figure 10.5*, to include an instance of the Recording Triggers extension into your project.

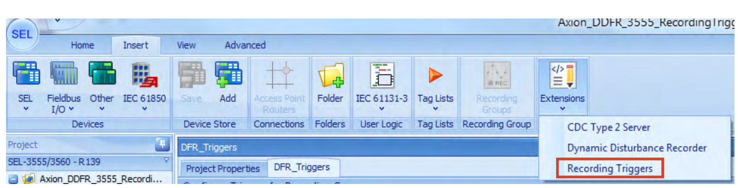


Figure 10.5 Insert a Recording Triggers Extension

Once you add a Recording Triggers Extension to a project, you will see the four tabs shown in *Figure 10.6*.

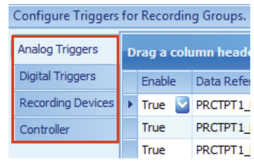


Figure 10.6 Recording Triggers Extension Tabs

The following sections describe the four tabs of the Recording Triggers Extension.

Analog Triggers Tab

The **Analog Triggers** tab is used to configure triggers on analog input signals. Data tags that are monitored for analog triggers are added to the **Analog Triggers** tab. Three analog triggers can be configured on each analog data type added to the **Analog Triggers** tab:

- High threshold
- Low threshold
- Rate-of-change

High-Threshold Trigger

The trigger output asserts if the value of the monitored analog signal raises above a predefined threshold value. High-threshold triggers have two operating modes: rising-edge and level.

- Rising-Edge Mode

  When the observed analog value moves above the threshold, the Recording Trigger Extension sends a one-processing-interval pulse to the Event_Trigger input of the associated Axion Recording Groups. Therefore, the associated Axion Recording Groups will produce an event file of minimum oscillography record length.

- Level Mode

  In this mode, the trigger output remains asserted for as long as the trigger condition exists. When the observed analog value moves above the threshold, the Recording Trigger Extension asserts the Event_Trigger input of the associated Axion Recording Group. The trigger condition deasserts when the observed analog value moves below a value determined by the trigger threshold value and the hysteresis value. This results in an extended-length oscillography recording.

Figure 10.7 illustrates the operation of the rising-edge and level modes of the high-threshold trigger.

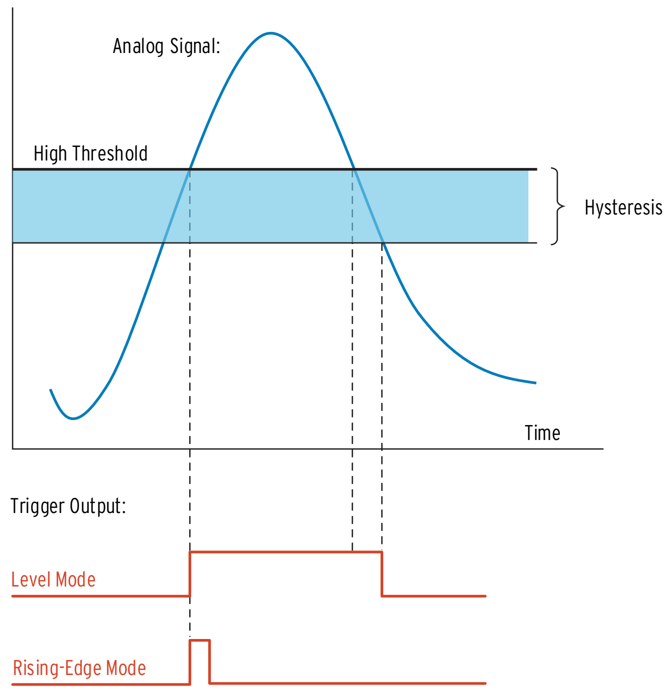


Figure 10.7 High-Threshold Trigger: Rising-Edge and Level Output Modes

Low-Threshold Trigger

The output trigger asserts if the value of the monitored analog signal drops below a predefined threshold value. Low-threshold triggers have two operating modes: rising-edge and level.

- Rising-Edge Mode

  When the observed analog value moves below the threshold, the Recording Trigger Extension sends a one-processing-interval pulse to the Event_Trigger input of the associated Axion Recording Group. Therefore, the Recording Group will produce an event file of minimum oscillography record length.

- Level Mode

  In this mode, the trigger output remains asserted for as long as the trigger condition exists. When the observed analog value moves below the threshold, the Recording Trigger Extension asserts the Event_Trigger input of the associated Axion Recording Groups. The trigger condition deasserts when the observed analog value moves above a value determined by the trigger threshold value and the hysteresis value. This results in an extended-length oscillography recording.

Figure 10.8 illustrates the operation of the rising-edge and level modes of the high-threshold trigger.

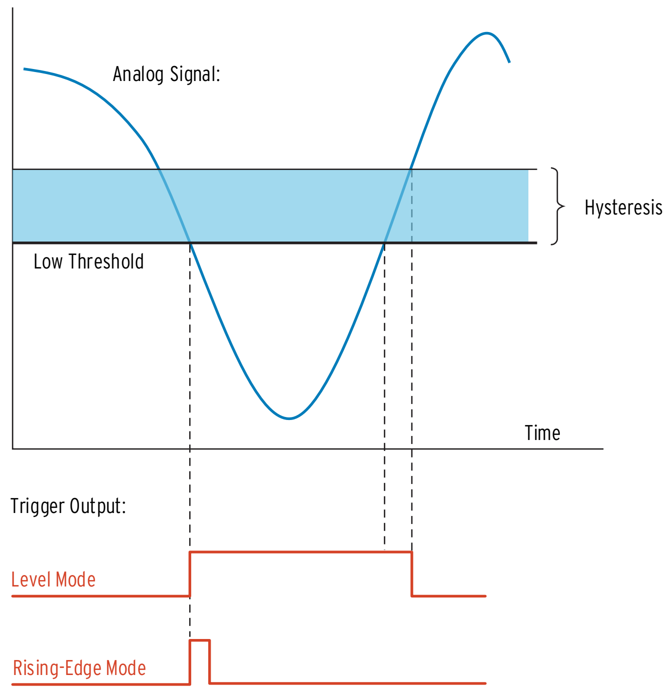


Figure 10.8 Low-Threshold Trigger: Rising-Edge and Level Output Modes

Rate-of-Change Trigger

The output trigger asserts if the value of the monitored analog signal rate-of-change exceeds a predefined rate over a predefined time period. Rate-of-change triggers have two operating modes: rising-edge and level.

- Rising-Edge Mode

  When a rate-of-change event condition is detected in the observed analog value, the Recording Trigger Extension sends a one-processing-interval pulse to the Event_Trigger input of the associated Axion Recording Groups. Therefore, the Recording Group will produce an event file of minimum oscillography record length.

- Level Mode

  In this mode, the trigger output remains asserted for as long as the rate-of-change condition exists. When a rate-of-change event condition is detected in the observed analog value, the Recording Trigger Extension asserts the Event_Trigger input of the associated Axion Recording Groups. The trigger condition deasserts when the rate-of-change of the analog signal returns to a value below the defined threshold.

Columns

Table 10.3 Analog Triggers Tab Column Definitions

Column	Description
Enable	The analog tag is monitored by the Recording Triggers Extension when True.
Data Reference	Analog tag that is monitored by the Recording Triggers Extension. Allowed types are REAL, MV, SINT, USINT, INT, UINT, DINT, and INS.
Data Type	The auto-detected data type of the monitored analog tag.

Column	Description
Channel ID	String to identify the channel (e.g., VA, RealPower). The specific trigger ID string is appended to the Channel ID string to identify the Recording Group output event file.
Threshold Units	Defines the units in which the trigger thresholds and hysteresis values are entered. There are three possible settings for this value: ➤ Engineering Units: The trigger thresholds and hysteresis values are entered as the actual status value of the analog channel. ➤ Percentage-of-Nominal: The trigger thresholds and hysteresis values are entered as percentage of a nominal value. ➤ Per-Unit: The trigger thresholds and hysteresis values are entered in per-unit values.
Nominal/PU Base	The analog tag nominal value or per-unit base. This value is required if the Threshold Units column is set to Percentage-of-Nominal or Per-Unit.
Enable High Trigger	If True, the analog tag is monitored for high-threshold event triggers.
High Threshold	Minimum value of the analog tag required to assert the trigger. This value must be entered in the units selected in the Threshold Units column.
High Trigger Type	Rising-edge mode or Level mode, as described in the High-Threshold Trigger section.
High Trigger Hysteresis	The hysteresis value. This value is subtracted from the High Threshold value to define the value at which the trigger deasserts when in Level mode. This value must be entered in the units selected in the Threshold Units column and is ignored in Rising-edge mode.
High Trigger ID	String to identify the high-threshold trigger (e.g., 59element). This string is appended to the Channel ID string value to identify the Recording Group output event file.
High Trigger Log	There are three settings for this column: ➤ SOE: An RTAC Sequence-of-Events (SOE) entry is created if a high-threshold condition is detected on the analog tag. ➤ SOE + Alarm: In addition to creating an SOE entry, the high-threshold trigger event is displayed in the Alarm Summary of the RTAC web interface. ➤ Disable: High-threshold trigger event conditions on the analog tag are not logged.
Enable Low Trigger	If True, the analog tag is monitored for low-threshold event triggers.
Low Threshold	Maximum value of the analog tag required to assert the trigger. This value must be entered in the units selected in the Threshold Units column.
Low Trigger Type	Rising-edge mode or Level mode, as described in the Low-Threshold trigger section.
Low Trigger Hysteresis	The hysteresis value. This value is added to the Low Threshold value to define the value at which the trigger deasserts when in Level mode. This value must be entered in the units selected in the Threshold Units column and is ignored in Rising-edge mode.
Low Trigger ID	String to identify the low-threshold trigger (e.g., 27element). This string is appended to the Channel ID string value to identify the Recording Group output event file.
Low Trigger Log	There are three settings for this column: ➤ SOE: An RTAC Sequence-of-Events (SOE) entry is created if a low-threshold condition is detected on the analog tag. ➤ SOE + Alarm: In addition to creating an SOE entry, the low-threshold trigger event is displayed in the Alarm Summary of the RTAC web interface. ➤ Disable: Low-threshold trigger event conditions on the analog tag are not logged.
Enable Rate-of-Change	If True, the analog tag is monitored for rate-of-change event triggers.
Rate-of-Change	Minimum change of the analog input tag over a user-specified time window (ROC Time column) required to assert the trigger. This value must be entered in the units selected in the Threshold Units column.
ROC Time (ms)	Time period, in milliseconds, at which the analog tag value is sampled to detect a rate-of-change condition. The effective value of the time window is a multiple of the main task cycle time.
ROC Trigger Type	Rising-edge mode or Level mode, as described in the Rate-of-Change Trigger section.

Column	Description
ROC Trigger ID	String to identify the rate-of-change trigger. This string is appended to the Channel ID string value to identify the Recording Group output event file.
ROC Log	There are three settings for this column: ➤ SOE: an RTAC Sequence-of-Events (SOE) entry is created if a rate-of-change condition is detected on the analog tag. ➤ SOE + Alarm: In addition to creating an SOE entry, the rate-of-change condition event is displayed in the Alarm Summary of the RTAC web interface. ➤ Disable: Rate-of-change condition events on the analog tag are not logged.

Digital Triggers Tab

The **Digital Triggers** tab is used to configure triggers on digital signals. Boolean data tags that are monitored for digital triggers are added to the **Digital Triggers** tab. Each digital trigger has four possible operating modes:

- Rising-Edge Mode

  The output trigger asserts for one RTAC processing interval when the monitored digital tag changes from False to True.

- Falling-Edge Mode

  The output trigger asserts for one RTAC processing interval when the monitored digital tag changes from True to False.

- State-Change Mode

  The output trigger asserts for one RTAC processing interval when a change of state is detected in the monitored digital tag.

- Level Mode

  The output triggers remain asserted as long as the monitored digital signal is asserted.

Table 10.4 Digital Triggers Tab Column Definitions

Column	Description
Enable	The digital tag is monitored by the Recording Triggers Extension when True.
Data Reference	Digital tag that is monitored by the Recording Triggers Extension. Allowed types are BOOL and SPS.
Data Type	The auto-detected data type of the monitored analog tag.
Trigger Type	Rising-edge, Falling-edge, State-change, or Level.
Trigger ID	String to identify the digital trigger. This string is used to identify the Recording Group output event file.
Trigger Log	There are three settings for this column: ➤ SOE: An RTAC Sequence-of-Events (SOE) entry is created if a digital trigger condition is detected on the analog tag. ➤ SOE + Alarm: In addition to creating an SOE entry, digital trigger condition event is displayed in the Alarm Summary of the RTAC web interface. ➤ Disable: Digital trigger condition events on the analog tag are not logged.

Recording Devices Tab

The **Recording Devices** tab is used to set the recording devices that are triggered when a trigger event condition, analog or digital, is detected. Two types of tags can be entered in this tab:

➤ Axion Recording Group POU Type

The Axion Recording Groups added to the **RecordingDevices** tab are automatically triggered by the Recording Triggers Extensions if a trigger event condition is detected. Additionally, the Recording Triggers Extensions sends the proper trigger ID to the Recording Group, as shown in *Figure 10.7*. The Axion Recording Group uses the trigger ID to identify the output event file.

➤ Boolean Type

Boolean tags added to the **Recording Devices** tab are automatically asserted by the Recording Triggers Extensions if a trigger event condition is detected. This allows to trigger a recording device different than a Recording Group, for example, the oscillography recording of an SEL-2245 Analog Input Module.

In the example shown in *Figure 10.9*, two tags have been associated to the Recording Devices tab: a Recording Group POU (named DFR) and the Event_Trigger input pin of an SEL-2245 Analog Input Module. If a trigger event occurs, the Recording Triggers extension will send the trigger signal to both recording devices to start data capture.

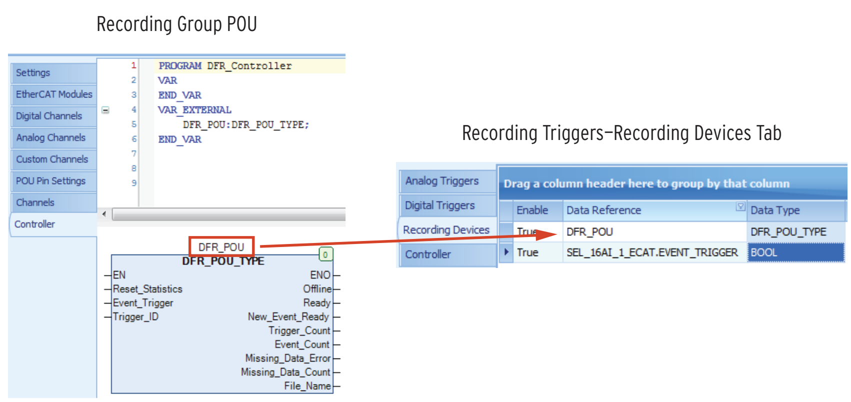


Figure 10.9 Assigning an Axion Recording Group to a Recording Triggers Extension

Controller Tab

The **Controller** tab contains the Recording Triggers Extension Program Organization Unit (POU). The Recording Triggers POU includes the following two outputs:

- Event_Trigger_Out

  This is the trigger output signal. This output becomes True when a trigger event condition is detected. This is the same signal that is sent to the devices in the **Recording Devices** tab to start data capture.

- Trigger_ID_Out

  If a rising edge is detected in the Event_Trigger_Out output, the Trigger_ID_Out pin outputs the trigger ID string that identifies the trigger type. This output is sent to the Trigger_ID input pin of the Recording Groups associated to the **Recording Devices** tab when a trigger occurs.

Figure 10.10 shows the Recording Triggers Extension POU block.

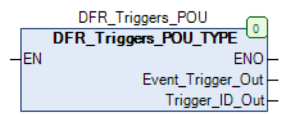


Figure 10.10 Recording Triggers Extension POU

The Event_Trigger_Out output pin may be used to trigger other recording units such as a Dynamic Disturbance Recorder (DDR) Extension. In this case, we need to assign the Event_Trigger_Out POU output pin of the Recording Triggers Extension to the Dynamic Disturbance Recorder input signal, as shown in *Figure 10.11*.

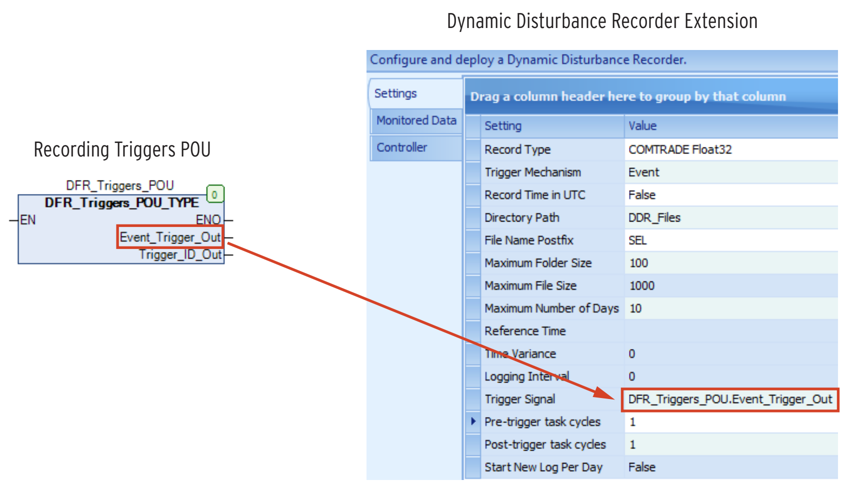


Figure 10.11 Triggering a DDR Using Recording Trigger Extension

Report Generator

Overview

Report Generator creates user-formatted reports on a triggered basis. Reports are saved in a user-specified directory in the RTAC file system. Report Generator offers several reporting modes, which allows individual files to contain one or more triggered reports. In addition to file generation, Report Generator also manages a circular file buffer in the target directory to keep the number of files within a user-specified limit. Report definition is accomplished via a generic, freeform text entry window. The reports can contain text as well as variables of types REAL, LREAL, STRING, DT, timestamp_t, validity_t, BOOL, and any of the various integer data types. When the trigger asserts, all text and the present value of any variables is written to a file. Reports can be saved as .txt, .html, or .csv files.

Use-Cases

Configure multiple Report Generator instances to inform users about notable system events by using report formats, report trigger conditions, and delivery mechanisms customized to each event type. The following are a few examples of possible reports:

- Daily voltage excursion statistics reports saved in the RTAC file system
- Per-event transformer health indicator summary in TXT format pushed to an FTP server
- Physical access alarm email notification formatted in HTML for easy viewing on mobile devices
- Diagnostics report for all connected IEDs, updated hourly and viewable via the SEL server **FILE SHOW** command

Global Settings Tab

The Settings tab contains the Global settings that are applied to each configured report in the Report Generator extension instance. The following reporting methods are available:

- **Single Report Files:** A new file is created to contain each triggered report. File names contain a time stamp that reflects the RTAC system time when the report was triggered. This time stamp has a resolution in seconds.
- **Multiple Report Files:** New triggered reports are appended to the same file. Reports appear in the file with the newest reports at the top. When the number of reports added to the file is equal to the user-configured Maximum Reports Per File setting, a new file is created to contain the new report. File names contain a time stamp that reflects the RTAC system time when the oldest report in the file was triggered. The time stamp resolution is down to the second.
- **Multiple Report Managed File:** New triggered reports are appended to the same file. Reports appear in the file with the newest reports at the top. When the number of reports added to the file is equal to the user-configured Maximum Reports Per File setting, the oldest report is deleted from the file and then the new report is added.

Table 10.5 Global Settings Tab

Name	Description
Directory[a, b]	The directory in which to store and manage report files. 100 characters max. / delimits the folder path. It cannot contain any file path manipulation variables (\\, /./, /../). If the folder does not exist, it is created the first time that a file is written.
Maximum Files Per Directory	The maximum allowed number of files in the directory. When the limit is exceeded, the oldest file is deleted until the remaining number of files is within the limit.
Report Method	Single Report Files, Multiple Report Files, or Multiple Report Managed File.
Maximum Reports Per File	Only configurable when Report Method is Multiple Report Files or Multiple Report Managed File.
Log Run-time Errors	If TRUE, run-time errors are logged to the RTAC SOE viewer.

[a]The Directory setting can contain all printable ASCII characters between 16#20(Space) and 16#7E(tilde) except for ", ', :, <, %, >, ?, \, and |.

[b]Each instance of Report Generator added to an RTAC project uses a dedicated directory manager process. Each instance of Report Generator should use a unique directory location to avoid directory management conflicts.

Reports Tab

Configure report-specific settings.

Table 10.6 Reports Tab

Name	Description
Report Name[a]	The name of the report to which the date and time are appended. 100 characters max.
Report Trigger	Boolean input that triggers a new report on a rising edge.
File Type	TXT, HTML, or CSV. This sets the file name extension but does not affect the formatting of the file content.

[a]The Directory setting can contain all printable ASCII characters between 16#20(Space) and 16#7E(tilde) except for ", ', :, <, %, >, ?, \, and |.

Controller Tab

Use the function block pins to view and modify the state of a Report Generator instance.

Table 10.7 Report Generator POU Output Pin Settings

Pin Name	Pin Type	Description	Default Value
ConsumedDirectorySize	ULINT	Size of all files in the monitored directory in units of bytes.	0
DataStorageWarning	BOOL	Asserts if available file system storage space is 50 MB or less.	FALSE
Error	BOOL	Indicates an internal processing error.	FALSE
ErrorMessage	STRING(255)	A brief description of the issue driving the Error pin.	
Busy	BOOL	When TRUE, this indicates that the report generator is processing a new report, in which case it ignores assertions of the Report Trigger.	TRUE

Text Editor Tab

The Report Generator extension adds a dedicated text editor object for every report configured in the Report tab. The text editor object allows as many as 65535 Unicode characters. The text editor interprets curly brackets ("{" and "}") as special characters. References to logic engine tags can be placed within curly brackets. When the report is triggered, the current value of the referenced tag is written to the report as a string. An open curly bracket enables the autocompletion feature, which aids in tag selection. Tags using the following data types can be converted to strings within the text editor by using curly brackets:

- INT
- DINT
- UINT
- UDINT
- SINT
- USINT
- LINT
- ULINT
- WORD
- DWORD
- LWORD
- BYTE
- REAL
- LREAL
- STRING (up to STRING(255))
- DT
- timestamp_t
- validity_t
- BOOL (single variable or Boolean expression)

Literal Usage of Curly Brackets

To incorporate curly brackets in a compiled report, use the backslash character to indicate the literal usage of a curly bracket character. For example, to print the string "Fault Type {2}" to a report, enter the following text in the Text Editor:

```
Fault Type \{2\}
```

Type Formatting Behavior

Report Generator uses the _TO_STRING type conversion functions noted in *Type Conversion Functions on page 693* for converting all simple data types to string. The following types are formatted using the indicated functions from the SELUtils library:

- REAL types are formatted with REAL_TO_FORMATTED_STRING()
- LREAL types are formatted with LREAL_TO_FORMATTED_STRING()
- timestamp_t types are formatted with TIMESTAMP_TO_STRING()
- validity_t types are formatted with VALIDITY_TO_STRING()

REAL_TO_FORMATTED_STRING() and LREAL_TO_FORMATTED_STRING() functions allow additional input arguments for number representation and decimal point accuracy.

Within the text editor, these input arguments can be added as comma-separated additions to a string conversion call in the format of {TagName, UseScientificNotation, SignificantDecimals}. For example:

```
{REALvar1, TRUE, 3}
```

See SELUtils in the Programming Reference (available from the help menu of the RTAC configuration software) for additional information on descriptions and ranges for allowed input arguments.

Processing Considerations

Each instance of the Report Generator extension uses one instance of the FileIO library class_BasicDirectoryManager and one or more instances of the FileIO library class_FileWriter and class_FileReader. The FileIO library supports a fixed number of file and directory manipulation operations per task cycle for the entire RTAC project. Calls for additional operations within a task cycle are buffered until resources are available to complete the work, which may lead to delays in file writing or directory management. Calls for additional operations within a task cycle over multiple task cycles may lead to an effective freeze of one or more FileIO class instances. While File Writer operations for this extension are largely dictated by assertions of the Report Trigger input, Directory Manager class instances conduct file and directory manipulation operations on a continual basis. For this reason, SEL recommends that you limit the total number of Directory Manager class instances in the RTAC project to eight. This includes instances of library classes and extensions that use FileIO directory management (e.g., Dynamic Disturbance Recorder, Trend Recorder, Report Generator).

FTPSync

Overview

The FTPSync extension provides file synchronization between the RTAC file system and a remote FTP server. User-specified directories and/or IEDs will be monitored for new or updated content for periodic synchronization with the FTP server. Directories and IED events are tracked to ensure that content is synchronized only once and that no content is missed. The extension supports both FTP and SFTP.

Any files or IED events in the RTAC file system can be synchronized. Events collected from the following sources are available for monitoring:

- SEL Protocol Client COMTRADE
- SEL Protocol Client CEV
- MMS Protocol Client COMTRADE
- Modbus Protocol Client COMTRADE
- EtherCAT Module COMTRADE

For directories and IEDs of interest, the directory structure on the RTAC is mirrored to the FTP server. This prevents naming conflicts and allows you to easily locate synchronized content. To ensure that files will be synchronized successfully, confirm that the FTP server has permissions configured to allow the user to upload files, modify files, and create directories.

Settings Tab

The **Settings** tab contains all configurable items for the extension. See the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description. Type any applicable comments in the blank column on the right side.

Directories and IEDs for monitoring are specified in a comma-separated list. In ACSELERATOR RTAC, IED names have the format <specified name>_<protocol>. When listing IEDs for monitoring, only include the specified name.

For example, if the IEDs shown in *Figure 10.12* are collecting COMTRADE events, they will be listed in the **Settings** tab as shown in *Figure 10.13*.

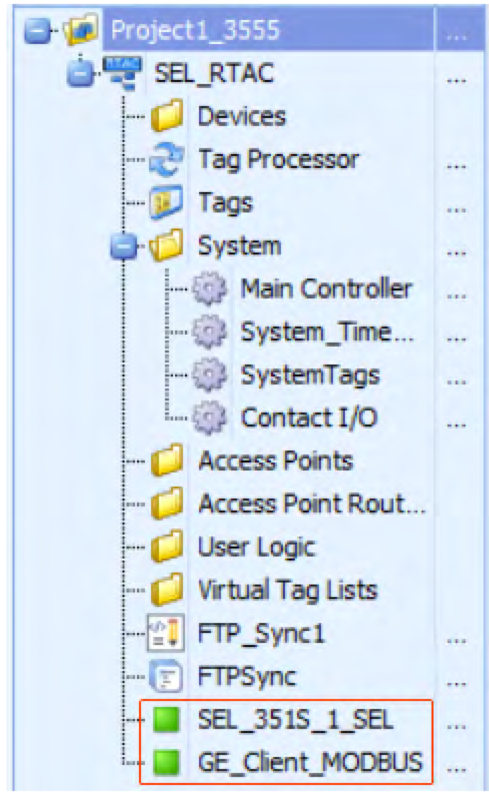


Figure 10.12 Example IEDs in a Project

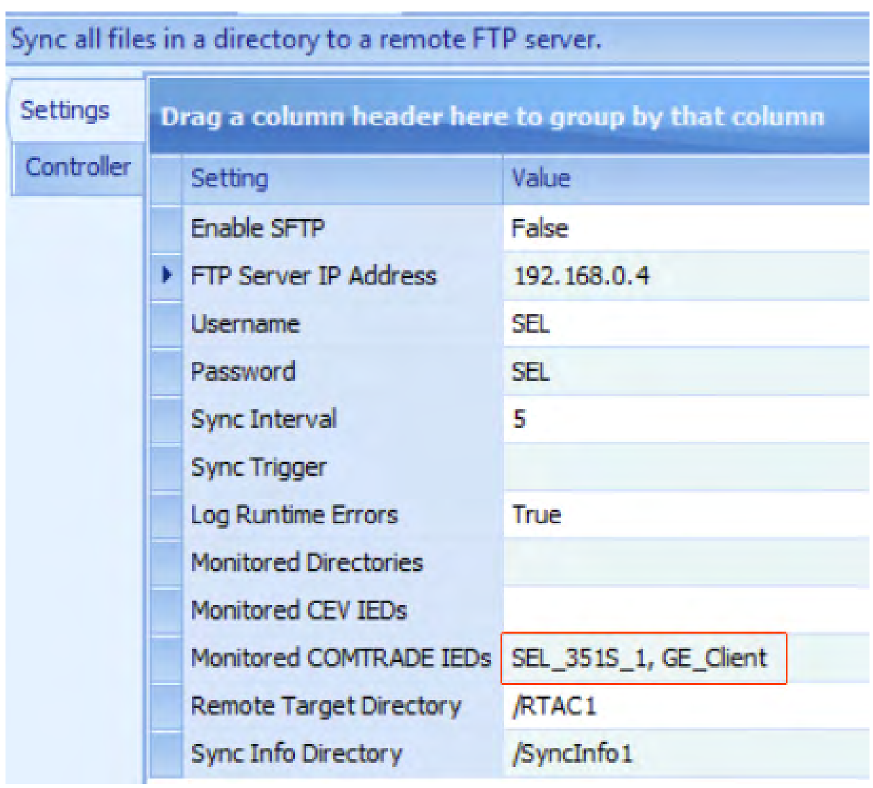


Figure 10.13 Example List of IEDs

SFTP Public Key Authentication

NOTE
Public key authentication is available in firmware versions R147-V0 and above.

Public key authentication can be used with SFTP as an alternative to providing a password. By placing the RTAC's public SSH key on the SFTP server, the SFTP server can authenticate the RTAC for file transfer and encrypt the communication.

First, it is assumed that a user exists on the SFTP server with permissions configured to allow the user to upload files, modify files, and create directories. The password for the user can be set to anything (because it will not be used) or left blank if required as such by the SFTP server. Consult the SFTP server's documentation pertaining to public key authentication for more information.

Next, prepare the key. In the RTAC web interface, select **SSH Keys** under **Security**. Copy the contents of the **Host Key** text box between "—– BEGIN SSH2 PUBLIC KEY —–" and "—– END SSH2 PUBLIC KEY —–".

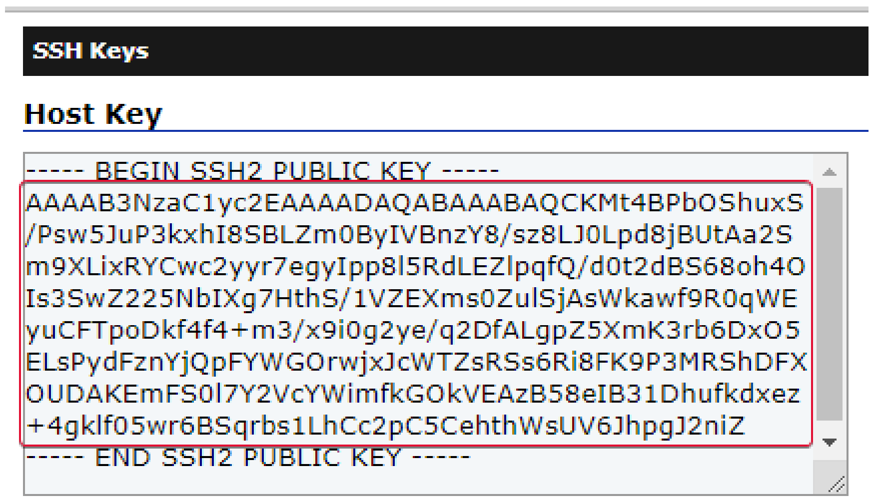


Figure 10.14 Copying the RTAC Host SSH Key

Paste the copied text into an empty text file (e.g., in Notepad++). At the beginning of the file, type "ssh-rsa", followed by a space. At the end of the file, type a space, followed by "<username>@<RTAC hostname>", replacing "<username>" by the target username and "<RTAC hostname>" by the RTAC hostname, e.g., " sally@RTAC1". The RTAC hostname can be found in the web interface by selecting **Interface** under **Network**.

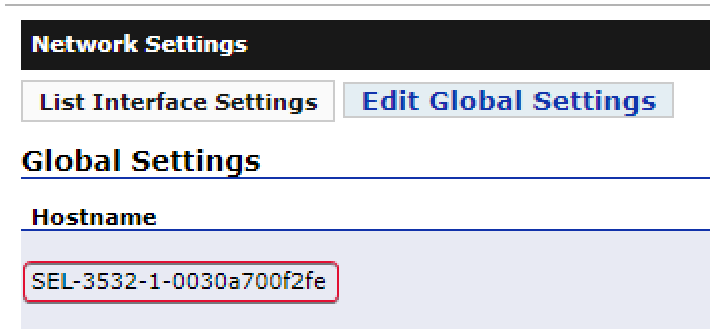


Figure 10.15 Locating the RTAC Hostname

The text should all be on a single line and look similar to the following (shown with line wrapping enabled):

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCStL3ko6ukv7+cZs+m4uvkTUJdIs0hPP3uvOrx+s
Y4qFBsUQQ9jMdxMFsWTcrTQtezAIKXVGYVVF+kVcoAIV+jyim2j7cGbNAJRZvj4C+krMTe
Wgk+tCWImBQk2KjjjKhChovGJd+/XXqxeHA0gW1Wcfsnewdh5zg2WaD3uCv0w9eUdpcVic
9RpfO5TNoqDwmlzauj8D8NLqkh3AMtI4L/99gHKrjdVzqOMR01YNYyJO3jXqa/hB8g2piG
hXb/pEYhBiM51Txq+42YgikSCJ1WiRxZ2Agj4SRaDB4Lm5Z6t3fK9ywHKYhVNwtZvi+678
dHgUHH5Lm53BemloCS/td5 sally@SEL-2241-0030A705AD5A
```

Figure 10.16 Example of Prepared Host Key

At this point, the host key can be copied to the SFTP server. The steps involved vary depending on the SFTP server. Consult the SFTP server's documentation pertaining to public key authentication for more information.

Lastly, to set the FTPSync extension instance to use public key authentication, set the **Username** field to the target username and the **Password** field to "HOST_SSH_KEY".

Controller Tab

Each directory and IED will have output pins on the controller showing the current sync state for that item, numbered uniquely. Additionally, FileSyncInProgress indicates whether or not a sync is active. See *Table 10.8* for the pin descriptions.

Table 10.8 FTPSync POU Pins

Pin Name	Pin Type	Description	Default
FileSyncInProgress	Output: BOOL	TRUE if a file sync is in progress	FALSE
DirectoryName<number>	Output: STRING(255)	The name of the directory or IED	N/A
LastError<number>	Output: STRING(255)	The last error encountered for the directory or IED	""
LastSyncSuccessful<number>	Output: BOOL	TRUE if the last sync was successful	False

Email Plus

Overview

The Email Plus extension provides Simple Mail Transfer Protocol (SMTP) client functionality and includes user authentication, Transport Layer Security (TLS) encryption, and detailed logging for each email sent. Multiple recipients can be specified, including carbon copy (Cc) and blind carbon copy (Bcc) recipients. The extension provides a text editor in which the email body can be formatted, including the ability to reference tags throughout the RTAC project. Emails are triggered by user logic or by any Boolean tag in the project, making the extension ideal for sending emails on alarm conditions.

Settings Tab

The **Settings** tab contains all configurable items for the extension. See the **Description** column for details on each configuration item. Move the slider or hover your cursor over a description to see the entire text of an item description. Type any applicable comments in the blank column on the right side. See the EmailPlus library documentation in the Programming Reference for detailed information on acceptable formatting of email addresses and hostnames. If a hostname is used for the email server, the interface that will communicate with the email server must be configured to use DHCP. DHCP is how the RTAC identifies the DNS server to resolve the hostname to a specific IP address.

Controller Tab

The extension controller has three output pins, as shown in *Table 10.9*.

Table 10.9 Email Plus POU Pins

Pin Name	Pin Type	Description	Default
Initialized	Output: BOOL	TRUE if the extension instance is successfully initialized	FALSE
RuntimeErrors	Output: STRING(255)	Lists any runtime errors encountered	
Busy	Output: BOOL	While TRUE, an email is being sent and no further emails can be triggered	FALSE

Text Editor Tab

The Email Plus extension provides a text editor for formatting email body content. The text editor behavior is described in detail in *Text Editor Tab on page 555*.

Simple Tag Mapper

Overview

The Tag Mapper is used to automatically create the link between a collection of tags from client devices to a list of tags assigned to a server device, generally for SCADA or HMI display purposes. The Tag Mapper is an alternative to the Tag Processor. Tag Mapper focuses on rules for automatically mapping client data to servers, while the Tag Processor offers high levels of customization for client-to-server mapping. Tag Mapper uses a server map and populates the map with data gathered from protocol clients. It also allows for the configuration of other commonly used features, such as Live Data and SOE Logging. The Tag Mapper reduces the amount of time required to map data from protocol clients to SCADA servers.

Setup

Inserting Tag Mapper

From the **Insert** tab, expand the available extensions by selecting the **Extensions** button, and then select **Simple Tag Mapper**.

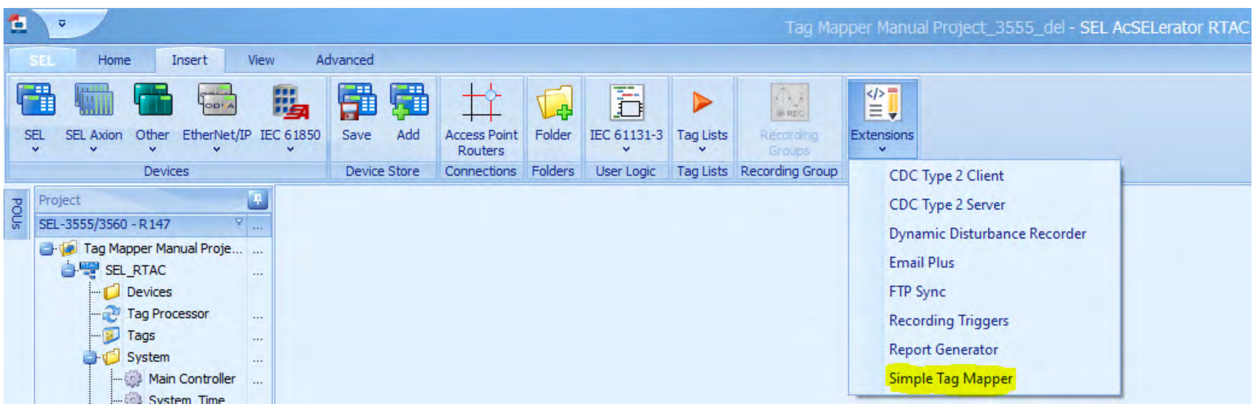


Figure 10.17 Creating Simple Tag Mapper

Tag Mapper supports mapping data to the following server protocol maps.

- DNP
- IEC 60870-5-101/104
- SES-92
- L&G 8979
- Modbus

Server Map Tab

Available server maps in the project will populate the drop-down in the cell located under the **Server Shared Maps** column. Adding an item in the **Server Map** tab will create a **Generated_Map** sub-item and associated row. This row will define the server map to map tags from client devices. The **Server Map** tab will be revisited after the data sources have been configured as a step to initiate the automatic tag mapping to the DNP server map.

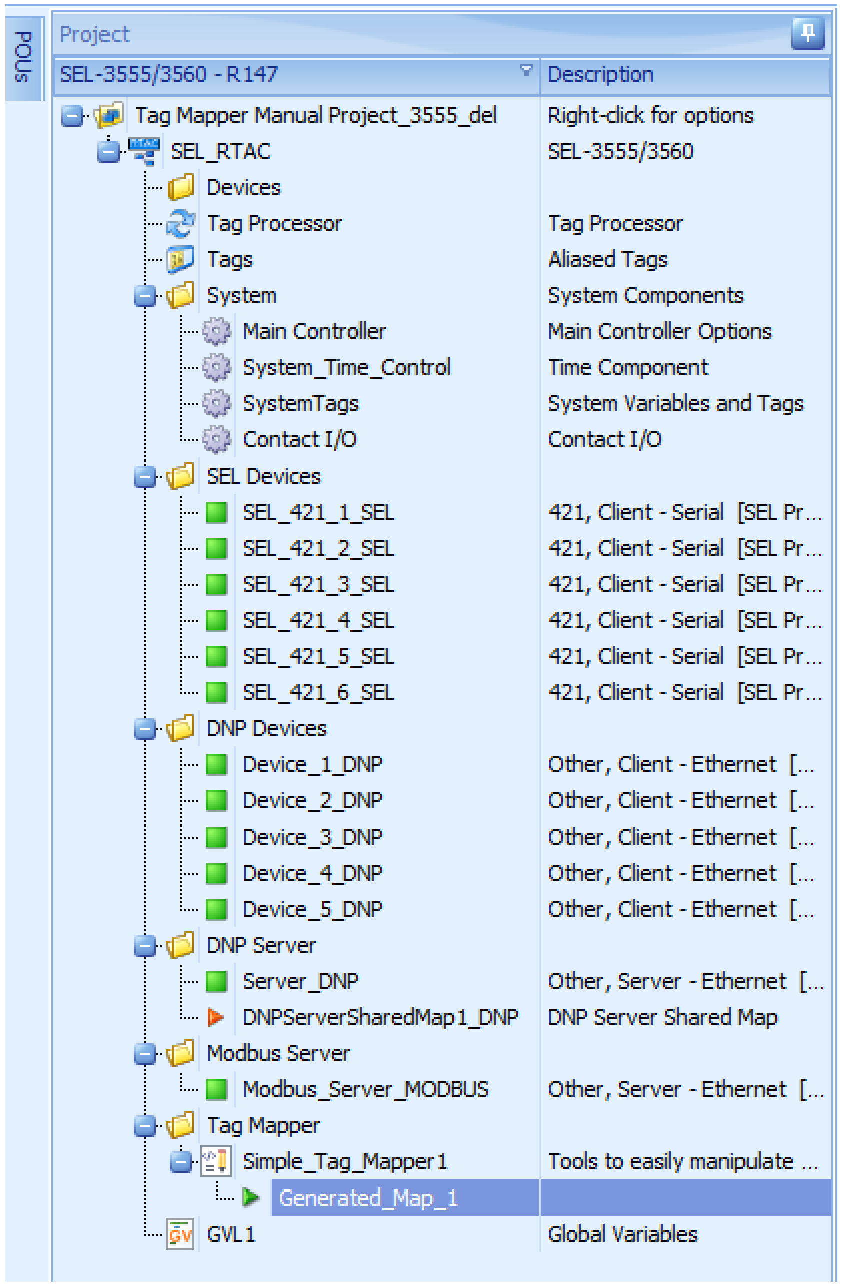


Figure 10.18 Generated Map

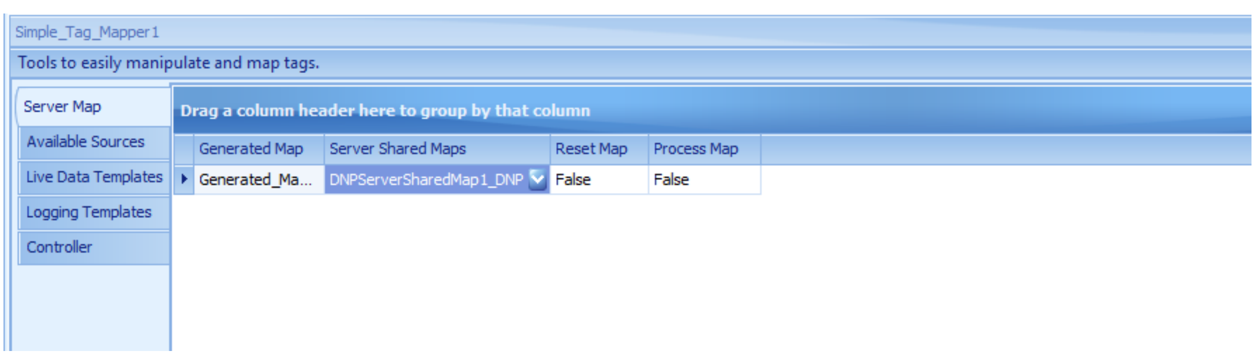


Figure 10.19 Assigning Server Shared Map

Available Sources Tab

The **Available Sources** tab is where clients whose data will be mapped to the specified server map are configured. To add sources (clients) to this list, use the + button and enter the number of desired rows; each row will allow for selecting a single client device.

Supported client device types include:

- CP 2179
- DNP
- EtherCAT I/O Modules
- Flex Parse
- IEC 60870-5-101/104
- IEC 60870-5-103
- IEC 61850 MMS
- IEEE C37.118
- L&G 8979
- Modbus
- SEL Protocol
- SES-92
- SNMP

The columns in the **Available Sources** tab are detailed in the following subsections.

Client

Use this blank cell to select the desired client device from a drop-down menu for each added row. The Tag Mapper will populate the server map with the tags from each client in the order that they appear in the Available Sources list. The order of the tags within a single client is controlled by:

1. Order of the data tabs in each protocol client.
2. The order that the tags were created on that tab. In most cases this will be the same as the order of the tags as they appear on the tab.

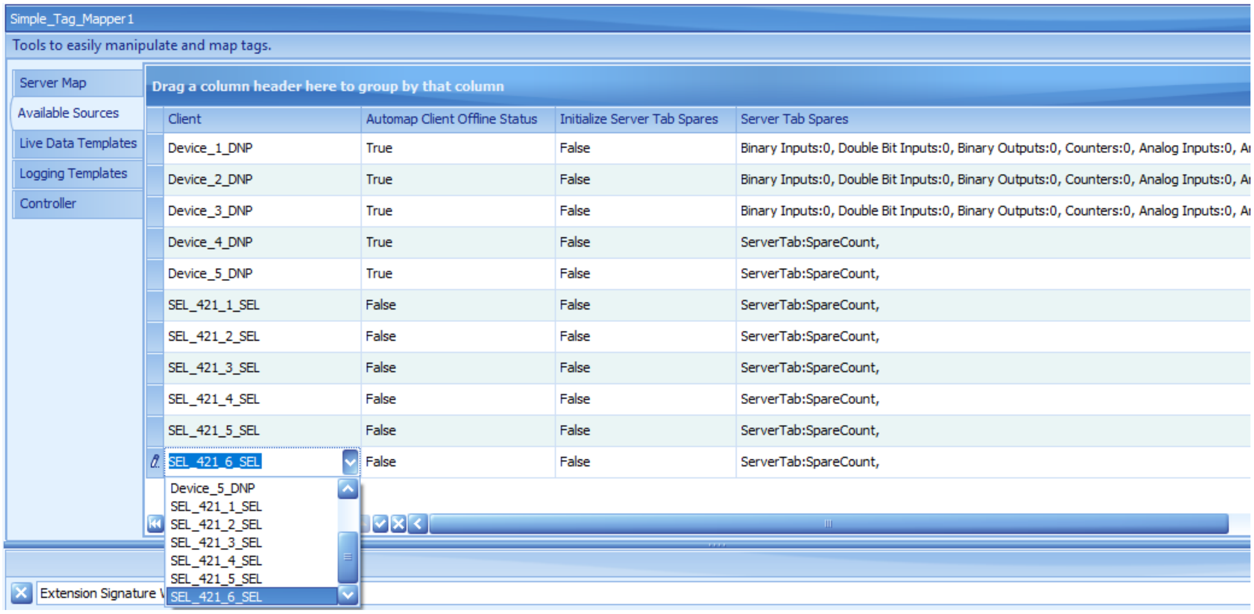


Figure 10.20 Available Sources

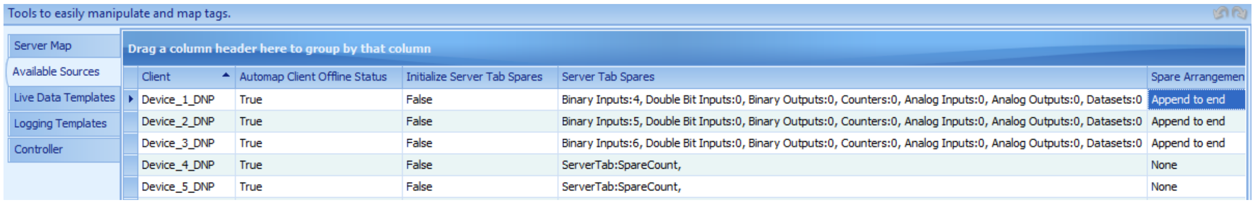
Automap Client Offline Status

If set to True, the Offline bit of the client device is automatically mapped to a point in the server map. The Tag Mapper will use the first Binary Input allocated for a device for the Offline bit. This setting only applies to protocol clients that have an Offline bit, thus EtherCAT I/O modules are not supported and the value of this setting will be ignored for client devices of this type.

Spare Points

Three columns are used to allow users to specify (per client) some spare SCADA tags to be left open in a contiguous space in the automatically generated map for future manual point expansion from the client. Spares are defined by the data category defined in each server protocol (for e.g., "Binary Inputs" on a DNP shared map). The number of spares to reserve for each tab is defined by the values in the **Server Tab Spares** field in a comma-separated list formatted with <tab-name>:<number of spare tags> entries. To initialize this field with default values for each possible server tab, toggle the **Initialize Server Tab Spares** cell to 'True' and then save your settings. After this operation, the **Server Tab Spares** field will contain each tab name from the specified type of server or map on the **Server Map** tab associated with 0 spares. The field can then be manually edited to remove unused tag types or tag types for which you do not intend to define spares.

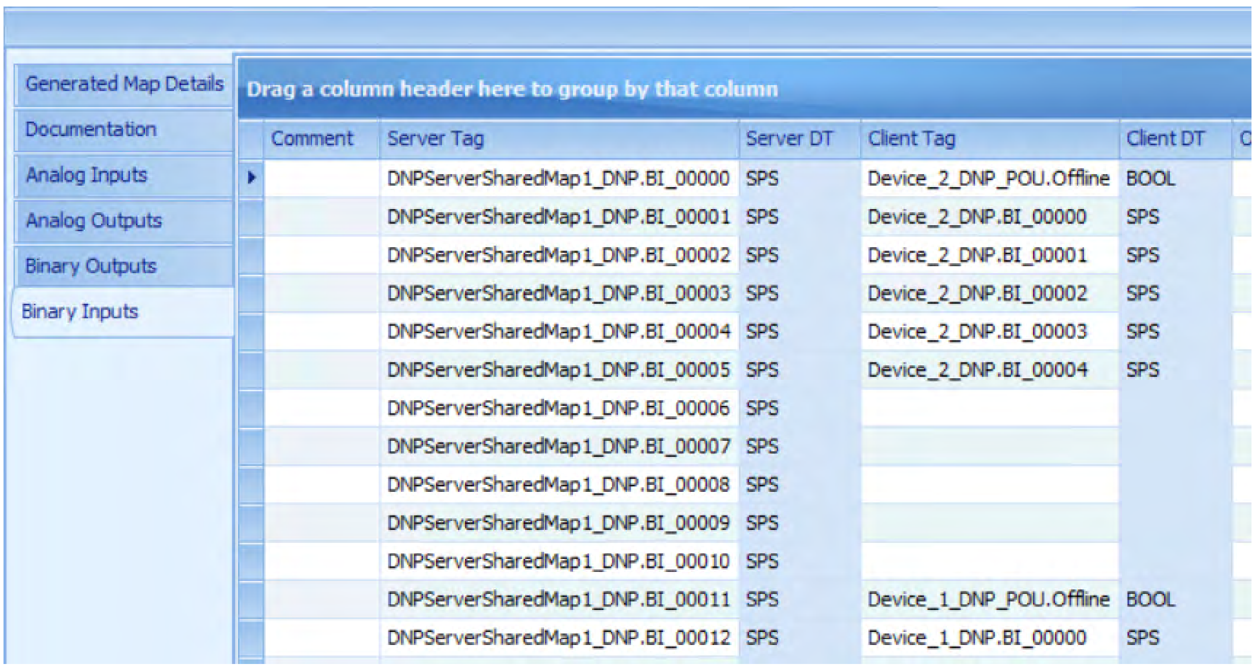
Figure 10.21 shows an example of the format for spare points in **Server Tab Spares** column. This example uses the DNP protocol. The **Spare Arrangement** column is for assigning where the spare points are located with respect to tags mapped from the client device. When the map is generated, the automated mapping function will reserve the spare positions that are configured.



Client	Automap Client Offline Status	Initialize Server Tab Spares	Server Tab Spares	Spare Arrangemen
Device_1_DNP	True	False	Binary Inputs:4, Double Bit Inputs:0, Binary Outputs:0, Counters:0, Analog Inputs:0, Analog Outputs:0, Datasets:0	Append to end
Device_2_DNP	True	False	Binary Inputs:5, Double Bit Inputs:0, Binary Outputs:0, Counters:0, Analog Inputs:0, Analog Outputs:0, Datasets:0	Append to end
Device_3_DNP	True	False	Binary Inputs:6, Double Bit Inputs:0, Binary Outputs:0, Counters:0, Analog Inputs:0, Analog Outputs:0, Datasets:0	Append to end
Device_4_DNP	True	False	ServerTab:SpareCount,	None
Device_5_DNP	True	False	ServerTab:SpareCount,	None

Figure 10.21 Spare Points

The options within a drop-down menu are **Append to start** and **Append to end**. *Figure 10.22* is an example with five spare points appended to the end of the Device_2_DNP client device tags. Note that no **Client Tag** has been assigned in the spare positions and these are left open for future use by the settings engineer, who can manually add in new client tags to these positions without affecting the server tag assignments for all subsequent rows.



Comment	Server Tag	Server DT	Client Tag	Client DT
	DNPServerSharedMap1_DNP.BI_00000	SPS	Device_2_DNP_POU.Offline	BOOL
	DNPServerSharedMap1_DNP.BI_00001	SPS	Device_2_DNP.BI_00000	SPS
	DNPServerSharedMap1_DNP.BI_00002	SPS	Device_2_DNP.BI_00001	SPS
	DNPServerSharedMap1_DNP.BI_00003	SPS	Device_2_DNP.BI_00002	SPS
	DNPServerSharedMap1_DNP.BI_00004	SPS	Device_2_DNP.BI_00003	SPS
	DNPServerSharedMap1_DNP.BI_00005	SPS	Device_2_DNP.BI_00004	SPS
	DNPServerSharedMap1_DNP.BI_00006	SPS		
	DNPServerSharedMap1_DNP.BI_00007	SPS		
	DNPServerSharedMap1_DNP.BI_00008	SPS		
	DNPServerSharedMap1_DNP.BI_00009	SPS		
	DNPServerSharedMap1_DNP.BI_00010	SPS		
	DNPServerSharedMap1_DNP.BI_00011	SPS	Device_1_DNP_POU.Offline	BOOL
	DNPServerSharedMap1_DNP.BI_00012	SPS	Device_1_DNP.BI_00000	SPS

Figure 10.22 Five Binary Input Spare Points Appended to End of Device_2_DNP Points

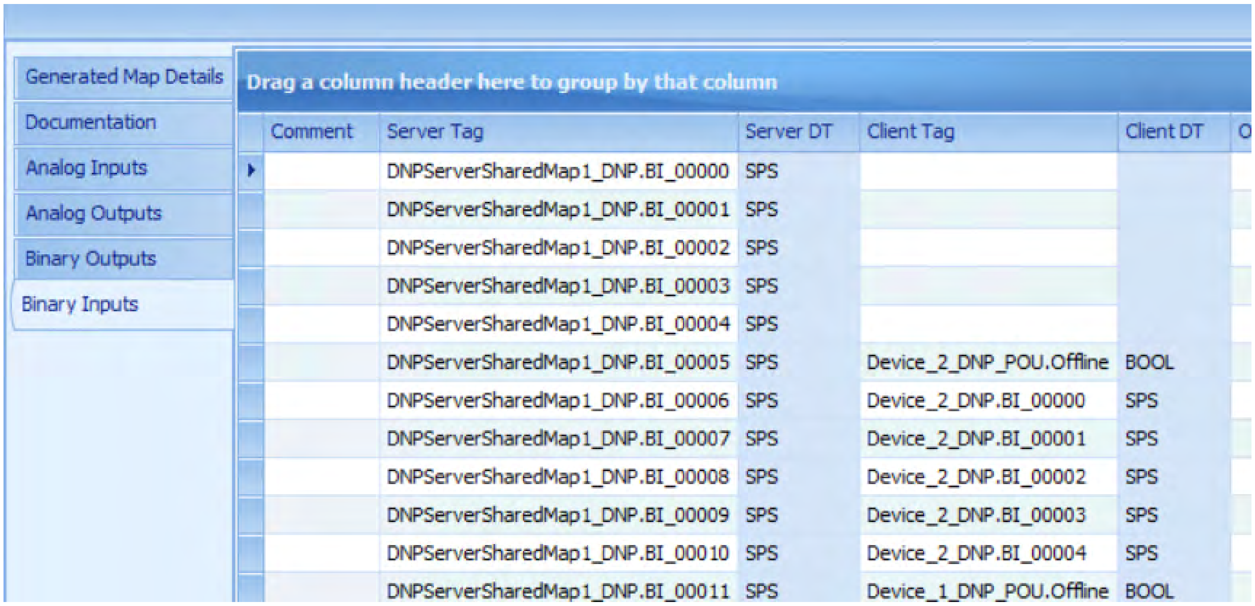


Figure 10.23 Five Binary Input Spare Points Appended to the Beginning of Device_2_DNP Points

Mapping Controls

When performing mapping of "command"-style tags (of type DNPC, DPC, IOC, MDBC, SBRC, SPC, SRBC) between dissimilar tag types, a best-fit operation is performed between the different command "types" within these tags (e.g., operSet, operClear, operPulse, etc.). These operations are applied such that a particular command type received on a server tag will be applied in a consistent way across the other command tags of a dissimilar tag type owned by the client devices. For example, a 'close' operation received on a DNP server (exposed as the operClose attribute of the DNPC tag) will always be applied as a 'remote bit set' (exposed as operSet) operation on an SRBC tag owned by a SEL Client device. The lookup table below shows the complete mapping available between the different command types. A <no-op> entry means that receiving that particular command type on a server tag of a given type will have no effect on the associated client tag (e.g., receiving a 'DNP pulse' on a DNPC tag and mapping that to an 'SBRC' [breaker bit control] tag on a SEL Client).

Server Tag Type			Client Tag Types						
			DNPC	**DPC**	**IOC**	**MDBC**	**SBRC**	**SPC**	**SRBC**
DNPC	operClose	→	operClose	operSet	operSet	operSet	operClose	operSet	operSet
	operLatchOff	→	operLatchOff	operClear	operClear	operClear	operTrip	operClear	operClear
	operLatchOn	→	operLatchOn	operSet	operSet	operSet	operClose	operSet	operSet
	operPulse	→	operPulse	<no-op>	operPulse	<no-op>	<no-op>	<no-op>	operPulse
	operTrip	→	operTrip	operClear	operClear	operClear	operTrip	operClear	operClear
DPC	operClear	→	operTrip	operClear	operClear	operClear	operTrip	operClear	operClear
	operSet	→	operClose	operSet	operSet	operSet	operClose	operSet	operSet
MDBC	operClear	→	operTrip	operClear	operClear	operClear	operTrip	operClear	operClear
	operSet	→	operClose	operSet	operSet	operSet	operClose	operSet	operSet

Server Tag Type			Client Tag Types						
			DNPC	DPC	IOC	MDBC	SBRC	SPC	SRBC
SPC	operClear	→	operTrip	operClear	operClear	operClear	operTrip	operClear	operClear
	operSet	→	operClose	operSet	operSet	operSet	operClose	operSet	operSet

Live Data Templates Tab

Live Data templates are added and configured in Tag Mapper and can be individually applied to the mapped tags. Adding and configuring templates is done from the **Live Data Templates** tab of the Tag Mapper using the + at the bottom of the active screen. The **Template Name** field associated with each entry on this tab should align with a corresponding name entered into the **Live Data Template** column that is present on each data type tab of **Generated_Map_1**. This allows for configuration of multiple live data templates and assignment of them on an ad-hoc basis to the tags in the generated map. The following lists the configurable columns with brief descriptions.

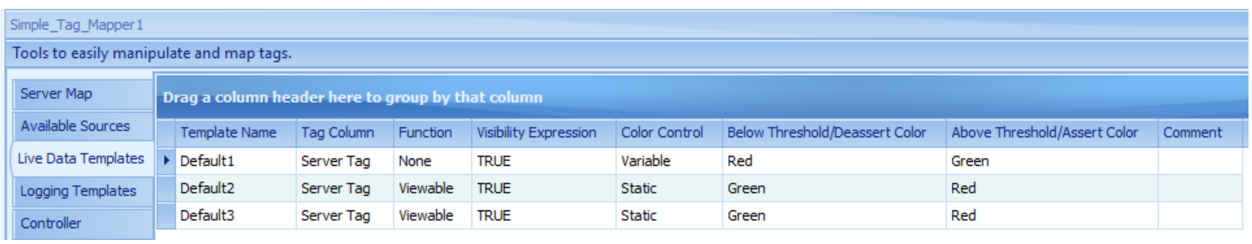


Figure 10.24 Live Data Templates

- Template Name: User-defined name. Unique names are enforced for each template.
- Tag Column: The drop-down assigns Live Data to Server Tag or Client Tag. With Server Tag selected, the Live Data viewer in the web interface will display the tag name as it appears in the **Server Tag** column of its respective data type. Likewise, with Client Tag selected, the Live Data viewer will display the tag name as it appears in the **Client Tag** column of its respective data type. Server Tag is the default setting.
- Function: The drop-down assigns None, Viewable, or Forceable to the Live Data template. If set to None, the tag will not be present in Live data on the RTAC Web Interface. If set to Viewable, the tag will be present. If set to Forceable, the **Prepared** column in the RTAC Web interface is accessible.
- Visibility Expression: This is a Boolean expression that can be used to dynamically control visibility of the signal in the Live Data viewer. For example, if a device goes offline, the POU.Offline Boolean for that device could be used to hide all now invalid data and leave only the communications failure alarm visible. Defaults to TRUE.

- Color Control: The drop-down assigns Variable or Static. Choosing **Static Color Control** will not animate the Live Data entry and all text will be displayed as black on the Live Data viewer. Live Data templates configured with a Variable color control can only be assigned to the following tag types: APC, BOOL, CMV, DPS, INC, INS, MV, SPS.
- Below (Above) Threshold/Deassert (Assert) Color: The drop-down assigns the Live Data text color to red, green, or off (inactive). The color refers to the color of the text of the tag in the RTAC Web Interface. For tag types of a discrete status type (BOOL, SPS, DPS), the assert/deassert colors from the template are assigned those colors to display in the web interface based on the present state of the tag.

For tag types of an analog type (APC, CMV, INC, INS, MV), the fields control the color of the live data entry when the analog is above or below a specified threshold. Thresholds are set in the **Live Data Threshold** column in the respective data type tabs of the generated map.

Logging Templates

Logging templates are added and configured in Tag Mapper and can be individually applied to the mapped tags. Tags that are logged are viewable in the SOE section of the RTAC Web Interface. Supported tag types for logging include APC, BOOL, CMV, DNPC, DPC, DPS, INC, INS, MDBC, MV SBRC, SPC, SPS, SRBC and STR. Adding and configuring templates is done from the **Logging Templates** tab of the Tag Mapper using the **+** at the bottom of the active screen. The **Template Name** field associated with each entry on this tab should align with a corresponding name entered into the **Logging Template** column that is present on each data type tab of **Generated_Map_1**. The following lists the configurable columns with brief descriptions.

- Template Name: User-defined name. Unique names for each template must be used.
- Tag Column: The drop-down assigns Logging to the Server Tag or Client Tag. With Server Tag selected, the Logging statements will be developed around the server tag. Likewise, with the Client Tag selected, the logging statements will be built around the client tag. Default setting is Server Tag.
- Log Condition: The drop-down assigns None, State, Time, State (initial), or Time (initial). The State conditions will log a change of state of the variable. For "State" configurations, any change in state will log the tag. For "State (initial)" configurations, a log entry will be generated when the tag value changes as well as an entry for the initial state of the tag. Likewise, for "Time" configuration options, any change in the time stamp will log the tag.
- Alarm: The drop-down assigns None, True, or False. This field will specify whether this signal is used to populate the 'Alarm' section of the RTAC web HMI when the value is in an alarm state. A value of True can be considered equivalent to 'Alarm on True' and a value of False can be considered equivalent to 'Alarm on False'. A value of None (default) means the signal will not be used for alarming.

- Log Priority: The drop-down assigns Debug, Informational, Notice, Warning, Error, Critical, or Alarm. The selected Log Priority will be displayed in the SOE report under the **Priority** column.
- Log Category: The drop-down assigns Security, Internal, Communications, Logic, Debug, or Tag Data. The selected Log Category will be displayed in the SOE report under the **Category** column. You can customize your own log category by double-clicking on this field and entering in your own string.

Initiating Automap or Making Changes to a Shared Map

Some changes made to the Tag Mapper configuration are not immediately applied to the generated map upon saving the RTAC project. For example, adding a new Client device to the **Available Sources** tab will adjust the configuration of the Tag Mapper, but tags from this client will not be mapped to the Generated_Map until the **Process Map** drop-down (in the **Server Map** tab) has been set to True and the project is saved (see *Figure 10.25*).

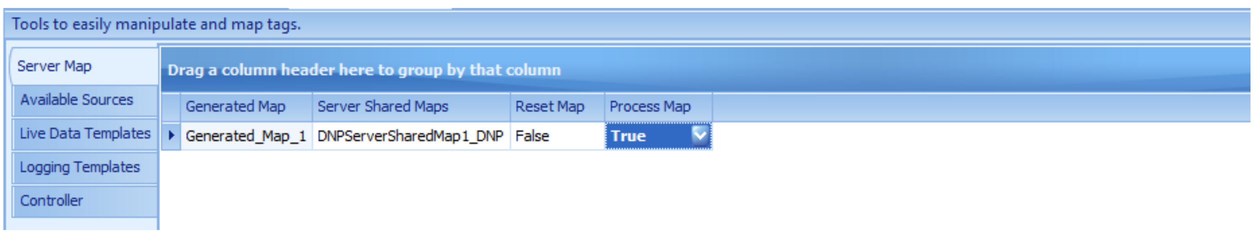


Figure 10.25 Processing Mapping Changes

The **Reset Map** drop-down, when set to True and saved, will rebuild the generated map, thereby undoing any fields in the Generated Map tabs that were manually edited, such as operator functions and logging/live data configuration.

The Generated_Map_1 sub-item includes tabs with various Tag Mapper details. The **Generated Map Details** tab displays the server name, server type, and total number of mapped tags by type (AI, AO, BI, BO, etc.) in the **Server Details** section. Warnings associated with client tags that are unsupported data types, and those that were not able to be resolved to appropriate server tags will also be populated in this section. The **Mapping Details** section displays each source client device that is mapped and the number of each tag that is mapped. The mapping details are separated by tag type.

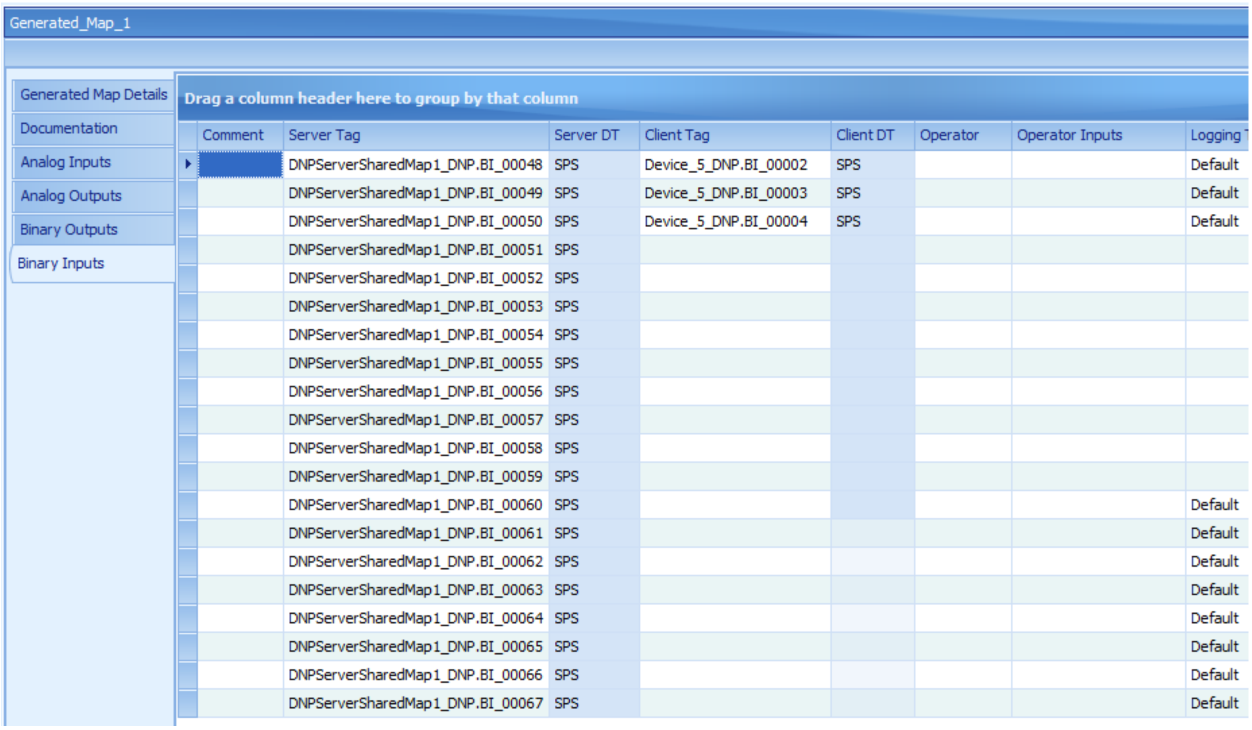
The **Documentation** tab includes information to aid in configuring Tag Mapper. The **Operator Functions** sections gives an overview and valid inputs associated with the **Operator** and **Operator Inputs** columns for the tag types. Other sections in this tab include information on Live Data and Logging functionality.

Editing an Existing Map

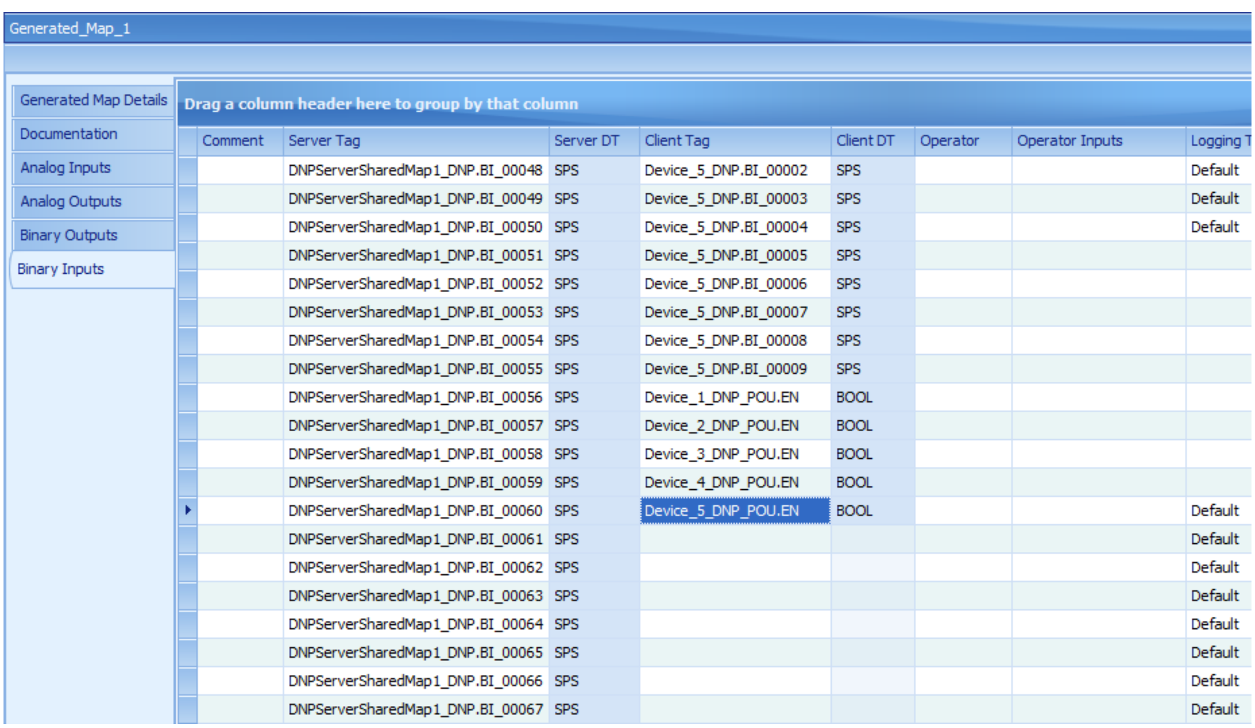
This section details the process of manually adding points to an existing generated map. Utilizing spares in the Tag Mapper allows for individual point configuration like Operator, Logging, and Live Data, for example, to be conserved.

To add points to the generated map, navigate to the Generated_Map_1 data type. The following example is for adding binary input points.

Step 1. Verify that a sufficient number of spare client tags exist.



Step 2. Populate the **Server Tag** and **Client Tag** columns in **Generated_Map_1** with the desired points. Complete any necessary column configuration for the added points.



Step 3. Save the project. If an incompatible data type is used, the ACSELERATOR RTAC software will flag the incompatibility and the client tag will need to be moved or removed before the project can be loaded into RTAC hardware.

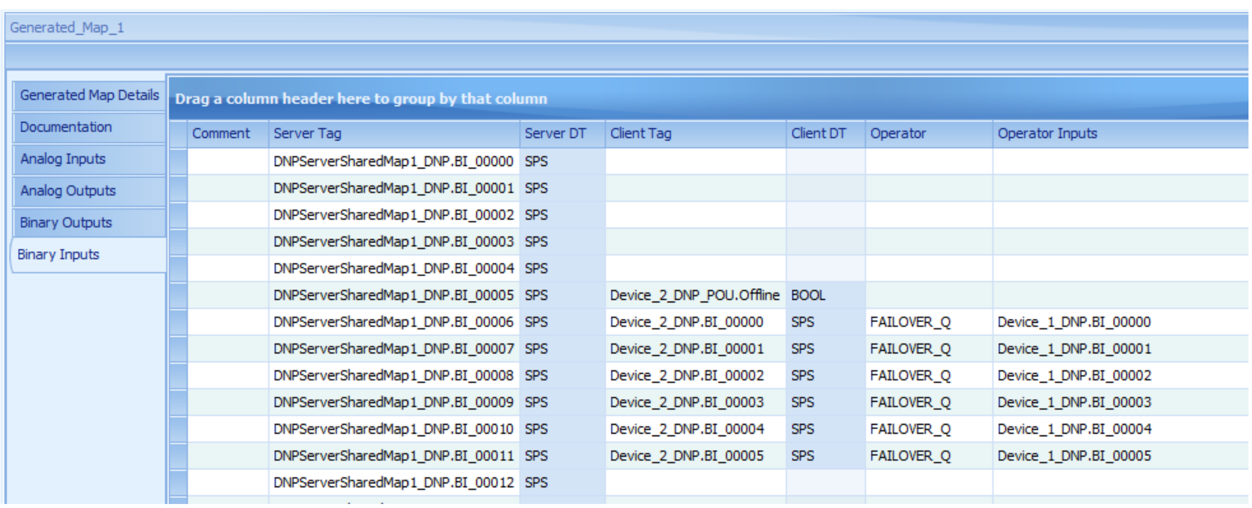
If spare points were included in the initial configuration of Generated_Map_1, points added to the Available Sources can be manually assigned to these spare points the same way as described in the previous example.

Operator Functions

The Tag Mapper has defined operator functions that can be applied to the mapping of client tags to the server map in the Generated Map. The operators are chosen from a drop-down list in the mapping tables and are applied individually The **Documentation** tab of the Generated Map includes instructions on all operator functions. Following is a list of operator functions, a brief description, and the data types that can be applied to the Tag Mapper.

FAILOVER_Q

Figure 10.26 shows an example of how the FAILOVER_Q operator can be used. In this example, Device_2_DNP and Device_1_DNP are redundant devices that are collecting the same data. Tag Mapper will map the client tag (Device_2_DNP.BI_00000, in this example) if the Device_2_DNP client tag quality is good. If the client tag quality transitions to invalid, then the quality value for the alternate Device_1_DNP client tag(s) in the **Operator Inputs** column will be evaluated, and if they are good, the tag will be mapped to the server tag.



Comment	Server Tag	Server DT	Client Tag	Client DT	Operator	Operator Inputs
	DNPServerSharedMap1_DNP.BI_00000	SPS				
	DNPServerSharedMap1_DNP.BI_00001	SPS				
	DNPServerSharedMap1_DNP.BI_00002	SPS				
	DNPServerSharedMap1_DNP.BI_00003	SPS				
	DNPServerSharedMap1_DNP.BI_00004	SPS				
	DNPServerSharedMap1_DNP.BI_00005	SPS	Device_2_DNP_POU.Offline	BOOL		
	DNPServerSharedMap1_DNP.BI_00006	SPS	Device_2_DNP.BI_00000	SPS	FAILOVER_Q	Device_1_DNP.BI_00000
	DNPServerSharedMap1_DNP.BI_00007	SPS	Device_2_DNP.BI_00001	SPS	FAILOVER_Q	Device_1_DNP.BI_00001
	DNPServerSharedMap1_DNP.BI_00008	SPS	Device_2_DNP.BI_00002	SPS	FAILOVER_Q	Device_1_DNP.BI_00002
	DNPServerSharedMap1_DNP.BI_00009	SPS	Device_2_DNP.BI_00003	SPS	FAILOVER_Q	Device_1_DNP.BI_00003
	DNPServerSharedMap1_DNP.BI_00010	SPS	Device_2_DNP.BI_00004	SPS	FAILOVER_Q	Device_1_DNP.BI_00004
	DNPServerSharedMap1_DNP.BI_00011	SPS	Device_2_DNP.BI_00005	SPS	FAILOVER_Q	Device_1_DNP.BI_00005
	DNPServerSharedMap1_DNP.BI_00012	SPS				

Figure 10.26 FAILOVER_Q

FAILOVER_LIMIT

Figure 10.27 shows an example of how the FAILOVER_LIMIT operator can be used. In this example, Device_2_DNP and Device_1_DNP are redundant devices that are collecting the same data. Tag Mapper will map the client tag (Device_2_DNP.AI_00000, in this example) if the value is above the first entry in the **Operator Inputs**, a setpoint of '100'. If the client tag drops below this threshold, then the value for the remaining tag(s) in the **Operator Inputs** column will be mapped to the server tag in the order that they appear as long as they are above the threshold value.

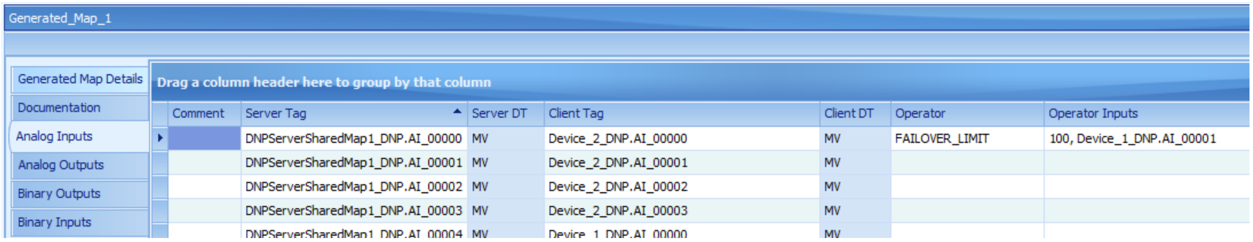


Figure 10.27 FAILOVER_LIMIT

NOT/AND/OR/NAND/NOR

The logic operators must only be used with client tag types BOOL or SPS. The following details the usage of each:

- NOT does not take any Operator Input parameters but will apply the appropriate NOT or NOT_SPS wrapper around the client tag and apply the logical result to the server tag.
- AND/OR will apply the selected logic operation between the client tag and all tags in the Operator Inputs list and apply the results to the server tag. With a client tag type of 'SPS', the OR_SPS and AND_SPS functions are not used and the logic will be written to operate on the stVal sub-attributes to allow use of the base 61131 AND/OR operators. The q.validity of the server tag will be assigned as 'invalid' if the client tag or any tags in the Operator Inputs present a q.validity not equal to 'good'.

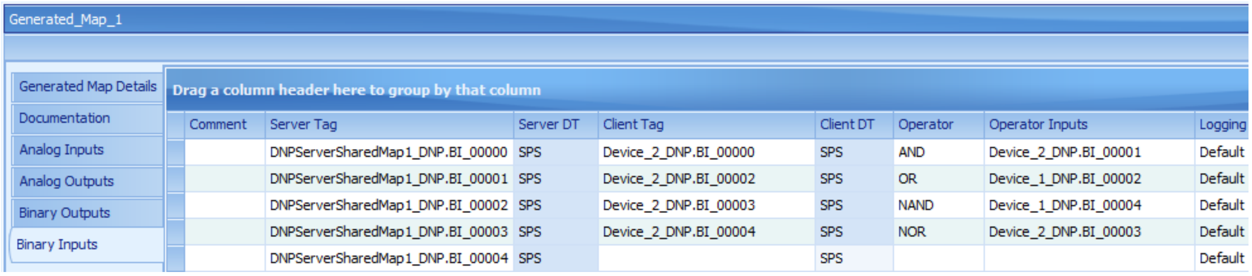
  The time stamp of the server tag will update at the moment the logical result of the operation is different than the stored value already in the server tag.
- NAND/NOR has the same behavior as AND/OR Operators, but the logical result is wrapped in a NOT.

Supported Client Tag Types: BOOL, SPS

Input(s): Operator Input(s) are only needed for AND/OR/NAND/NOR operators.

They should be entered as a comma-separated list of tags (of a like data type with the client tag) to apply in a logical operation against the client tag (e.g., Tag2, Tag3 as operator inputs with an operator of 'AND' would resolve as 'ClientTag AND Tag2 AND Tag3').

In the following example, the AND, OR, NAND, and NOR logical operators are used.



Generated_Map_1

Generated Map Details
Documentation
Analog Inputs
Analog Outputs
Binary Outputs
Binary Inputs

Drag a column header here to group by that column

Comment	Server Tag	Server DT	Client Tag	Client DT	Operator	Operator Inputs	Logging
	DNPServerSharedMap1_DNP.BI_00000	SPS	Device_2_DNP.BI_00000	SPS	AND	Device_2_DNP.BI_00001	Default
	DNPServerSharedMap1_DNP.BI_00001	SPS	Device_2_DNP.BI_00002	SPS	OR	Device_1_DNP.BI_00002	Default
	DNPServerSharedMap1_DNP.BI_00002	SPS	Device_2_DNP.BI_00003	SPS	NAND	Device_1_DNP.BI_00004	Default
	DNPServerSharedMap1_DNP.BI_00003	SPS	Device_2_DNP.BI_00004	SPS	NOR	Device_2_DNP.BI_00003	Default
	DNPServerSharedMap1_DNP.BI_00004	SPS		SPS			Default

Figure 10.28 Boolean Operators

The following sample logic drawing shows a visual representation of the logic operation in place for the first row shown in *Figure 10.29*.

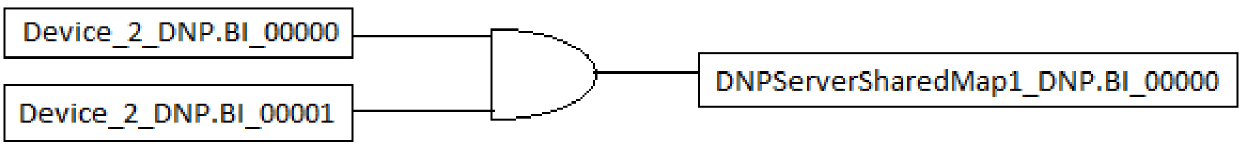
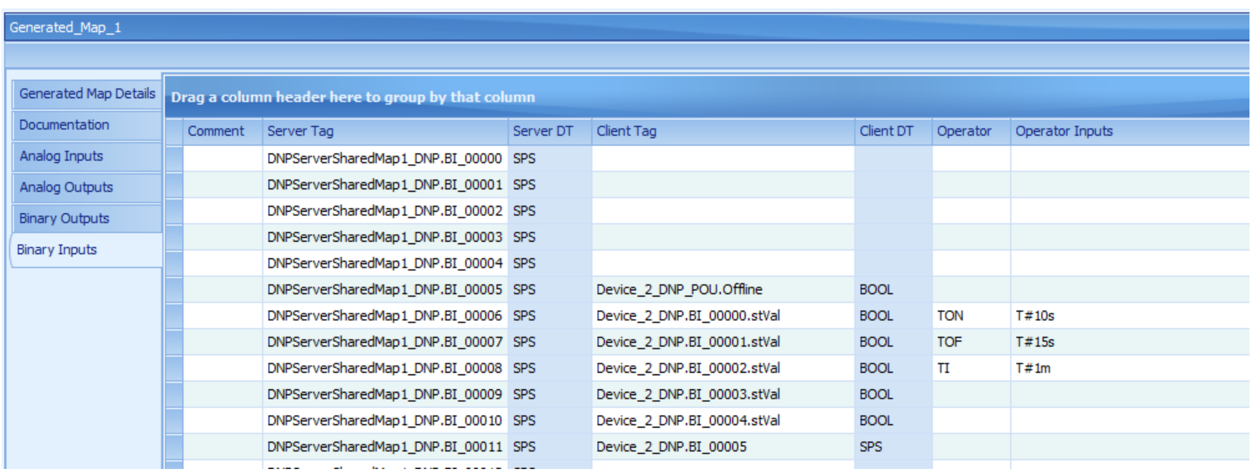


Figure 10.29 Boolean Operator: AND Logic Diagram

TI/TON/TOF/TPUDO

The timer operators must only be used with a client tag of types BOOL or SPS. The following details the usage of each.



Generated_Map_1

Generated Map Details
Documentation
Analog Inputs
Analog Outputs
Binary Outputs
Binary Inputs

Drag a column header here to group by that column

Comment	Server Tag	Server DT	Client Tag	Client DT	Operator	Operator Inputs
	DNPServerSharedMap1_DNP.BI_00000	SPS				
	DNPServerSharedMap1_DNP.BI_00001	SPS				
	DNPServerSharedMap1_DNP.BI_00002	SPS				
	DNPServerSharedMap1_DNP.BI_00003	SPS				
	DNPServerSharedMap1_DNP.BI_00004	SPS				
	DNPServerSharedMap1_DNP.BI_00005	SPS	Device_2_DNP_POU.Offline	BOOL		
	DNPServerSharedMap1_DNP.BI_00006	SPS	Device_2_DNP.BI_00000.stVal	BOOL	TON	T#10s
	DNPServerSharedMap1_DNP.BI_00007	SPS	Device_2_DNP.BI_00001.stVal	BOOL	TOF	T#15s
	DNPServerSharedMap1_DNP.BI_00008	SPS	Device_2_DNP.BI_00002.stVal	BOOL	TI	T#1m
	DNPServerSharedMap1_DNP.BI_00009	SPS	Device_2_DNP.BI_00003.stVal	BOOL		
	DNPServerSharedMap1_DNP.BI_00010	SPS	Device_2_DNP.BI_00004.stVal	BOOL		
	DNPServerSharedMap1_DNP.BI_00011	SPS	Device_2_DNP.BI_00005	SPS		

Figure 10.30 Timer Examples

SCALE

Supported Client Tag Types: APC, CMV, DINT, INC, INS, MV, REAL, UDINT

Input(s): Comma-separated list of scaling parameters (e.g., 100, 0, 32767, -32768):

- Input 1: scaling multiplier (decimal values accepted)—mandatory
- Input 2: offset (optional, defaults to '0')—optional

- Input 3: ceiling (optional, defaults to full-scale maximum)—optional
- Input 4: floor (optional, defaults to full-scale minimum)—optional

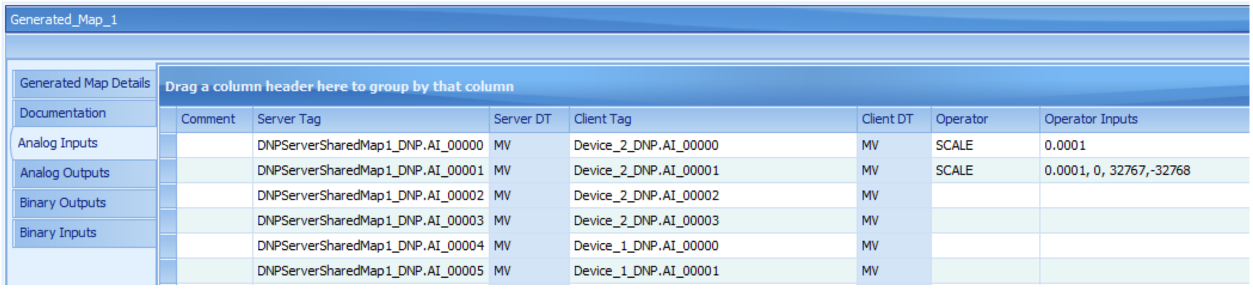


Figure 10.31 Scaling Examples

PACK

The bit-pack function is only applicable for INS server tags. The client tag is ignored and the Operator Inputs field contains a comma-separated list of at least 1 (and up to 32) Boolean quantities and copies them into the individual bits in the .stVal DINT value contained in the INS server tag. The first operator input parameter is assigned to stVal.0, the second to stVal.1, and so on until stVal.31.

Supported Server Tag Types: INC, INS

Inputs: Comma-separated list of between 1–32 Boolean quantities to bit-pack into the server tag DINT (e.g., TRUE, FALSE, 0, 1, TRUE).

In the following example, the lowest six bits of the DNPServerSharedMap1_DNP.AI_00000 tag will be populated as per the following:

- 0 = TRUE
- 1 = TRUE
- 2 = FALSE
- 3 = TRUE
- 4 = TRUE
- 5 = Boolean state of Device_2_DNP.BI_00000.stVal

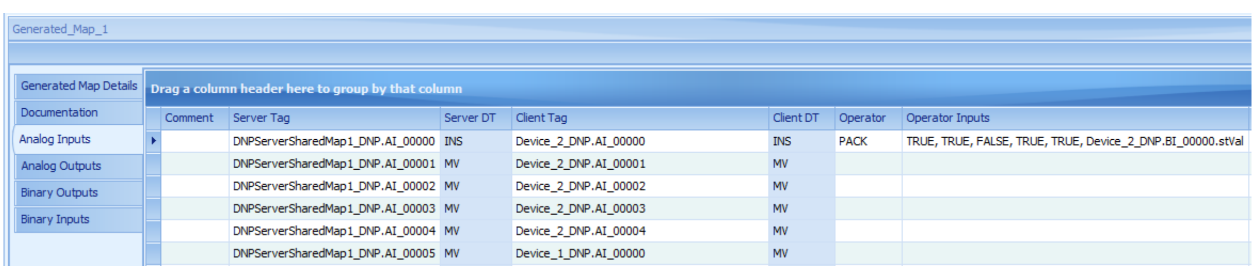


Figure 10.32 Bit Packing Example

PAIRED

Configure a pair of client tag remote bits (e.g., RBxx:RByy) to align with DNP functionality in SEL IEDs. Issuing a 'trip' or 'latch-off' via a DNPC server tag will result in a pulse being issued to the SRBC tag present in the **Client Tag** column. This tag can be thought of as 'RBxx' in the remote bit pair. Issuing a 'close' or 'latch-on' or 'pulse-on' via a DNPC server tag will result in a pulse being issued to the SRBC tag present in the **Operator Inputs** field. This tag can be thought of as 'RByy' in the remote bit pair.

Input(s): A single client tag of type SRBC that specifies RByy in the remote bit pair.

> **⚠WARNING**
> Do not run a 'Process Map' or 'Reset Map' after paired remote bits have been configured. Otherwise, additional manual steps must be performed to correct the Binary Outputs table following this operation. Tag Mapper assumes that binary output tag assignments are a single server tag to single client tag and this is the mapping that will be generated whenever the 'Process Map' or 'Reset Map' toggle is selected. If a remote bit pair is configured between RB1 and RB2, for example, then this assumption is no longer valid because we now have a single server tag mapped to two client tags. The subsequent row containing RB2 in the client tag field must be deleted to avoid duplicating the assignment to this tag. However, if setting Process Map or Reset Map to True, then the automated mapping algorithm will re-add the RB2 binary output row that was removed and RB2 would have two assignments again.

The **Paired** operator is selectable for Binary Outputs in the **Operator** column of **Generated_Map_1**. The **Paired** operator is used when a binary output in the shared server map is applied to the two binary outputs of two available sources. The **Client Tag** in the **Binary Outputs** of **Generated_Map_1** is the primary target of the **Server Tag** and the paired target is defined in the **Operator Inputs** column.

In *Figure 10.33*, the first binary output of the shared server map, **DNPServerSharedMap1_DNP.BO_00000**, will pulse **Ethernet_451_SEL.FO_RB_RB1** and whenever a pulse, trip, or latch-on operation is issued to that binary output. Similarly, if a close or latch-off operation is issued to **DNPServerSharedMap1_DNP.BO_00000**, then **Ethernet_451_SEL.FO_RB_RB2** is pulsed.

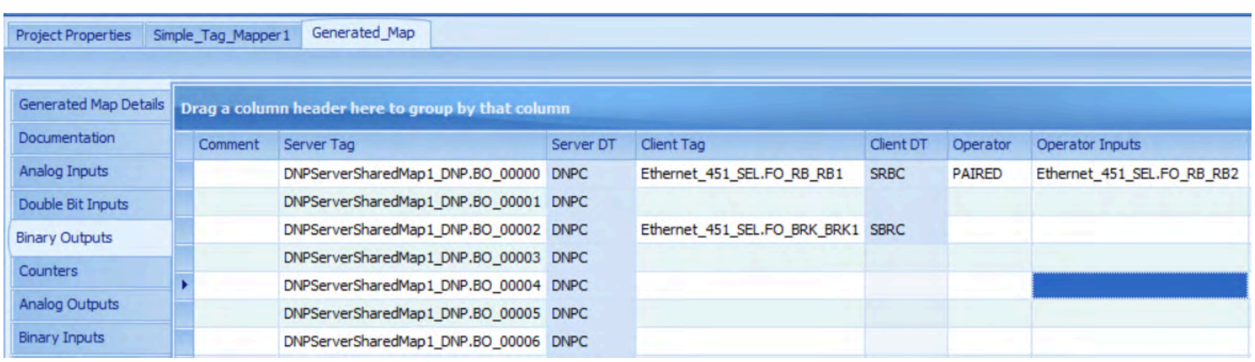
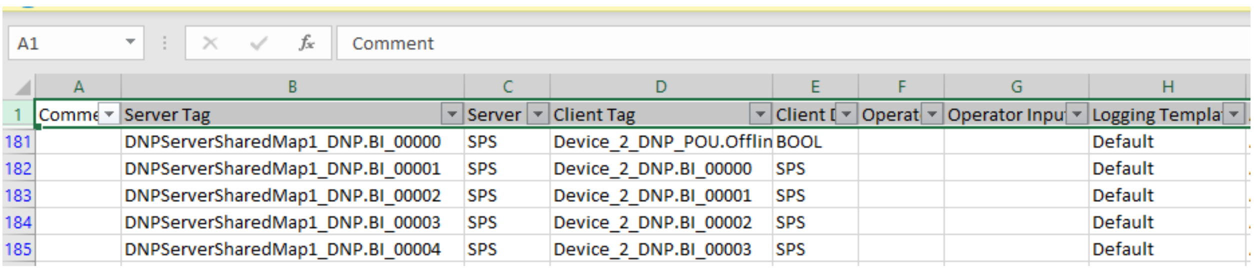


Figure 10.33 Paired Binary Outputs

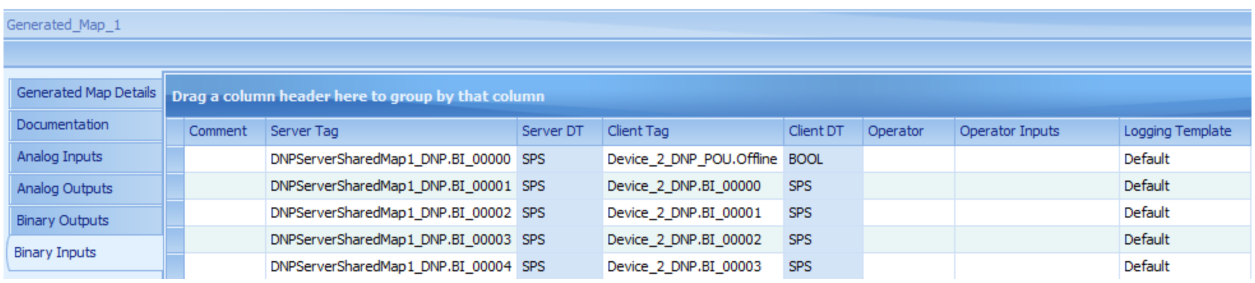
CSV Output File

Tag Mapper creates a user-readable .csv file that serves as a record for the tag mapping and configuration of all tags in tabular form. This file can be provided to the HMI or SCADA programmers as documentation for how the HMI/SCADA client should be configured to collect data from the RTAC. The following figures show the top row of the .csv matches the column titles in **Generated_Map_1**.



	A	B	C	D	E	F	G	H
1	Comme	Server Tag	Server	Client Tag	Client [	Operat	Operator Inpu	Logging Templa
181		DNPServerSharedMap1_DNP.BI_00000	SPS	Device_2_DNP_POU.Offlin	BOOL			Default
182		DNPServerSharedMap1_DNP.BI_00001	SPS	Device_2_DNP.BI_00000	SPS			Default
183		DNPServerSharedMap1_DNP.BI_00002	SPS	Device_2_DNP.BI_00001	SPS			Default
184		DNPServerSharedMap1_DNP.BI_00003	SPS	Device_2_DNP.BI_00002	SPS			Default
185		DNPServerSharedMap1_DNP.BI_00004	SPS	Device_2_DNP.BI_00003	SPS			Default

Figure 10.34 Column Header: Output .csv File



Comment	Server Tag	Server DT	Client Tag	Client DT	Operator	Operator Inputs	Logging Template
	DNPServerSharedMap1_DNP.BI_00000	SPS	Device_2_DNP_POU.Offline	BOOL			Default
	DNPServerSharedMap1_DNP.BI_00001	SPS	Device_2_DNP.BI_00000	SPS			Default
	DNPServerSharedMap1_DNP.BI_00002	SPS	Device_2_DNP.BI_00001	SPS			Default
	DNPServerSharedMap1_DNP.BI_00003	SPS	Device_2_DNP.BI_00002	SPS			Default
	DNPServerSharedMap1_DNP.BI_00004	SPS	Device_2_DNP.BI_00003	SPS			Default

Figure 10.35 Column Header: Tag Mapper

The file is initially created the first time **Process Map** is set to True and the project is saved. Subsequent project saves will update the .csv file if the **Process Map** is set to True and changes have been made to **Generated_Map_1**.

The name of the .csv file is titled **SimpleTagMapper_ClientToServerMapping.csv** and is created in the following Windows directory path:

Documents\AcRTAC\Logs\Extension Logs

CtPt Monitor

Overview

The CtPt Monitor extension provides health assessment of CT or PT assets through differential analysis of streaming measurements from redundant CTs or PTs.

Theory of Operation

When redundant measurement transformers (CTs or PTs) are connected to a single line or bus (respectively), the set of transformers will produce equivalent secondary output currents or voltages (within a known margin of error as specified by the CT or PT manufacturer), which can be converted to digital measurements. By continually scrutinizing these streaming measurements for relative differences, it is possible to identify a failing measurement transformer. Secondarily, this comparative analysis can be used to validate and monitor the analog-to-digital conversion processes in the connected sampling equipment. A user-specified deviation percentage and pickup time are used to drive a time-stamped alarm, indicating an asset health issue.

This extension uses class_StreamingCTPTMonitor from the ConditionMonitoring library. See ConditionMonitoring in the Programming Reference for additional information.

> **NOTE**
> The CtPt Monitor extension requires the RTAC to have the ConditionMonitoring library license.

Features

This extension provides the following features:

- Monitor two or more redundant assets per monitor group.
- Multiple monitor groups can be added to a single extension instance.
- Assets are monitored in two ways:
  - Against a specified reference measurement
  - The reference for comparison is an average of all monitored measurements. The averaging function removes outliers from its calculation.
- Monitor groups can be enabled or disabled dynamically with user logic.
- Optionally enabled generic time-alignment algorithm provides time-coherence among measurements sourced from various communication protocols that do not provide built-in time alignment.
- Support for CMV, MV, and INS data types provides compatibility with measurements sourced from various communication protocols. Note that processing is performed on magnitudes only. Thusly, CMV angles are discarded.
- Settable scaling factor provides for monitoring of CTs or PTs separated by a power transformer.

Settings

The following sections describe the CtPt Monitor settings interface.

Static vs Dynamic Settings

The following sections describe the settings of the extension. The **Setting Type** column indicates one of the following options:

1. Static: This input type is intended to be set once and cannot change after going online. Settings of this type cannot be assigned to a tag reference, logic engine object, or the output of an expression/equation.
2. Dynamic: This accepts several types of inputs:
   - Tag references
   - Function Block outputs/Function returns
   - Expressions
   - Literals (for example, 'True' for a Boolean data type or '5.0' for a REAL data type)

For non-literal inputs to dynamic settings, the **Data Type** column indicates the required data type for the input tag reference, logic engine object, or expression output.

Monitor Group Settings

Use the **Monitor Groups** tab to define global settings for each group of redundant measurement Assets.

Table 10.10 CtPt Monitor Group Settings

Setting Name	Data Type	Setting Type	Description	Default Value
Group ID	STRING	Static	Identification string for the monitor group.	Group_<n> where 'n' is the row number
Enable Monitor Group	BOOL	Dynamic	Enable or disable the Monitor group with a Boolean value or expression.	FALSE
Minimum Signal Magnitude	REAL	Dynamic	Minimum signal level required from all measurements associated with a given asset monitor subprocess to enable the monitor.	0
Alarm Deviation Threshold	REAL	Dynamic	(Percentage) Absolute deviation from the reference required to drive a health alarm indication.	5.00
Alarm Pickup Time	INT	Static	(Seconds) Maximum time a sustained deviation beyond Alarm Deviation Threshold is allowed. After the initial deviation, momentary normalization is disregarded.	60
Monitor Mode	STRING	Static	Select defined or relative reference modes.	Relative-Reference
Enable Time-Alignment	BOOL	Static	Apply time-alignment to all reference and/or monitored channels associated with the monitor group.	FALSE
Data Rate	INT	Static	(Msg/sec) Time-alignment rate.	1
Maximum Wait-Time	INT	Static	(Milliseconds) The amount of time that the time-alignment subprocess waits to receive all expected samples for a given time stamp. Values less than the task cycle time are rounded up to the task cycle time.	200

Monitor Mode Considerations

The **Monitor Mode** setting is provided to allow for monitoring scheme flexibility. Advantages and limitations of each mode are summarized as follows:

- Monitor Mode = Relative Reference
  - General use-case: Each redundant CT or PT connected to a common power system asset is monitored against each other.
  - Advantages: No single point of failure.
  - Limitations: This monitoring scheme is only capable of accurately identifying simultaneous CT or PT failures in as many as (N / 2 - 1) CTs/PTs for an even number of monitored CTs/PTs and (N - 1) / 2 CTs/PTs for an odd number of monitored CTs/PTs (where N is the number of monitored PTs or CTs).
  - Special Considerations: This mode calculates a reference signal for comparison against each monitored channel. This reference signal is derived from the outlier-reduced average of all monitored channel samples for a snapshot in time. See class_StreamingCTPTMonitor in the Programming Reference for more information.
- Monitor Mode = Defined Reference
  - General use-case: One CT or PT is designated as a point-of-reference for use in monitoring other CTs or PTs connected to a common power system asset.
  - Advantages: This monitoring scheme can identify simultaneous CT or PT failures in as many as (N - 1) CTs or PTs (where N is the number of monitored PTs or CTs).
  - Limitations: This scheme is contingent on the reference PTs or CTs continually operating within manufacturer specifications. Thus, the reference CTs or PTs are single points of failure in this scheme.

Measurement Channel Settings

For each row added to the **Monitor Groups** tab, a new child sub-item will be added to the extension, as shown in *Figure 10.36*. The child object is used to specify all measurements associated with the given monitor group.

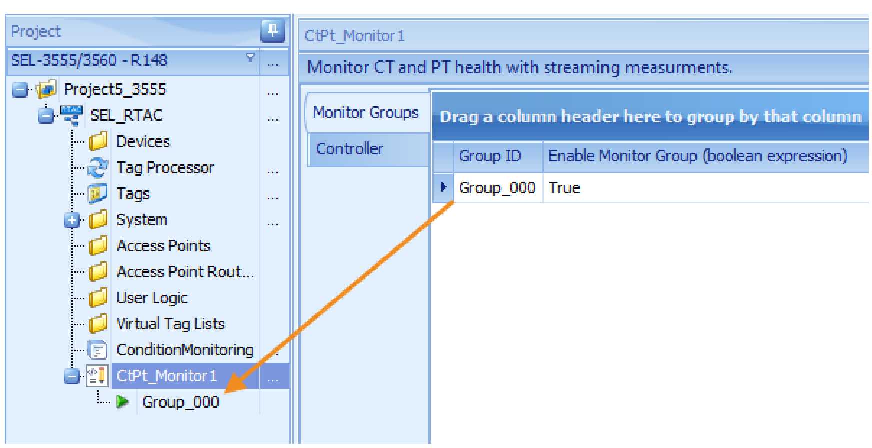


Figure 10.36 CtPt Monitor Group Measurement Channel Specification Page

Monitored Channels

Each sub-item will contain a **Monitored Channels** tab for specifying measurements for all CTs or PTs to be monitored. Select the + button to add as many rows as needed for the monitor group.

Table 10.11 Monitored Channel Settings

Setting Name	Data Type/ Range	Setting Type	Description	Default Value
CT/PT ID	STRING	Static	Identification string for the monitored asset.	Dynamically generated, based on row number.
Channel Source Tag	CMV, MV, or INS	Dynamic	Measurement channel source tag.	
Phase	0, 1, 2, A, B, C, or NA	Static	Electrical phase or sequence component associated with the Channel Source Tag. For Monitor Mode = Relative Reference, each phase represented must correspond to at least two monitored channels.	A
Scale Factor	REAL	Static	Multiplication factor to be applied to the channel measurement, prior to processing.	1.0

Reference Channels

For monitor groups configured with Monitor Mode = Defined Reference, the associated sub-item contains a **Reference Channels** tab in addition to the **Monitored Channels** tab. Use the **Reference Channels** tab to specify all measurements from reference CTs or PTs. Reference CT/PT measurements are used to assist in determining the health status of monitored CTs/PTs (as specified on the **Monitored Channels** tab). Reference CTs/PTs are not monitored for health status. Select the + button to add a reference channel row.

Special considerations about the **Reference Channels** tab:

- Reference Channel settings are identical to Monitor Channel settings with the exception that a CT/PT ID string setting is not included.
- Each reference channel must be assigned unique Phase.
- The Phase of each monitored channel within the associated **Monitored Channels** tab must have a corresponding reference channel measurement with the same Phase setting.

Outputs

The CtPt Monitor generates a POU object on the **Controller** tab with the following outputs.

Table 10.12 CtPt Monitor POU Outputs

Output Name	Data Type	Description	Default Value
Initialized	BOOL	TRUE if all monitor groups have successfully initialized.	FALSE
Error	BOOL	TRUE if any monitor group is in an unrecoverable error state.	FALSE
ErrorMessage	STRING(255)	Error description for general errors pertaining to the extension instance.	

Output Name	Data Type	Description	Default Value
<Group ID 1>_Status	Custom Data Type	Custom data type containing asset health information for all monitored channels in <Group ID 1>.	
<Group ID 1>_Error	BOOL	TRUE if the monitor group is in an unrecoverable error state.	FALSE
<Group ID 1>_ErrorMessage	STRING(255)	Error description for the first monitor group defined on the **Monitor Groups** page.	
⋮	⋮	⋮	⋮
<Group ID n>_Status	Custom Data Type	Custom data type containing asset health information for all monitored channels in <Group ID n>.	
<Group ID n>_Error	BOOL	TRUE if the monitor group is in an unrecoverable error state.	FALSE
<Group ID n>_ErrorMessage	STRING(255)	Error description for the last monitor group defined on the **Monitor Groups** page.	

An example of the POU output pins is shown below for a scenario where three monitor groups were added to the extension: LineX_CTs, LineY_CTs, and LineZ_CTs.

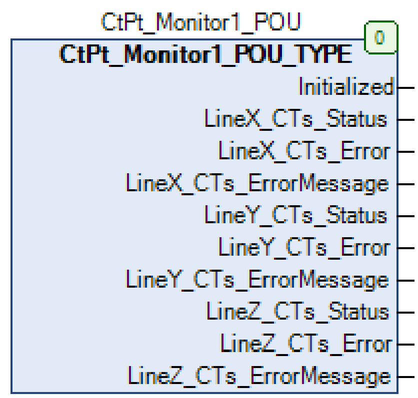


Figure 10.37 Example CtPt Monitor POU Outputs

Definition of Custom Data Types

The extension will dynamically generate data structures that are tailored to each monitor group. For example, the LineX_CTs_Status, LineY_CTs_Status, and LineZ_CTs_Status outputs shown in *Figure 10.37* each represent a custom data structure. The purpose of the structure is to contain asset health information for each monitored asset in each monitor group. These custom data structures take the following form.

Table 10.13 Custom Data Types Used on CtPt Monitor POU Output

Element Name	Data Type	Description
<CtPtID 1>	Struct_CtPtStatusOutput	A dedicated status structure for the first channel added to the **Monitored Channels** tab for the given monitor group. The element name corresponds to the user-specified **CT/PT ID**.
⋮	⋮	⋮

Element Name	Data Type	Description
<CtPtID n>	Struct_CtPtStatusOutput	A dedicated status structure for the last channel added to the **Monitored Channels** tab for the given monitor group. The element name corresponds to the user-specified **CT/PT ID**.
MonitorAddressArray	ARRAY [1 .. ArrayEndIndex] OF POINTER TO struct_CTPTStatusOutput	Array of pointers to elements <CtPtID 1> through <CtPtID n>.
ArrayStartIndex	UDINT	Shall be fixed at '1'. This is useful when navigating the MonitorAddressArray with a loop.
ArrayEndIndex	UDINT	Shall be equal to the number of monitored channels added to the **Monitored Channels** page for the given monitor group.

General Usage of Custom Data Structures

Each custom data structure contains one struct_CTPTStatusOutput structure (defined in *ConditionMonitoring* in the Programming Reference) for each channel on the **Monitored Channels** tab of each monitor group. This allows status information from each monitored asset to be referenced in subsequent logic via the dot notation, as shown in the following example:

> ExtensionInstance_POU.GroupID_Status.CtPTID.<Element from struct_CTPTStatusOutput>

Advanced Usage of Custom Structures

Asset health information for a given monitor group can be accessed programmatically by using the MonitorAddressArray, ArrayStartIndex, and ArrayEndIndex in a FOR or WHILE loop.

Usage of struct_CTPTStatusOutput

For each monitor group defined in the extension, the associated POU provides an output structure containing status information for each asset specified on the **Monitored Channels** page. This status for each asset is contained in a struct_CTPTStatusOutput structure and is populated as described in *Table 10.14*.

Table 10.14 CtPt Monitor Status Structure (struct_CTPTStatusOutput)

Element Name	Data Type	Description
CtPtID	STRING(255)	Set equal to the user-specified **CT/PT ID**.
Phase	Enum_PhaseID	Set equal to the user-specified **Phase**.
MonitorEnabled	BOOL	TRUE if the deviation monitor for the monitored channel is enabled.
Alert	SPS	Time-stamped health alarm indicator for the CT or PT. Alert.t shall update on the rising and falling edge of Alert.stVal.
AlertMag	REAL	Equal to the monitored channel instantaneous magnitude on the rising edge of Alert.stVal. Resets to zero on the falling edge of Alert.stVal.
PercentDeviation	REAL	Percent deviation of the monitored channel from the reference. Updates continuously.

Element Name	Data Type	Description
QualityAlert	BOOL	TRUE if the .q.validity attribute of either the monitored or reference data streams is not good.
Status	STRING(255)	Description of the status of the deviation monitor for the monitored channel.

Example 1: 4-CT Line

This example demonstrates the necessary configuration to monitor four redundant CT sets (all phases) for the following topology example.

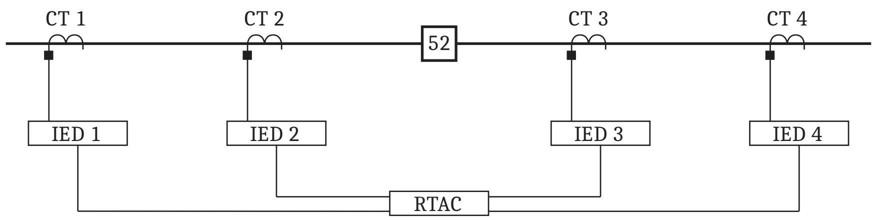


Figure 10.38 Example One-Line With Four Redundant CT Sets

Objective

Monitor health of Phase A, B, and C for CT Groups 1–4 (see *Figure 10.38*).

Assumptions

1. IED1–IED4 are SEL-421 relays.
2. The X terminal of each IED is wired to each respective CT.
3. IED data is streamed to the RTAC over the C37.118 protocol.
4. C37.118 clients are added to the RTAC project, each with a single PMU dataset.
5. C37.118 client PMUs are named SEL_421_*n*_IED*n* where *n* = 1–4.
6. The line monitored by IED1–IED4 is referred to as Line_1.
7. Minimum current to consider line active: 30 A.

Method

In this example, we will use a single monitor group from a single instance of the extension to track the health of Phases A, B, and C for all four CT sets.

Step 1. Configure four C37.118 Clients.

Step 2. Add an instance of the CtPt Monitor extension via the **Insert** menu and **Extensions** button.

Step 3. Select + on the **Monitor Groups** tab to add a new monitor group row, as shown in *Figure 10.39*.

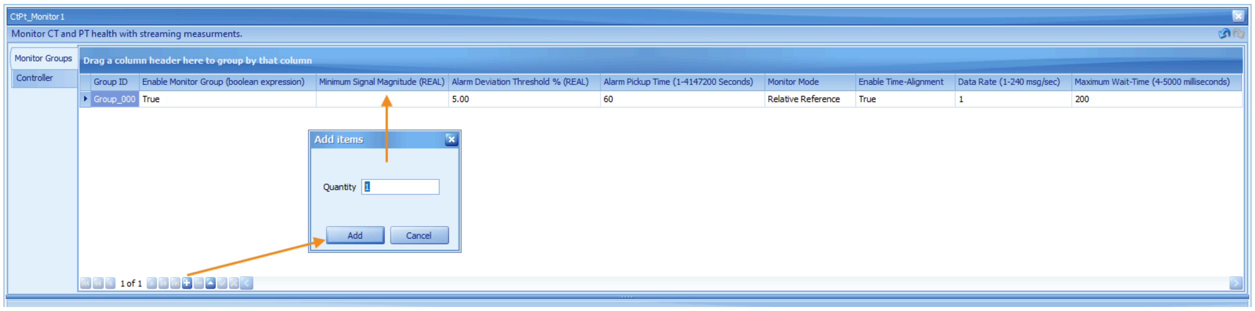


Figure 10.39 Add a Monitor Group for All CTs on the Line

Step 4. Configure the new monitor group with the following settings.

Table 10.15 CtPt Monitor Example 1 Monitor Group Settings

Setting Name	Example Setting	Notes
GroupID	Line_1_CTs	Descriptor for the redundant CT group
Enable Monitor Group	NOT (SEL_421_1_C37_118_POU.Offline OR SEL_421_2_ C37_118_POU.Offline OR SEL_421_3_ C37_118_POU.Offline OR SEL_421_4_ C37_118_POU.Offline)	Enables monitor group only if all data streams are established.
Minimum Signal Magnitude	30	
Alarm Deviation Threshold	5.00	Alarm on greater than 5% variance in the CT values
Alarm Pickup Time	60	
Monitor Mode	Relative Reference	
Enable Time-Alignment	FALSE	Time alignment is not needed since the RTAC already time-aligns all C37.118 client streams prior to data entry into the logic engine.
Data Rate	NA	
Maximum Wait-Time	NA	

Step 5. Open the child sub-item labeled Line_1_CTs, as shown in *Figure 10.40*.

Figure 10.40 Select the Child Page for the Line_1_CTs Group

Step 6. On the **Monitored Channels** tab, select the + button and add 12 rows.

Step 7. Configure the settings for the 12 rows, as shown in *Figure 10.41*.

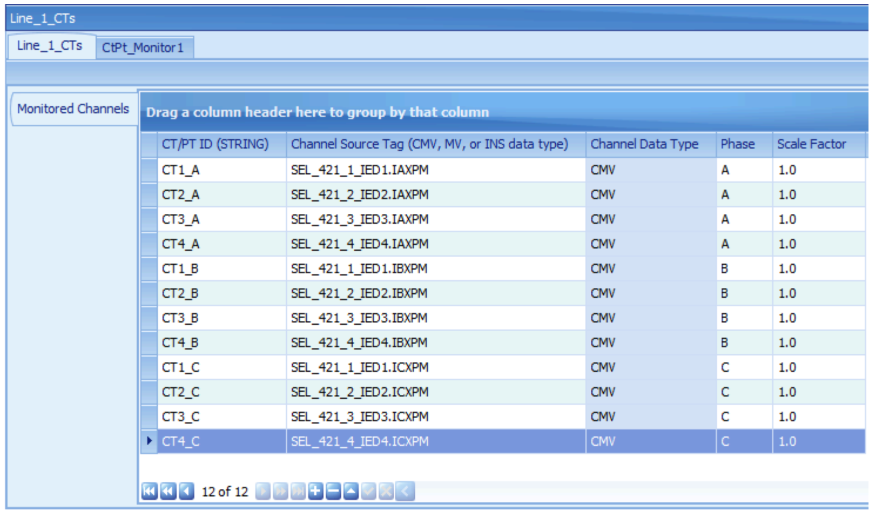


Figure 10.41 Configure Monitored Channels for Line_1_CTs Group

Each channel added to the **Monitored Channels** tab will correspond to a status structure that is available as an output of the CtPT Monitor POU.

Step 8. Map status output indicators from the POU output to the desired target. Examples include:

- ➤ Virtual Tag List for use with an RTAC HMI
- ➤ SCADA protocol map (DNP, Modbus, etc.)
- ➤ Direct reference in a Report Generator extension instance for building custom asset health reports
- ➤ Log asset status to the RTAC SOE

The POU output structure for this example, as viewed from the **Controller** tab, is shown in *Figure 10.42*.

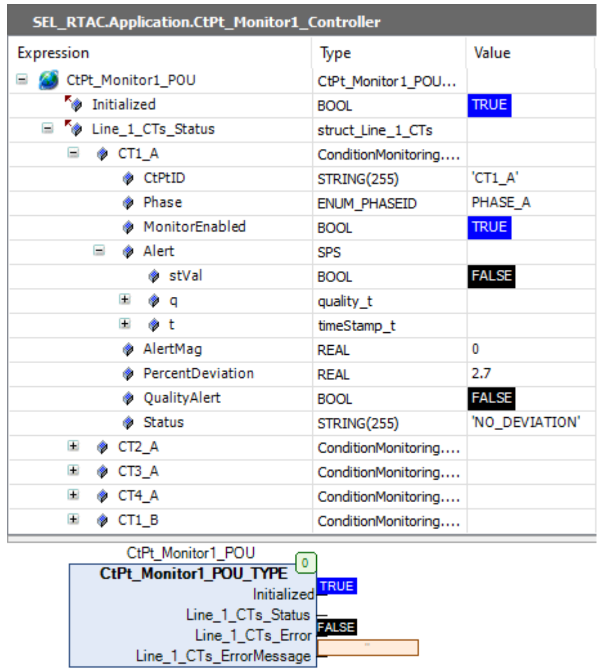


Figure 10.42 POU Output Status Indicators

Example 2: 3-CT Monitoring Group With Applied Scaling for Transformer Low-Side CT

This example demonstrates the necessary configuration to monitor two redundant CT sets (all phases) against a known-good CT set and incorporates scaling to factor out an intermediate transformer. The topology for this example is shown in *Figure 10.43*.

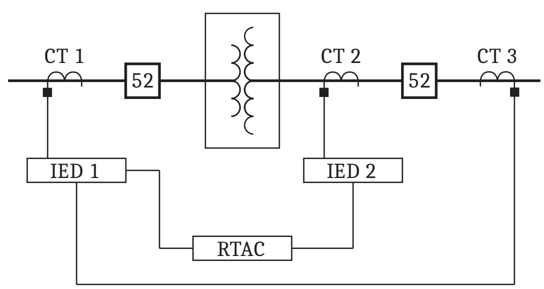


Figure 10.43 CT Monitoring Group With Applied Scaling for Transformer Low-Side CTs

Objective

Monitor health of Phase A, B, and C for CT sets CT1 and CT3, shown in *Figure 10.43*, using CT set CT2 as a defined reference.

Assumptions

1. IED1 and IED2 are SEL-421 relays that correspond to 'SEL_421_1_SEL' and 'SEL_421_2_SEL' in this example RTAC configuration.
2. CT set CT2 is used as a known-good reference against which sets CT1 and CT3 will be monitored.
3. IED data is streamed to the RTAC over SEL Fast Meter protocol at polling rate of 1 message/second.
4. The line monitored by IED1 and IED2 is referred to as Line_2.
5. Minimum (post-scaled) current to consider line active: 30 A.
6. Transformer turns ratio: 1:100. This corresponds to a current scaling ratio of 1:0.01.

Method

In this example, we will use a single monitor group from a single instance of the extension to track the health of Phases A, B, and C for CT sets 1 and 3.

Step 1. Configure two SEL clients in the RTAC project.

Step 2. Add an instance of the CtPt Monitor extension via the **Insert** menu and **Extensions** button.

Step 3. Select + on the **Monitor Groups** tab to add a new monitor group row, as shown in *Figure 10.39*. Add a single monitor group for all CTs on the line.

Step 4. Configure the new monitor group with the following settings.

Table 10.16 CtPt Monitor Example 2 Monitor Group Settings

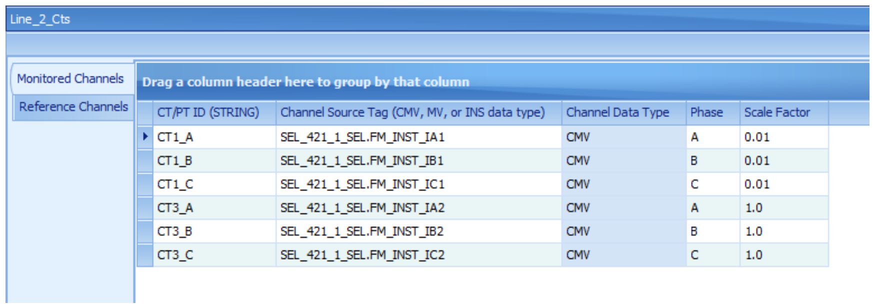
xSetting Name	Example Setting	Notes
GroupID	Line_2_CTs	Descriptor for the redundant CT group
Enable Monitor Group	NOT (SEL_421_1_SEL_POU.Offline OR SEL_421_2_ SEL_POU.Offline)	Reference to a Boolean value defined in a separate program
Minimum Signal Magnitude	30	
Alarm Deviation Threshold	5.00	Alarm on greater than 5% variance in the CT values
Alarm Pickup Time	60	
Monitor Mode	Defined Reference	
Enable Time-Alignment	TRUE	Time alignment is applied in this example to guarantee time coherence between IED1 and IED2 measurements.
Data Rate	1	samples/second
Maximum Wait-Time	3000	milliseconds

Step 5. Open the child sub-item labeled as **Line_2_CTs**, as shown in *Figure 10.44*.

Figure 10.44 Select the Child Page for the Line_2_CTs Group

Step 6. On the **Monitored Channels** tab, select the + button and add six rows.

Step 7. Configure the settings for the six rows, as shown in *Figure 10.45*.

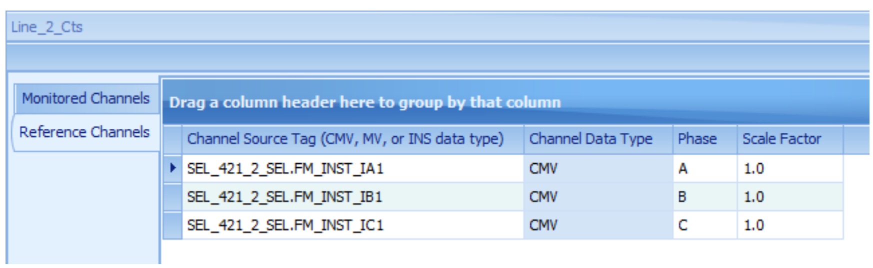


CT/PT ID (STRING)	Channel Source Tag (CMV, MV, or INS data type)	Channel Data Type	Phase	Scale Factor
CT1_A	SEL_421_1_SEL.FM_INST_IA1	CMV	A	0.01
CT1_B	SEL_421_1_SEL.FM_INST_IB1	CMV	B	0.01
CT1_C	SEL_421_1_SEL.FM_INST_IC1	CMV	C	0.01
CT3_A	SEL_421_1_SEL.FM_INST_IA2	CMV	A	1.0
CT3_B	SEL_421_1_SEL.FM_INST_IB2	CMV	B	1.0
CT3_C	SEL_421_1_SEL.FM_INST_IC2	CMV	C	1.0

Figure 10.45 Configure Monitored Channels for Line_2_CTs Group

Step 8. On the **Reference Channels** tab, select the + button and add three rows.

Step 9. Configure the settings for the three rows, as shown in *Figure 10.46*.



Channel Source Tag (CMV, MV, or INS data type)	Channel Data Type	Phase	Scale Factor
SEL_421_2_SEL.FM_INST_IA1	CMV	A	1.0
SEL_421_2_SEL.FM_INST_IB1	CMV	B	1.0
SEL_421_2_SEL.FM_INST_IC1	CMV	C	1.0

Figure 10.46 Configure Reference Channels for Line_2_CTs Group

NOTE
Only channels defined in the **Monitored Channels** tab will correspond to status structures on the POU output. Reference channels do not have a corresponding status structure.

Step 10. Map status output indicators from the POU output to the desired target, as described in *Example 1: 4-CT Line on page 584*.

Indirect Tagging

The Indirect Tagging extension automatically generates the user logic that is described in the SEL application guide *Using Shared Maps to Optimize an RTAC HMI* (AG2018-16). The purpose of this user logic is to dynamically map tag values from a single client (from a group of clients with identical tags) to a common set of data tags that can be in turn mapped to an RTAC HMI and used to animate a template screen. Only mapping the single common set of tags to the RTAC HMI (as opposed to all the tags separately for each client) for use on a template screen can reduce burden on the system and increase performance, while decreasing overall configuration time of the HMI due to the use of template screens. Note that while AG2018-16 describes building the Indirect Tagging logic in the Tag Processor, the extension will generate raw IEC 61131 logic and does not require the use of the Tag Processor.

The extension requires that the user create a two Virtual Tag lists. These two lists serve the following purposes:

- The first virtual tag list is known as the Data Map. It contains 'destination tags' that have matching tag names and tag types for each of the tags present on the templated identical client devices.

  The tags on this virtual tag list are intended to be imported to the RTAC HMI and then used to populate a template HMI diagram that is common for multiple clients of a given common IED type. This list should be assigned to the Data Map field on the Shared Tag Map tab.
- The second virtual tag list is the Indirect Tagging Control list. It contains indirect tagging "control & indication tags" that are used to select and indicate the client device whose tags are presently mapped into the Data Map shared tag list. This list should contain the following tags:
  - Per Indirect Tagging instance:
    - On the Inputs tab, a tag of type 'STR' used for the purpose of a common Device Description.

      This tag is used to provide a destination for a plain-text description Device Description Label of the presently selected client device that is mapped to the Data Map shared tags (e.g., "Feeder 5033").

      The q.validity attribute of this STR tag is automatically forced to good by the Indirect Tagging extension logic.
  - Per client device:
    - On the Inputs tab, a tag of type 'SPS' used for the purpose of a Device Selection Indication.

      This tag is set to TRUE if the client device is presently selected to map its tags into the Data Map shared tags.

      This tag can be used to configure the Status Tag field present on a RTAC HMI Indirect Tagging Link Object.

      The q.validity attribute of this SPS tag is automatically forced to good by the Indirect Tagging extension logic.
    - On the DNP Controls tab, a tag of type 'DNPC' used for the purpose of a Device Selection Control.

      This tag is used to map to a controllable object in the RTAC HMI, such as an Indirect Tagging Link.

      When the user selects this controllable object, the user logic will detect this action and re-configure the states of the device selection indication tags to map tags from the newly-selected client to the Data Map list.

      This tag can be used to configure the Output Tag Name field present on a RTAC HMI Indirect Tagging Link Object.

Shared Tag Map Tab

The Shared Tag Map tab contains a single row with global settings related to the Indirect Tagging logic generation.

- Data Map: A selection field that allows the user to choose the Virtual Tag List that contains tags with equivalent names to the tags on each of their templated devices. When the Indirect Tagging logic is executed, the tags on this virtual tag list become the 'destination tags' to which a given set of client tags are mapped.
- Device Description Tag: A field where the user enters the Device Description tag from the Indirect Tagging Control virtual tag list. It must be of type 'STR'.
- Tag Type: The detected tag type from the Device Description Tag field.
- Unresolved Client Tags: When the Indirect Tagging user logic is generated by the extension, it will attempt to map each tag it finds on the client devices configured on the Available Sources tab to a matching tag on the Data Map virtual tag list. If the mapping algorithm does not locate an appropriate match, this setting can be used to either Ignore this condition or flag an Error or Information message to the settings engineer that not all client tags were mapped.
- Process Tags: During a normal project 'save' operation, the data map and client device tag lists are not refreshed in the Indirect Tagging extension unless another settings field in the extension has changed. If changes are made to the Data Map virtual tag list or the tags on any of the client devices, toggle this field to 'True' and save your settings to force the extension to refresh these lists of tags and rebuild the IEC 61131 logic.

Available Sources Tab

The Available Sources tab can be configured with a dynamic number of rows, with each row representing a single client device and the attribute tags associated with it.

- Client: A selection field that allows the user to choose a unique client device whose tags will be automatically queried, and subsequently mapped into Data Map Virtual Tag List tags in the user logic.
- Multiple Device Client: A True or False toggle that specifies whether the client device configured in the Client field has tags that represent a single device, or whether it has tags that represent many devices that will all need to be independently mapped into the Data Map. An example of this would be a tiered RTAC system where a lower tier RTAC communicates to several IEDs and concentrates data from multiple IEDs into a DNP server map. An upper tier RTAC that communicates to this DNP server would contain a DNP client device that would be considered a Multiple Device Client because it is a single client that contains tags from multiple devices. To reference the tags from each individual client, create multiple rows on the Available Sources tab and reference the client device multiple times, specifying a unique Multi-Device Name Filter with each row.
- Multi-Device Name Filter: This is a string field where a substring is specified to uniquely divide the tags belonging to a Multiple Device Client into the individual clients that make up that list. This string is assumed to be part of the tag name and is separated by an underscore with

the 'common' portion of the tag name following it. This string should not include the underscore that separates the 'per-device' portion of the tag name from the 'common' portion. For example, a DNP client may have a set of tags available such as the following.

- MyDNPClient_DNP.Feeder1_Breaker52A
- MyDNPClient_DNP.Feeder2_Breaker52A
- MyDNPClient_DNP.Feeder3_Breaker52A

In this configuration, the user should configure three rows on the Available Sources tab, reference the MyDNPClient_DNP device for each row, and configure the Multi-Device Name Filter for each row as "Feeder1", "Feeder2", and "Feeder3". The Data Map virtual tag list should contain a single tag named Breaker52A to resolve to the tags from each of the individual clients.

- Device Description Label: This is a string field where the user can specify a per client string that is automatically copied into the Device Description tag contained in the Indirect Tagging Control virtual tag list.
- Device Selection Control Tag: This is a tag field that specifies a per-client operSPC tag from the Indirect Tagging Control virtual tag list. It must be of tag type operSPC, otherwise an error condition is generated.
- Device Selection Indication Tag: This is a tag field that specifies a per-client SPS tag from the Indirect Tagging Control virtual tag list. It must be of tag type SPS, otherwise an error condition is generated.
- Map Status From APC/INC/MDBC Tags: A True or False toggle that can be used to isolate and map only the *.status* component from tags of type APC, INC, and MDBC that also contain an *.oper* component. This is commonly used with Modbus client devices configured with Holding Register tags that are populated by Read Holding Register Polls and where the oper/writable component of the tag is not of interest (such as a metering value). If this option is enabled for a Modbus client device, any Holding Register tags of the following types will expect to find a corresponding identically named tag in the Data Map virtual tag list with the following relationships:
  - HReg Type -> Data Map Type
  - APC -> MV
  - INC -> INS
  - MDBC -> SPS

Controller Tab

The Controller tab is a placeholder for the Indirect Tagging function block to execute and does not contain any useful POU pins.

Indirect Tagging Extension Example

The following settings demonstrate an example use case for the indirection tagging extension.

Four identical SEL Protocol client devices are configured with these names:

1. SEL_351S_1_SEL
2. SEL_351S_2_SEL
3. SEL_351S_3_SEL
4. SEL_351S_4_SEL

These identical client devices are configured with the following tags enabled on the Meter and Remote Bits tabs:

Tab	Tag Name	Tag Type
Meter	FM_INST_IA	CMV
Meter	FM_INST_IB	CMV
Meter	FM_INST_IC	CMV
Meter	FM_INST_52A	SPS
Remote Bits	FO_RB_RB1	SRBC

A virtual tag list named TagControl is configured with the following tags:

Tab	Tag Name	Tag Type
Inputs	DeviceDescription	STR
Inputs	Device1_Selected	SPS
Inputs	Device2_Selected	SPS
Inputs	Device3_Selected	SPS
Inputs	Device4_Selected	SPS
DNP Controls	Select_Device1	DNPC
DNP Controls	Select_Device2	DNPC
DNP Controls	Select_Device3	DNPC
DNP Controls	Select_Device4	DNPC

A virtual tag list named HMI_Tags is configured with the following tags:

Tab	Tag Name	Tag Type
Inputs	FM_INST_IA	CMV
Inputs	FM_INST_IB	CMV
Inputs	FM_INST_IC	CMV
Inputs	FM_INST_52A	SPS
Remote Bit Controls	FO_RB_RB1	SRBC

An instance of the Indirect Tagging extension has been inserted into the project using the **Insert > Extensions** toolbar item. The Tabs on this extension are configured as follows.

Shared Tag Map tab:

Data Map	Device Description Tag	Tag Type	Unresolved Client Tags	Process Tags
HMI_Tags	TagControl.DeviceDescription	STR	Ignore	True

Available Sources tab:

Client	Multiple Device Client	Multi-Device Name Filter	Device Description Label
SEL_351S_1_SEL	False		Feeder 1001
SEL_351S_2_SEL	False		Feeder 1002
SEL_351S_3_SEL	False		Feeder 1003
SEL_351S_4_SEL	False		Feeder 1004

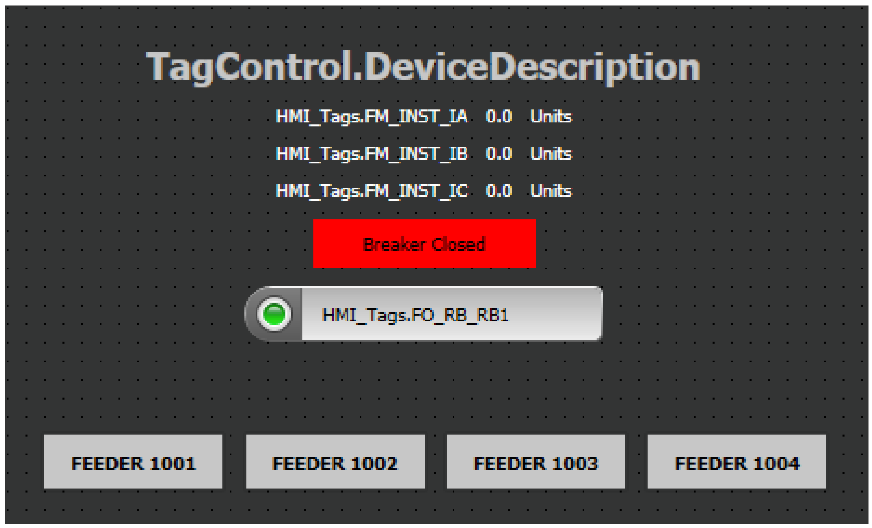
Device Selection Control Tag	Tag Type	Device Selection Indication Tag	Tag Type	Map Status From APC/INC/ MDBC Tags
TagControl.Select_Device1.operPulse	operSPC	TagControl.Device1_Selected	SPS	False
TagControl.Select_Device2.operPulse	operSPC	TagControl.Device2_Selected	SPS	False
TagControl.Select_Device3.operPulse	operSPC	TagControl.Device3_Selected	SPS	False
TagControl.Select_Device4.operPulse	operSPC	TagControl.Device4_Selected	SPS	False

When the project is saved and the Indirect Tagging Extension automatically creates the backend IEC 61131 user logic, the following statements will be generated:

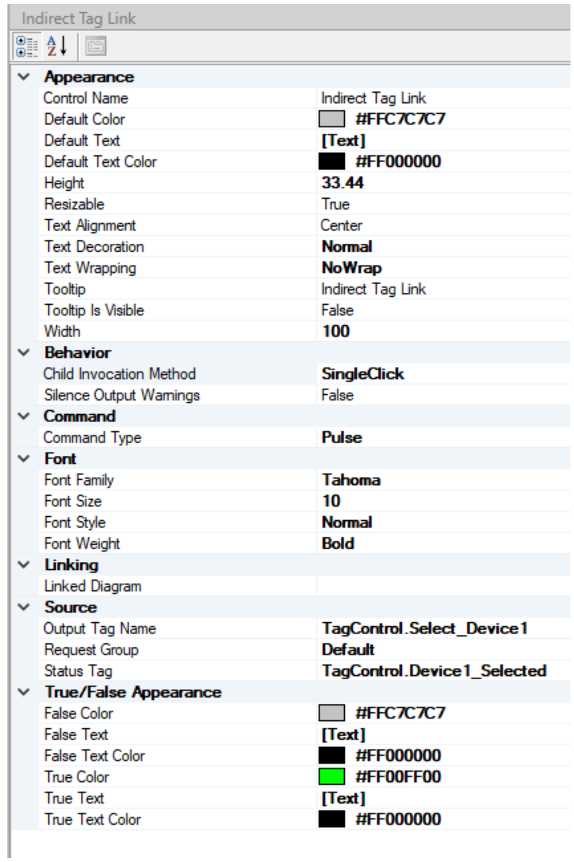
```
IF TagControl.Select_Device1.operPulse.ctlVal THEN
    TagControl.DeviceDescription.strVal := 'Feeder 1001';
    TagControl.Device1_Selected.stVal := TRUE;
    TagControl.Device2_Selected.stVal := FALSE;
    TagControl.Device3_Selected.stVal := FALSE;
    TagControl.Device4_Selected.stVal := FALSE;
ELSIF TagControl.Select_Device2.operPulse.ctlVal THEN
    TagControl.DeviceDescription.strVal := 'Feeder 1002';
    TagControl.Device1_Selected.stVal := FALSE;
    TagControl.Device2_Selected.stVal := TRUE;
    TagControl.Device3_Selected.stVal := FALSE;
    TagControl.Device4_Selected.stVal := FALSE;
ELSIF TagControl.Select_Device3.operPulse.ctlVal THEN
    TagControl.DeviceDescription.strVal := 'Feeder 1003';
    TagControl.Device1_Selected.stVal := FALSE;
    TagControl.Device2_Selected.stVal := FALSE;
    TagControl.Device3_Selected.stVal := TRUE;
    TagControl.Device4_Selected.stVal := FALSE;
ELSIF TagControl.Select_Device4.operPulse.ctlVal THEN
    TagControl.DeviceDescription.strVal := 'Feeder 1004';
    TagControl.Device1_Selected.stVal := FALSE;
    TagControl.Device2_Selected.stVal := FALSE;
    TagControl.Device3_Selected.stVal := FALSE;
    TagControl.Device4_Selected.stVal := TRUE;
END_IF
IF TagControl.Device1_Selected.stVal THEN
    SEL_351S_1_SEL.FO_RB_RB1 := HMI_Tags.FO_RB_RB1;
    HMI_Tags.FM_INST_52A := SEL_351S_1_SEL.FM_INST_52A;
    HMI_Tags.FM_INST_IA := SEL_351S_1_SEL.FM_INST_IA;
    HMI_Tags.FM_INST_IB := SEL_351S_1_SEL.FM_INST_IB;
    HMI_Tags.FM_INST_IC := SEL_351S_1_SEL.FM_INST_IC;
ELSIF TagControl.Device2_Selected.stVal THEN
    SEL_351S_2_SEL.FO_RB_RB1 := HMI_Tags.FO_RB_RB1;
    HMI_Tags.FM_INST_52A := SEL_351S_2_SEL.FM_INST_52A;
    HMI_Tags.FM_INST_IA := SEL_351S_2_SEL.FM_INST_IA;
    HMI_Tags.FM_INST_IB := SEL_351S_2_SEL.FM_INST_IB;
    HMI_Tags.FM_INST_IC := SEL_351S_2_SEL.FM_INST_IC;
ELSIF TagControl.Device3_Selected.stVal THEN
    SEL_351S_3_SEL.FO_RB_RB1 := HMI_Tags.FO_RB_RB1;
    HMI_Tags.FM_INST_52A := SEL_351S_3_SEL.FM_INST_52A;
    HMI_Tags.FM_INST_IA := SEL_351S_3_SEL.FM_INST_IA;
```

```
    HMI_Tags.FM_INST_IB := SEL_351S_3_SEL.FM_INST_IB;
    HMI_Tags.FM_INST_IC := SEL_351S_3_SEL.FM_INST_IC;
ELSIF TagControl.Device4_Selected.stVal THEN
    SEL_351S_4_SEL.FO_RB_RB1 := HMI_Tags.FO_RB_RB1;
    HMI_Tags.FM_INST_52A := SEL_351S_4_SEL.FM_INST_52A;
    HMI_Tags.FM_INST_IA := SEL_351S_4_SEL.FM_INST_IA;
    HMI_Tags.FM_INST_IB := SEL_351S_4_SEL.FM_INST_IB;
    HMI_Tags.FM_INST_IC := SEL_351S_4_SEL.FM_INST_IC;
END_IF
TagControl.DeviceDescription.q.validity := validity_t.good;
TagControl.Device1_Selected.q.validity := validity_t.good;
TagControl.Device2_Selected.q.validity := validity_t.good;
TagControl.Device3_Selected.q.validity := validity_t.good;
TagControl.Device4_Selected.q.validity := validity_t.good;
```

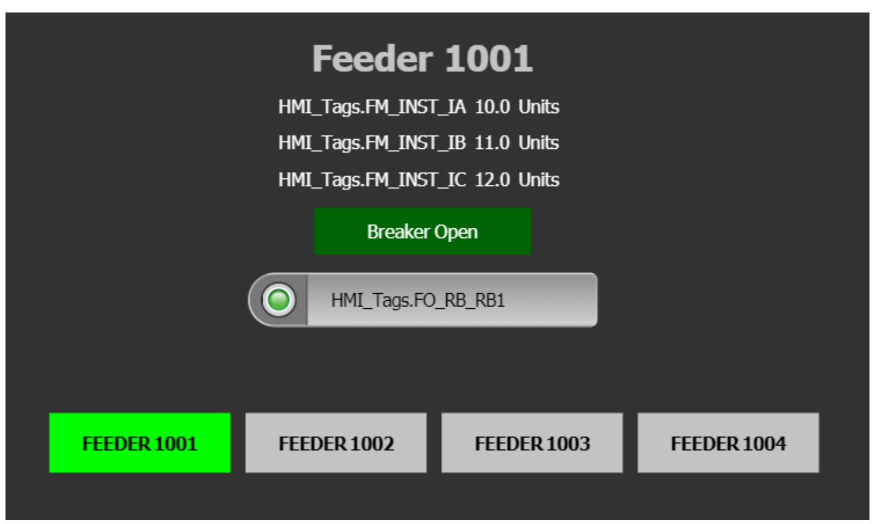
To interface the RTAC HMI with this Indirect Tagging logic, a sample Diagram Builder template screen could be configured as shown in the following figure:



The four buttons on the bottom of this template screen are Indirect Tagging Link controls configured as shown in the following figure. Note the use of the Select_Device and Device_Selected virtual tags from the TagControl virtual tag list.



In runtime, the template screen is shown in the following figure with a title at the top and the Indirect Tagging Link controls animated in green to indicate which device is presently selected for viewing on the template screen.



SECTION 11

Testing and Troubleshooting

Overview

Use the data and procedural information this section provides for aid in SEL-2240 Axion® testing and for troubleshooting performance, communication, and run-time issues. This information is applicable to all RTAC modules, the SEL-2243 Power Coupler, and all Axion I/O modules.

Self-Test

The RTAC performs a variety of startup self-tests for all Axion modules and stores the results of these tests in system tags. The RTAC also updates various run-time statistics data in system tags. See *Table 11.2* for a list of system tags. The system tags listed in *Table 11.2* represent a superset of items available in RTAC devices. Not all of the system tags apply to all RTACs. For example, the SEL-3530 RTAC has programmable LEDs and integrated inputs, represented by system tags, which do not exist in the SEL-3530-4 and SEL-2241.

POST

The RTAC performs a Power-On Self-Test (POST) at startup. You can view POST values on the Dashboard page of the web interface. *Table 11.2* lists the items the RTAC tests. These include memory, communication drivers, and hardware controllers.

Run-Time Diagnostics

The RTAC monitors CPU usage, memory usage, power, and many other system variables continuously. The system stores the values of these monitored variables in system tags and also displays some of them on the RTAC web interface **Dashboard**. You can configure system tags so that the RTAC will log these tags when they stray beyond a user-configurable boundary or state. View the resulting logs and some real-time values in the web interface. See *Section 5: Web HMI and Reports* for instructions on how to set up tags for logging. View system jumper settings and web session status from the indicators at the top of the RTAC web interface (see *Figure 11.1*).

Use ACSELERATOR RTAC SEL-5033 Software to view all real-time system tag values while you are online with the RTAC. Once you are online, click on **System Tags** in the left menu, and then click the **Tags** tab. For example, under **System Tags > Diagnostics**, the Eth_XX_Connections_Active shows how many

TCP connections are currently active on that interface on the RTAC. A single web session can have multiple TCP connections. You can also click on **Tags** in the left menu to see the total time in microseconds to complete processing for the currently loaded project (see *Figure 11.2*).

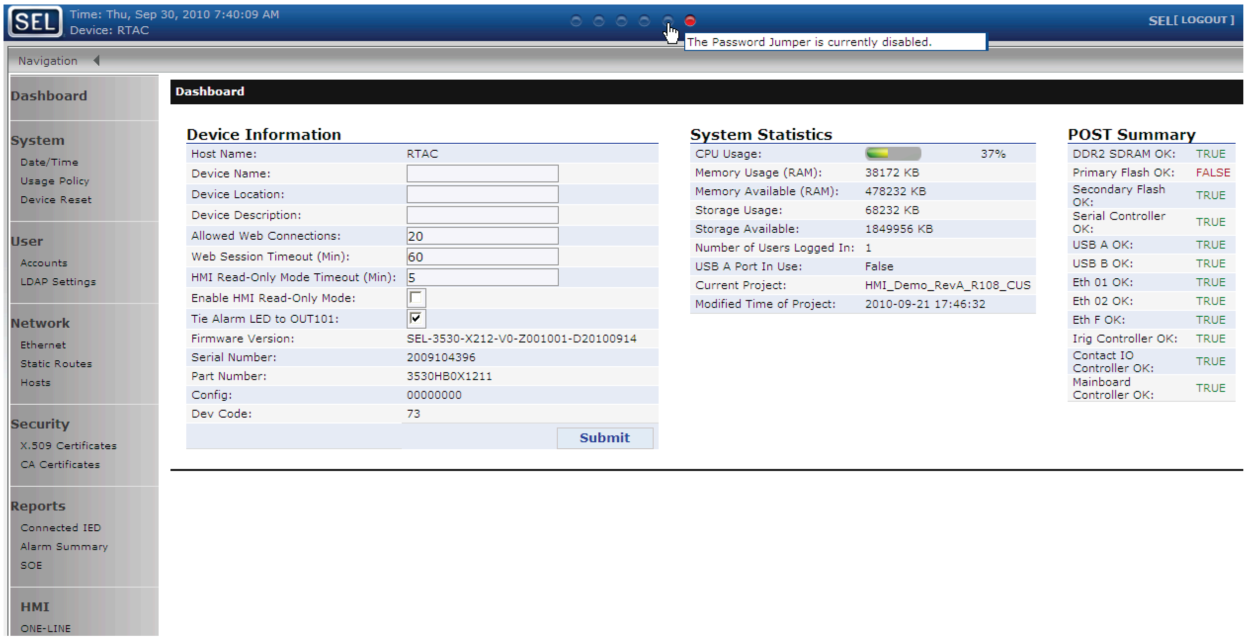


Figure 11.1 Web Interface Dashboard

NOTE

Eth_XX_Connections_Active under the Diagnostics tab does not reflect the number of users logged on. It shows all TCP connections to the RTAC. A single web session will have multiple TCP connections.

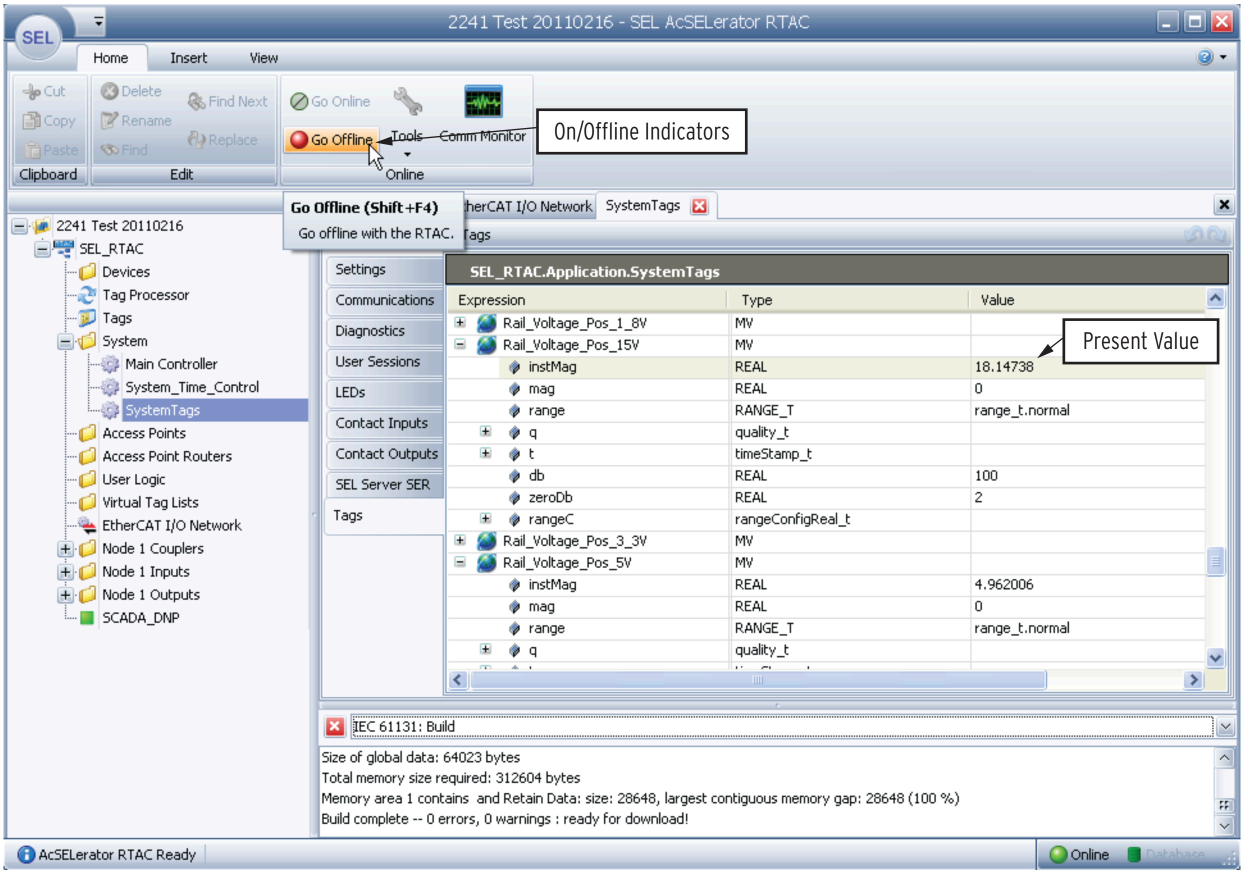


Figure 11.2 View Diagnostics Online

Power Source Scale

Use the power source scale factor to calibrate dc source voltage readings. The measured supply voltage is multiplied by the power source scale factor to populate the supply voltage system tag. The supply voltage is measured as an instantaneous value, so if the supply voltage is ac, the supply voltage system tag will continually change to reflect samples along the ac sine wave. The power source scale, therefore, is not applicable when supply voltage is from an ac source.

NOTE
Supply measurements are not rms values. Consequently, a value of 160 for a 120 Vac input is not unusual.

Using Online Debug

While online with the RTAC, you can use testing and debugging tools to force tags to static values, create tag watch windows, and change the display mode of tags.

Force Tags

Create a project in ACSELERATOR RTAC and then save and send the project to a connected RTAC. Once you have sent the project successfully, ACSELERATOR RTAC remains in the online state until you click the Go Offline icon or until the online time-out expires. This example illustrates forcing a toggle of the alarm contact on and off. While online with the RTAC, do the following:

Step 1. Click SystemTags on the left menu.

Step 2. Click **Tags** in the **System Variables and Tags** window.

NOTE
The RTAC forces tag values at the beginning of each task or program organizational unit (POU). If you force a DNP tag, for example, the force will occur directly before DNP runs. DNP can overwrite the forced value, but the RTAC refreshes the DNP tag to the forced value when the next POU runs.

Step 3. Expand the OUT101 tag by clicking .

Step 4. Expand the variables operSet and operClear.

Notice that the values are FALSE for both variables.

Step 5. Click in the **Prepared Value** field for the operSet->ctlVal line and type **TRUE**.

Step 6. Click off the field to remove focus from that field.

Step 7. Press **<F6>**.

The value will change to TRUE, and the relay will close. The **ALARM** LED also illuminates.

NOTE
The software automatically releases all forced values and goes offline with the RTAC project if no user activity is detected within the online time-out. You can change the online time-out by clicking the round SEL action button, selecting Options, and then selecting the Preferences tab.

Step 8. Click on the **Prepared Value** field again. Notice that ACSELERATOR RTAC displays a dialog box in which you can enter a new forced value. Type **FALSE** and click **OK**. Next, press **<F6>** to force ctlVal to **FALSE**.

Step 9. Click in the **Prepared Value** field for the operClear->ctlVal field and type **TRUE**.

Step 10. Press **<F6>**.

The value changes to True, the relay opens, and the **ALARM** LED extinguishes. Press **<Shift+F6>** to unforce all values.

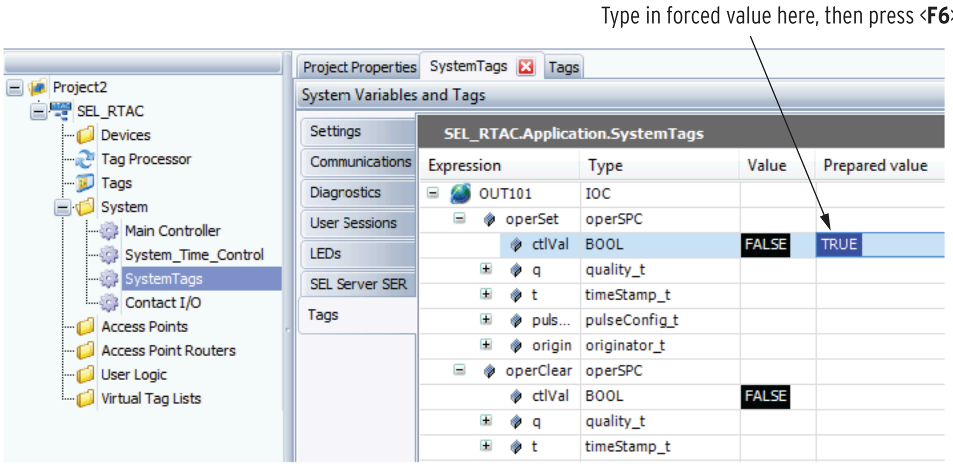


Figure 11.3 Forcing Values While Online

NOTE
Because function memory is temporary in nature, you cannot view live values or step through logic in a function as you would a program without placing a breakpoint in that function.

Debugging

Select **Tools** on the **Home** ribbon once you are online with the project. You will notice many features that allow you to set break-points in user-developed IEC 61131 logic and controls to step through the code and run to a certain point. **Clean Build**, also located on the **Home** ribbon, is used to completely rebuild the entire project. Although this is visible, your project is already rebuilt each time you save and send to the RTAC without using this feature.

Table 11.1 Advanced Debugging Tools

Function Key	Name	Description
<F5>	Start	Resume program execution after stopping at a breakpoint or run to operation.
<Shift+F8>	Stop	Stops execution of the project.
<F6>	Force Values	Force the value that you have manually entered in the Prepared Value column while online with the project.
<Shift+F6>	Unforce and Restore Values	Unforce all forced values and return them to the values they were previous to the force action.
<Ctrl+Shift+F6>	Unforce Values	Unforce all forced values but leave the forced value in place. The project can overwrite the value any time after this operation.
<F8>	Step into	Use this operation after a stop in execution (e.g., a breakpoint) to execute the next single step.
<F10>	Step over	Use this operation after a stop in execution (e.g., a breakpoint) to execute the next single line of code. Or, if the next line is a function/ function block call, execute all the code associated with that call.
<Shift+F10>	Step out	Use this operation after a stop in execution (e.g., a breakpoint) to return to the start of the application or, if executing nested routine, to the calling instruction.

Function Key	Name	Description
<F12>	Run to Cursor	Executes all instructions up to the line where the cursor is currently residing.
	Set Next Statement	Executes the line where the cursor is currently positioned.
	Show Current Statement	Brings the current position of execution back to focus.
	New Breakpoint	Adds a new breakpoint at the current position of the cursor.
<F9>	Toggle Breakpoint	Toggles enable/disable of breakpoint at the current position of the cursor.
	View Breakpoints	Provides a list of all currently defined breakpoints.
	View Callstack	Displays a list of POUs, showing relationships and current execution location.
	Watch Values	Opens watch windows in which you can view all forced tags, or only certain tags that are in watch status.
	Display Mode	Provides ability to change the format of displayed data.

Tag Cross-References

Cross references show all locations of a tag or variable within a project and indicate how the tag or variable is being used in each location. Examples of usage include declaration of the tag or variable, write operations, and read operations. You can find all locations of any tag or variable in your project by performing the following steps:

Step 1. Within a structured text program, function, or function block, right-click the tag or variable you want to cross reference.

Step 2. Select **Browse > Browse Cross References**.

The software will find all instances of the tag or variable within the project and display the results in an output window. Double-click any of the instances to navigate to the location of that tag or variable usage. By default, the scope of this search is limited to the active program. To find references elsewhere in the project, you must change the scope to include the entire project.

Force Window

You can use the **Watch Forced Values** window found on the **Tools** menu of the **Home** ribbon or press **<Ctrl+Alt+F>** to view all forced tags.

Step 1. While online with the RTAC in ACSELERATOR RTAC, force several values according to the explanation in the previous example.

Step 2. Click **Tools** in the **Home** ribbon menu.

Step 3. Select **Watch Values**, and then select **Watch Forced Values**.

The Watch all Forces pane will appear at the bottom of the window. From this window, you can view and change the values of all presently forced tags. You can also unforce selected tags or all tags, leaving the values untouched or restoring the tags to their original values. If this window is not visible, click the **Watch all Forces** tab at the bottom of the pane.

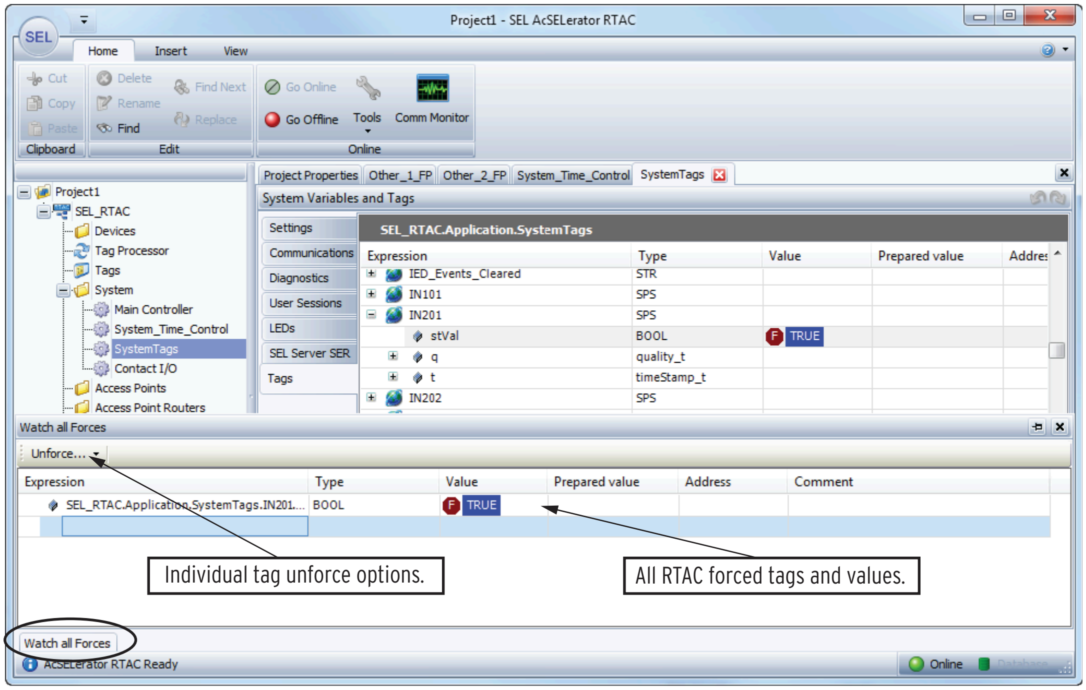


Figure 11.4 Forced Values Window

Watch Window

You can also configure as many as four watch windows to view live data of selected tags.

Step 1. From the **Online** group in the **Home** ribbon, select **Tools**. Then click on **Watch Values** and then on **Watch1** (**<Ctrl+Alt+W>**).

NOTE

The RTAC requests display data updates from its database every 200 ms.

Step 2. In the **Watch 1** pane, type in the tag names in the **Expression** column.

Step 3. You can force and view any tag in the **Watch** window(s).

Step 4. If you have more than one **Watch** pane, switch to the others by selecting the tabs at the bottom of the screen.

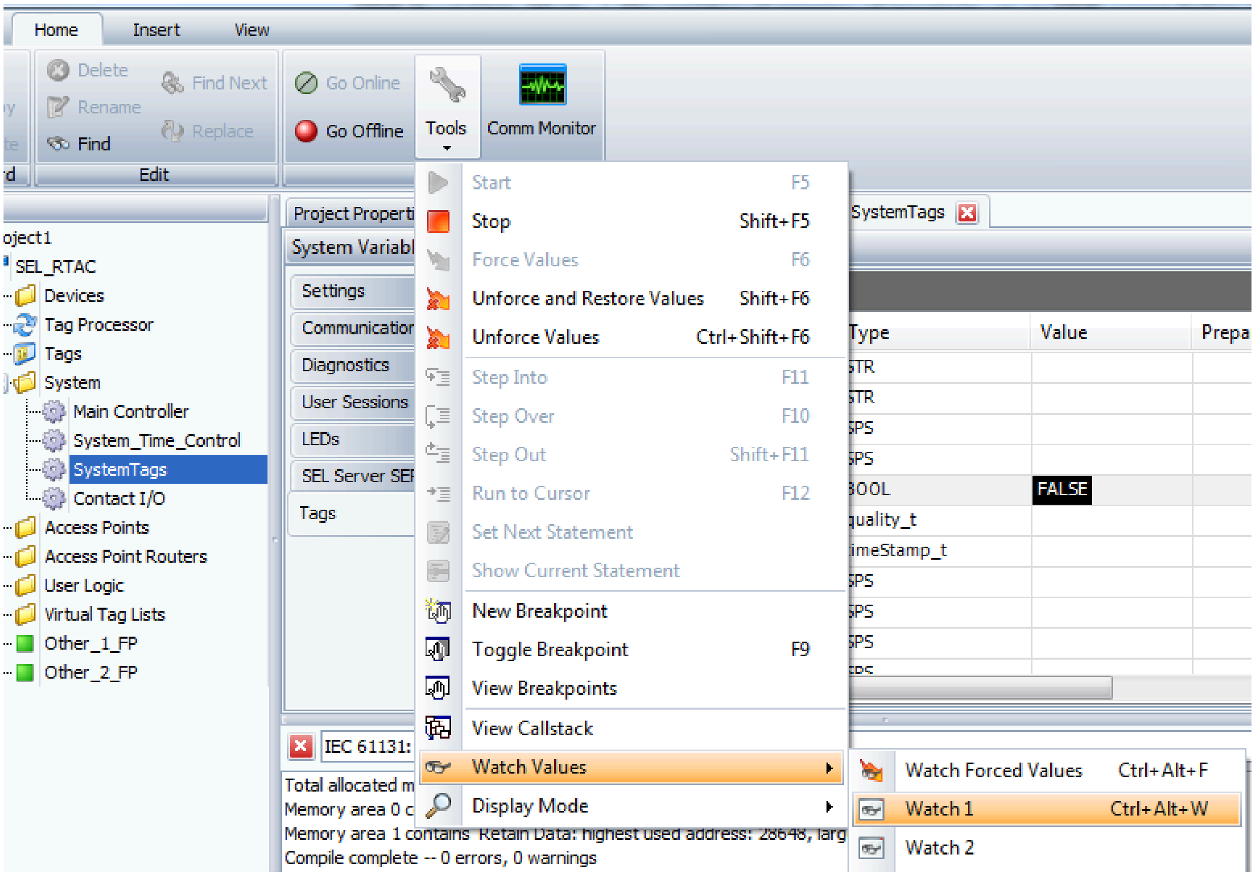


Figure 11.5 Create a Watch Window

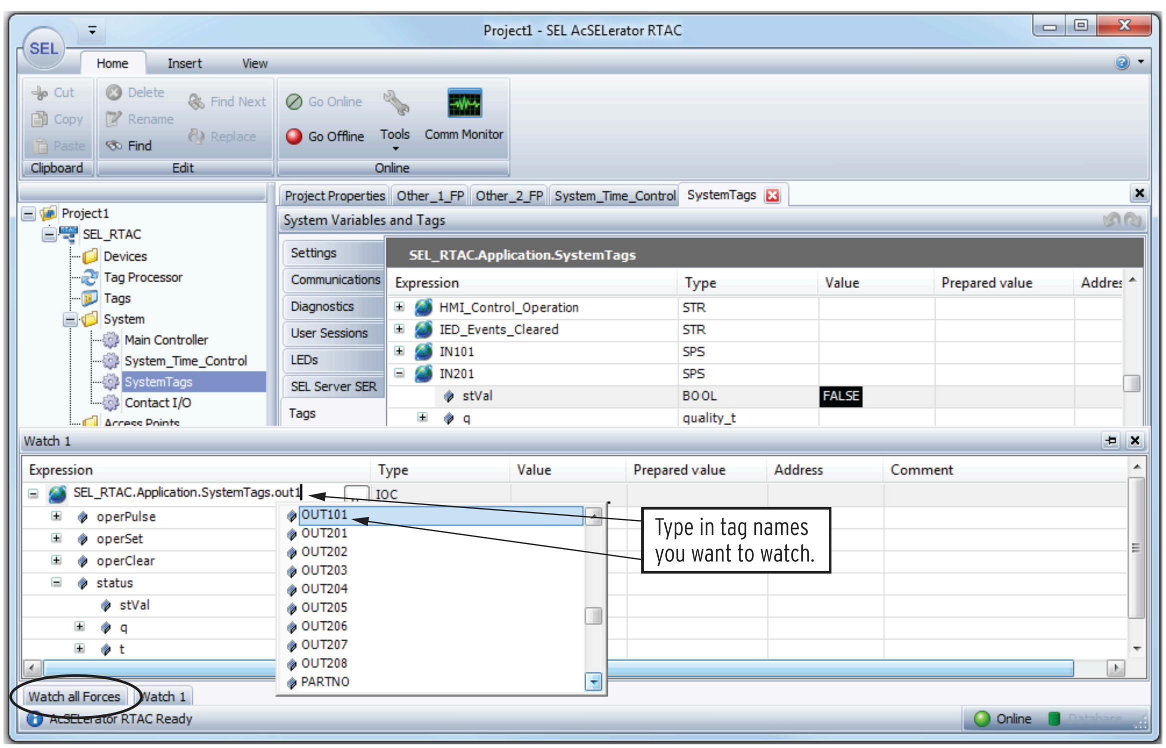


Figure 11.6 Watch Values Window

Data Display Mode

Change the online tag value data display mode to decimal, hex, or binary.

Step 1. While online, click **Tools** on the **Home** ribbon, and then select **Display Mode**.

Step 2. Select the display mode you want to use. The displayed data values will change to reflect the new setting.

Example: CPU_Burden_Percent has a value of 37 percent. The display value will change if you have selected hex or binary. Hex is represented by a preceding 16#, whereas 2# represents binary.

Watchdog

A watchdog is a type of system self-test designed to prevent errant tasks or configurations from rendering the RTAC unusable. The RTAC has five watchdog tasks.

NOTE
Find errant tasks by logging the diagnostics tag Application_Status in the Tag Processor. The RTAC will log an event every time any tasks restart.

1. **Real-Time Engine (RTE) Performance Watchdog**: Monitors run time of tasks to ensure that no task attempts to monopolize CPU resources (a program that has entered into an endless loop, for example). From the Main Controller setting tab, set the task watchdog time in ACSELERATOR RTAC. See *Figure 11.7* to set RTE Performance Watchdog. Time-out results in the following:

   - The default project replaces the existing project in the RTAC.
   - After the default project loads, ACSELERATOR RTAC posts a log message indicating expiration of the RTEWatchdog. The RTAC **ALARM** LED illuminates.
   - All contacts deassert.

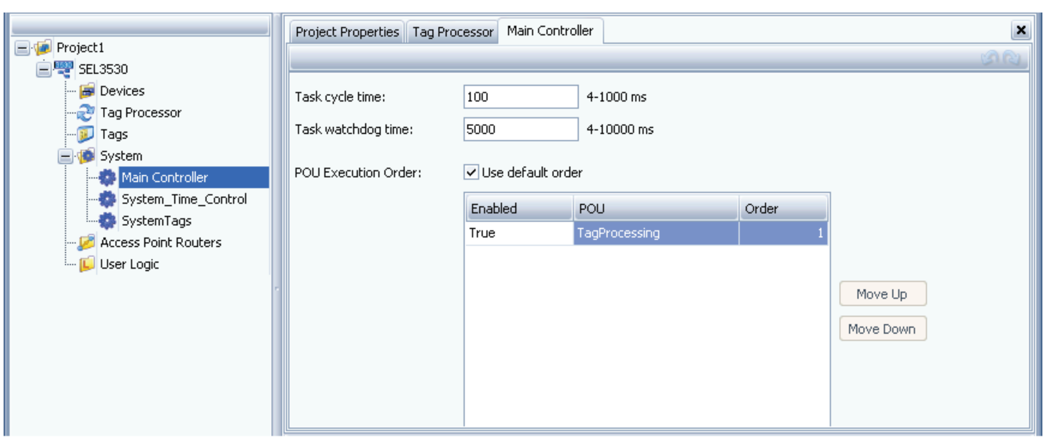


Figure 11.7 Set RTE Performance Watchdog Time-Out

2. **RTE Exception Watchdog**: Monitors task exceptions such as NULL pointer references. When it detects an exception, the RTAC will:
   - ➤ Replace the existing project with the default project.
   - ➤ Log a message indicating that the RTE experienced an exception.
   - ➤ Cause the RTAC ALARM LED to illuminate.
   - ➤ Deassert all contacts.
3. **Application Watchdog**: Monitors critical applications such as protocol tasks, date/time, etc. If a task exits unexpectedly, the RTAC takes the following steps:
   - ➤ Restarts the application.
   - ➤ Logs a configurable message indicating which application restarted.
   - ➤ Leaves the contact states or the ALARM LED unaffected.
4. **Out of Memory (OOM) Watchdog**: Monitors system for out-of-memory conditions. If the RTAC detects out-of-memory conditions, it does the following:
   - ➤ Loads default project in place of the existing one.
   - ➤ Logs an out-of-memory message when the default project is loaded.
   - ➤ Deasserts all contacts.
   - ➤ Stops all communication (serial and Ethernet).
   - ➤ Illuminates the ALARM LED.
5. **SystemWatchdog**: Detects faulted processes/hardware to avoid unstable operating conditions. If the system watchdog is not reset within 10 seconds, the RTAC takes the following steps:
   - ➤ Illuminates the RTAC ALARM LED.
   - ➤ Stops all serial and Ethernet communication.
   - ➤ Deasserts all contacts.
   - ➤ Writes to diagnostics region of RAM to indicate expiration of the system watchdog.
   - ➤ Creates a user-configurable log message after reboot. The log message indicates expiration of the System Watchdog.
   - ➤ Remains in an alarm state until the next cycling of power.
6. **Denial-of-Service (DoS) Detection**: Detects excessive activity on the Ethernet channels that could be an indication of a DoS attack. Adjusting the TaskWatchdog Time in the Main and Automation Task tabs of your project can assist in delaying these actions. If the system detects excessive Ethernet traffic, the SEL-2241 takes the following steps:
   - ➤ At 80 percent of watchdog timeout, adjusts priorities of Ethernet traffic to prevent a watchdog timeout.
   - ➤ At 80 percent of watchdog timeout, logs the message, "Impending watchdog failure; adjusting system priorities" in the Sequence of Events (SOE) log.
   - ➤ After 5 minutes of lowered priorities and Ethernet traffic returning to normal levels, Ethernet traffic priorities adjust back to normal.

- ➤ After 5 minutes of lowered priorities and Ethernet traffic returning to normal levels, logs the message "System priorities restored to normal" in the SOE log.
- ➤ If watchdog times out, follows steps described for SystemWatchdog and logs message, "System Watchdog has expired."

Multiple System Watchdog timeouts may indicate a hardware or firmware problem with the RTAC.

User Sessions

The RTAC logs run-time user session changes (log in, log off, etc.) by inserting a log entry with a configurable string. You can customize this string value in ACSELERATOR RTAC. For example, SystemTags.User_Logged_On has a default string value of "<username> Logged On". Customize the string value by performing the following steps:

Step 1. Click on the Tag Processor tab in ACSELERATOR RTAC.

Step 2. Find SystemTags.User_Logged_On in the Destination Tag Name column. If you have deleted the default logging entries, type in the tag in the first open row.

Step 3. Right-click on the heading bar and select Logging Layout.

Step 4. Scroll to the right until you find the column titled Logging On Message.

Step 5. Replace the contents with the string value you want (maximum 255 characters) enclosed in single quotes (').

Step 6. Find SystemTags.User_Logged_Off in the Destination Tag Name column. If you have deleted the default logging entries, type in the tag in the first open row.

Step 7. Replace the contents in that row of the Logging On Message with the string you want (maximum 255 characters) enclosed in single quotes (').

Step 8. Save the project and load it into the RTAC.

The Logging On Message is the message the RTAC logs when the logging condition is true. The RTAC will log the new string messages the next time someone logs on or off the RTAC web interface.

Notice in *Figure 11.9* that the Logging On Message column is next to the Tag Type column to make it easier to see. You can make navigation easier by dragging any columns in the Tag Processor grid to other positions.

The default string is the strVal of the tag. If you want to revert to using the default string, enter SystemTags.User_Logged_On.strVal for the Logging On Message.

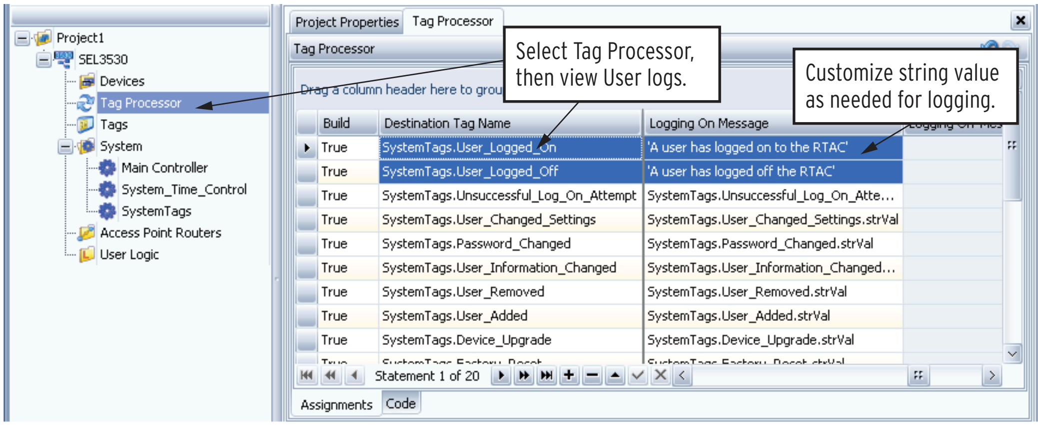


Figure 11.8 Customize Diagnostic Strings

Table 11.2 Self-Test System Tags

System Tag Name	Description	Data Type
Power		
Rail_Voltage_Pos_15V	+15 Vdc rail voltage	MV
Rail_Voltage_Pos_5V	+5 Vdc rail voltage	MV
Rail_Voltage_Neg_5V	–5 Vdc rail voltage	MV
Rail_Voltage_Pos_3_3V	+3.3 Vdc rail voltage	MV
Rail_Voltage_Neg_3_3V	–3.3 Vdc rail voltage	MV
Rail_Voltage_Neg_15V	–15 Vdc rail voltage	MV
Rail_Voltage_Pos_1_8V	+1.8 Vdc rail voltage	MV
Rail_Voltage_Pos_1_2V	+1.2 Vdc rail voltage	MV
Supply_Voltage_Neg_Gnd	DC value between negative and ground (not available in the SEL-2241)	MV
Supply_Voltage_Pos_Neg	DC value between positive and negative (not available in the SEL-2241)	MV
Supply_Voltage_Pos_Gnd	DC value between positive and ground (not available in the SEL-2241)	MV
Hardware		
Mainboard_Temperature	Temperature of main board	MV
LEDs (Also for Remaining Aux LEDs)		
Aux_LED_01.offColor	The color of the LED when operSet is FALSE and operclear is TRUE (Options are red, green, off)	colorRG_t
Aux_LED_01.onColor	The color of the LED when operSet is TRUE and operclear is FALSE (Options are red, green, off)	colorRG_t
Aux_LED_01.operClear	Set operClear.ctlVal to TRUE and operSet.ctlVal to FALSE to turn LED to offColor	operSPC
Aux_LED_01.operSet	Set operSet.ctlVal to TRUE and operClear.ctlVal to FALSE to turn LED to onColor	operSPC
Aux_LED_01.status	Current state of the LED (TRUE if onColor is TRUE, FALSE if offColor is TRUE)	SPS

System Tag Name	Description	Data Type
Power-On Self-Test (POST)		
POST_DDR2_SDRAM_OK	SDRAM configured and working	SPS
POST_Primary_Flash_OK	Primary Flash configured and working	SPS
POST_Secondary_Flash_OK	Secondary Flash configured and working	SPS
POST_Serial_Controller_OK	Serial controller configured and working	SPS
POST_USB_B_OK	USB-B port configured and working	SPS
POST_Eth_01_OK	Ethernet_01 controller configured and working	SPS
POST_Eth_02_OK	Ethernet_02 controller configured and working	SPS
POST_Contact_IO_Controller_OK	Contact I/O controller configured and working	SPS
POST_Mainboard_Controller_OK	Main board controller configured and working	SPS
System Statistics		
Application_Status	The name of any application that has restarted	STR
RTC_Battery_Voltage	Real time clock battery voltage	MV
System_Hardware_Failure	The system has detected a hardware failure, such as a RAM failure	STR
System_Watchdog_Expired	The system has reset due to expiration of the watchdog timer	STR
Out_Of_Memory_System_Reset	The system has reset due to an out of memory condition	STR
HMI_Analog_Write_Operation	A string message with the tag name of the analog output control issued by the HMI	STR
HMI_Control_Operation	A string message with the username who issued the analog output via the HMI	STR
IED_Events_Cleared	A string message with the username who deleted archived event reports	STR
Factory_Reset	Settings have been reset to factory defaults	STR
Power_Up	Unit has had a power cycle	SPS
Storage_Flash_KBytes_Used	Number of kilobytes used in Flash	INS
Storage_Flash_KBytes_Remaining	Number of unused kilobytes in Flash	INS
Storage_Flash_Total_Bad_Erase_Blocks	Number of bad erase blocks in Flash	INS
Storage_Flash_Maximum_Erase_Count	Maximum physical erase block erase count	INS
Memory_KBytes_Used	Kilobytes of memory in use	INS
Memory_KBytes_Remaining	Kilobytes of unused memory available	INS
CPU_Burden_Percent	Instantaneous CPU burden percent	INS
CPU_Burden_Percent_5_Second_Average	Five second average CPU burden percent	INS
Security and Audit		
Logs_Cleared	Indication that someone cleared the logs	STR
Device_Upgrade	Someone upgraded the firmware	STR
Settings_Change	Someone changed the settings	SPS
User_Added	Someone added a user	STR
User_Removed	Someone removed a user	STR

System Tag Name	Description	Data Type
User_Information_Changed	Someone changed user information	STR
Password_Changed	Someone changed a password	STR
User_Changed_Settings	Someone changed user settings	STR
Unsuccessful_Log_On_Attempt	A string message indicating that there were unsuccessful login attempts. The user's name is not supplied until the third consecutive attempt.	STR
User_Logged_Off	A user logged off	STR
User_Logged_On	A user logged on	STR
Number_Of_Users_Logged_On	The number of users logged on	INS
Number_Of_Files_Open	The number of files open	INS
Number_Of_Logon_Errors	The number of logon errors	INS
Disable_Password_Jumper_Enabled	The disable password jumper is enabled	SPS
System_Master_Reset_Jumper_Enabled	The system master reset jumper is enabled	SPS
Power_Up_Description	Text string indicating devices have powered up	STR
Serial_Number	RTAC serial number	STR
Communications (Also for Eth_02 and USB)		
Eth_01_Bad_Packets_Transmitted	Number of bad packets transmitted	INS
Eth_01_Bytes_Received	Number of bytes received	INS
Eth_01_Bytes_Transmitted	Number of bytes transmitted	INS
Eth_01_Collisions	Number of collisions detected	INS
Eth_01_Connections_Active	Number of active connections	INS
Eth_01_Link	Link is detected	SPS
Eth_01_Packets_Received	Number of packets received	INS
Eth_01_Packets_Transmitted	Number of packets transmitted	INS
Eth_01_Ports_Active	Number of TCP ports active	INS
Eth_01_Received_Packets_Dropped	Number of received packets dropped	INS
Eth_01_Transmitted_Packets_Dropped	Total transmitted packets that were dropped	INS
EtherCAT		
ECAT_ERROR	SEL-2241 embedded EtherCAT controller error code	STR
ECAT_STATUS	SEL-2241 embedded EtherCAT controller status code	STR
RAID		
RAID_Error_Message	RAID Error Message	STR
RAID_KBytes_Capacity	Capacity of the RAID in kilobytes, or 0 if RAID has not been created	INS
RAID_KBytes_Remaining	The RAID amount of disk space remaining	INS
RAID_KBytes_Used	The RAID amount of disk space consumed	INS
RAID_Progress_Percent	Progress of the RAID operation (0–100%)	INS
RAID_Status	Status of the RAID (Initialization, Ready, Checking, Degraded, Rebuilding, Failed)	STR

System Tag Name	Description	Data Type
RAID_Status_Message	Information about the new status of RAID	STR
SATA_<n>_Connected	Device exists on SATA <n>, where <n> is 01 through 04	SPS
SATA_<n>_KBytes_Capacity	Capacity of the device attached to SATA <n>, where <n> is 01 through 04	INS
SATA_<n>_RAID_Status	RAID status of the device attached to SATA <n>, where <n> is 01 through 04	STR
SATA_<n>_System	Device exists on SATA <n> and is the system boot disk, where <n> is 01 through 04	STR

Troubleshooting

Use this table to identify possible solutions to RTAC problems you are having.

Problem	Possible Cause	Solution
Front-panel ENABLED LED is dark	Input power not present	Verify input power is present. Check fuse.
	Invalid logic project installed	Reload a known good project with ACSELERATOR RTAC software.
PC does not recognize USB-B as an Ethernet port	Missing USB Ethernet driver	Download and install the SEL USB Ethernet driver SELusbGadget.inf
	PC failed to reinitialize driver	Reboot PC and retry.
RTAC does not respond to a device connected to Ethernet (ETH F, ETH 1, or ETH 2)	Incorrect IP address	Verify IP address.
	Invalid subnet	Verify subnet and submask.
	Incorrect or disconnected cable	Verify cable is cross-over if needed.
	Ethernet port disabled	Verify settings via web interface.
	More than one port configured for the same subnet	Use different subnet for each Ethernet port on the RTAC.
RTAC does not work correctly with device connected to a serial port	Cable not connected	Verify cable is secure.
	Cable is incorrect type	Verify the cable pinout and type needed.
	The RTAC and other device have mismatched communications parameters	Verify communications parameters.
	One device has received an XOFF, halting communications	Verify XON/XOFF are not used unless needed.
RTAC does not synchronize with IRIG-B	Configuration incorrect	Verify configuration in the ACSELERATOR RTAC project, and reload into the RTAC.
	Cable disconnected	Verify cable is secure.
Unable to download a project with ACSELERATOR RTAC	Version mismatch between RTAC firmware and PC software	Verify versions. Use web interface with RTAC and About ACSELERATOR on PC software.

Problem	Possible Cause	Solution
Unable to go online with project, or other network related issues are occurring	Certain needed network ports are closed. The software needs the following ports to operate correctly:	Enable use of these ports by either turning off or adding exceptions to the firewall and antivirus software.
	5432 (TCP Incoming): ACSELERATOR RTAC project download/retrieval	
	1217 TCP Incoming): ACSELERATOR RTAC online status monitoring and point forcing	
	5433 (TCP Incoming): RTAC firmware upgrades	
	15453 (TCP Incoming): Communications Traffic Monitoring for diagnostics/testing	
	443 (TCP Incoming): HTTPS (RTAC web server)	
	80 (TCP Incoming): HTTP (RTAC web server, automatically redirects to port 443)	
Password not accepted in web interface and during project download	Incorrect password	Use correct password. Override login and reset if needed.
Front-panel **ENABLED** LED is dark, **ALARM** LED is illuminated	RTAC project has an invalid project	Normally, the RTAC will reset an invalid project to the factory-default project. If the RTAC remains in this disabled state, you can reset to factory settings.
ALARM LED is illuminated on all Axion modules	EtherCAT protocol not enabled in RTAC	Verify intended project was loaded in the RTAC.
	EtherCAT error	Use Online mode in ACSELERATOR RTAC to determine root cause of the error (see examples in this section).
All module LEDs in an Axion node are dark	Input power is not present	Verify input power is present.

Troubleshooting an EtherCAT Network

NOTE
Always ensure that power couplers are not connected to any Ethernet switches or other Ethernet devices, including RJ45-to-fiber-optic converters. The power coupler EtherCAT ports are only for direct connections between power couplers of the same type or to an RTAC EtherCAT port. Connecting a power coupler to any other Ethernet device can cause the EtherCAT network to stop communicating completely or have unpredictable behavior.

ACSELERATOR RTAC provides a number of online diagnostics to assist troubleshooting in case of an EtherCAT network error. While the Axion is designed and tested to operate reliably over many years, the most common sources of network problems you will experience will be either a network cable failure between nodes or component failure in an I/O module. Recall from *Controlling EtherCAT Response to Network Errors on page 228* that you can select one of two automatic responses to a network error: 1) immediate shutdown of the network, or 2) "ride-through" operation where the Axion attempts to continue operation with all available modules.

We will use the sample network shown in *Figure 11.9* to illustrate the basic troubleshooting steps necessary to find root cause of a failure in an Axion system. The sample system includes four Axion nodes with an SEL-2241 RTAC module installed in Node 1. The diagram is simplified to only show the RTAC and power couplers, but each node also has a number of I/O modules.

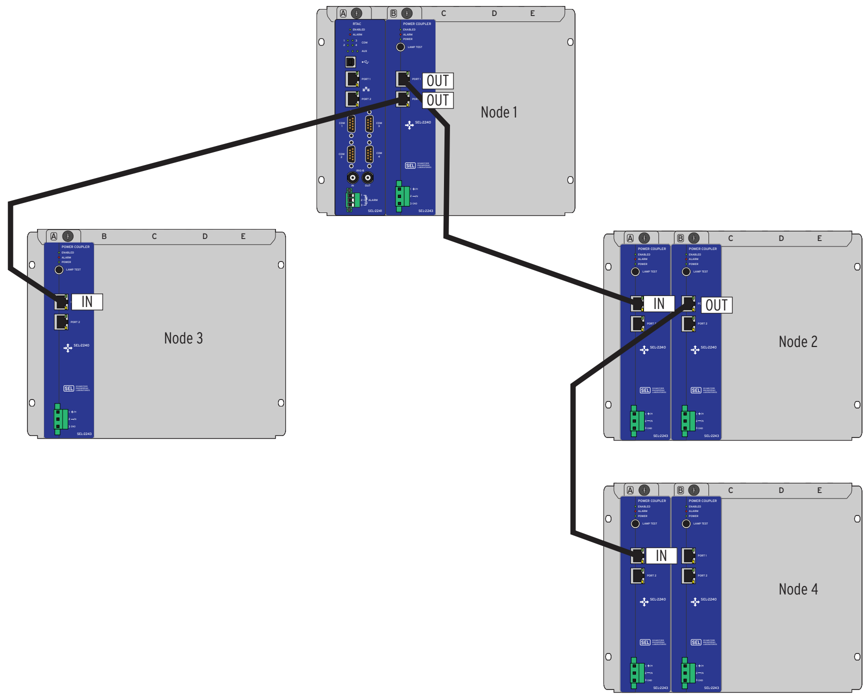


Figure 11.9 Sample Axion System for Troubleshooting Illustrations

The SEL-2243 Power Coupler provides two Ethernet ports dedicated for EtherCAT. These ports are labeled **PORT 1** and **PORT 2** respectively. For the following figures, we have used IN and OUT labels to denote the ports operating in the IN mode and OUT mode for EtherCAT. An IN port is the port by which a node is connected to an upstream node or an external RTAC and is designated in the figures by a label of IN. **PORT 2** in a power coupler never operates as an **IN** port. **OUT** ports make a connection to a downstream node and are designated in the figures by a label of OUT. When you use an SEL-2241 RTAC rather than an external RTAC, the first power coupler in the node containing the RTAC uses the backplane connector as an **IN** port for EtherCAT. Therefore, you can use both ports on the first coupler in OUT mode.

Example 1: Configuration Error

As described in *Node Connections Editor on page 227*, the RTAC will automatically initialize, configure, and start the EtherCAT network after you load a project from the ACSELERATOR RTAC software. To accomplish this, the ACSELERATOR project must include specific information about the placement of modules within an SEL-2240 node and the network connections between nodes. This example illustrates the network errors you would receive in case a physical I/O module does not match the module listed in the software project. Referring back to *Figure 11.10*, in slot **H** of Node 1 we installed a digital output module that had all Form A contacts. However, in the project, the module was configured to have eight Form A and eight Form B contacts. The remainder of the project and physical network were matched. *Figure 11.11* shows the properties for the erroneous module.

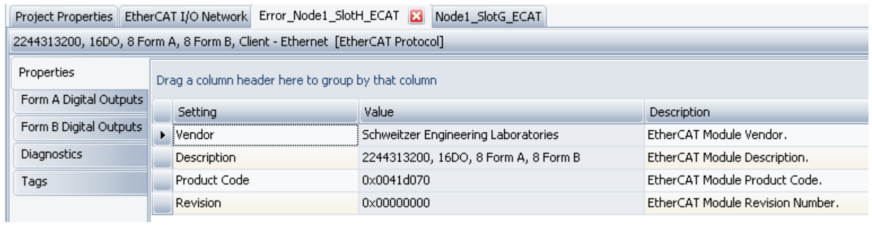


Figure 11.10 Network Error Illustration for Example 1

In online mode, the node connections editor (Connections tab) provides a visual summary of the network state. As you can see in *Figure 11.11*, a white-on-red "X" will display next to any module that is not operational on the network. During network initialization, all modules will have such an indicator. The **ALARM** LED on the modules will be illuminated, as well.

The RTAC initializes EtherCAT after it is enabled, and the run-time engine is started. Be aware that other protocol connections may be fully enabled before EtherCAT starts running. During EtherCAT startup, ACSELERATOR RTAC (in online mode) will display startup status at the bottom of the software window in the status bar. *Figure 11.11* shows the initial message during this process "Beginning EtherCAT boot sequence...".

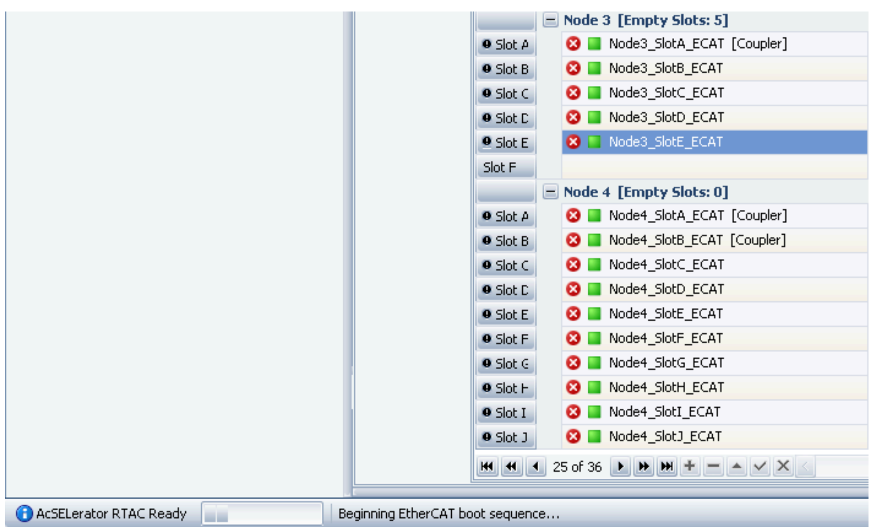


Figure 11.11 Network Initialization

Figure 11.12 shows the status messages for the remaining portions of a successful configuration process. Even though this process may take a number of minutes, you can be confident that the procedure is progressing successfully as long as the status bar does not display an error message. No user interaction is needed during the configuration process. The **ENABLED** LED on the modules will illuminate once the network is operating.

Figure 11.12 Network Configuration Status Messages

For this example, the RTAC will recognize the module error during the Addressing phase of network initialization. *Figure 11.13* illustrates the error indicators in ACSELERATOR RTAC. In the node connections editor Connections tab, notice that slots **H** and **I** both have an error indication (there was no module installed in Slot **J**); the rest of the modules in Node 1 do not display an error. Also, the status bar provides the following message, "EtherCAT server alias program error during Client State 2." Network aliasing is part of the automatic EtherCAT network addressing process. This error message indicates that at least one module on the network does not match the project. The RTAC was unable to finish configuration and startup of the network. Depending on the location of the erroneous module, other nodes in the network may display error indicators for all modules. The RTAC stops addressing the network once an error occurs, so all unaddressed modules will continue to display the error indicator.

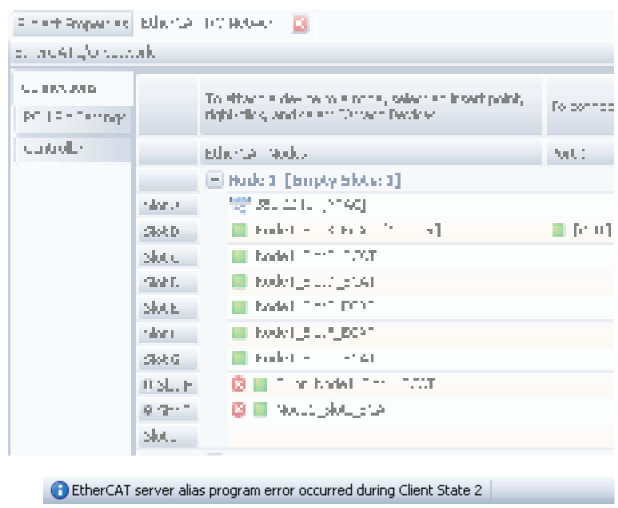


Figure 11.13 Configuration Error Because of Mismatched Module

When troubleshooting an error similar to this case, the first module in a node with an error indication is likely the cause of the problem.

NOTE
ACSELERATOR RTAC would display exactly the same error indicators if the two modules in slots H and I were installed in reverse order.

Use the following steps to troubleshoot the problem:

Step 1. Visually verify that the part number of the installed module matches the ACSELERATOR RTAC project. In this case we would see that the module and project do not match. Once the project is modified and downloaded, the network would initialize normally. If there were more than one error, ACSELERATOR RTAC would display a different error location the second time the project is loaded.

Step 2. If you do not find a part number mismatch, verify that the module is properly inserted into the node and has power by checking that the **ALARM** LED is illuminated.

Step 3. Verify that all internode network connections match the project settings (listed in the Node Connections Editor).

Step 4. If the problem persists, replace the module and attempt to restart EtherCAT (refer to *Replacing a Failed Module* in the *SEL-2240 Instruction Manual*).

In this example the server alias program error was caused by a module mismatch. You would receive the same error code if you connected the wrong Ethernet port on an external SEL-3530 or SEL-3530-4 to an Axion node. For example, if you set Ethernet Port 1 on an SEL-3530 as the dedicated EtherCAT port, but then physically connect to Ethernet Port 2, the system will report a server alias program error because the RTAC module would not be able to communicate with the Axion I/O modules.

Example 2: Cable Failure to a Terminal Node

The next two examples describe the symptoms and troubleshooting for internode network cable failures. As illustrated in *Figure 11.14*, we will first look at a cable failure between Node 1 and Node 3.

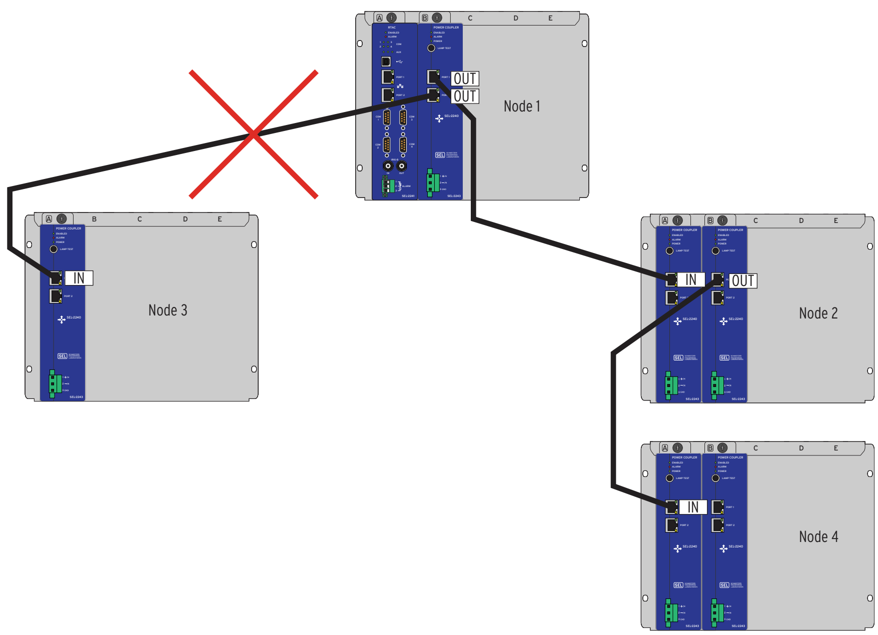


Figure 11.14 Network Cable Fault

If this cable becomes disconnected or broken, online you would see the network status shown in *Figure 11.15*. Nodes 1, 2, and 4 would not display an error, but all modules in Node 3 would have an error indication. Additionally, the status bar would display the message "EtherCAT network failed."

If you have the EtherCAT POU pin "Disable_On_Network_Error" set to TRUE (default value), then all EtherCAT control messages will stop as soon as the RTAC recognizes that a network fault has occurred. All outputs will deassert and the **ALARM** LEDs on all modules will illuminate. If the "Disable_On_Network_Error" pin is set to FALSE, the RTAC will continue to read and write data to all available modules (Nodes 1, 2, and 4). Only the **ALARM** LEDs on the Node 3 modules would be illuminated.

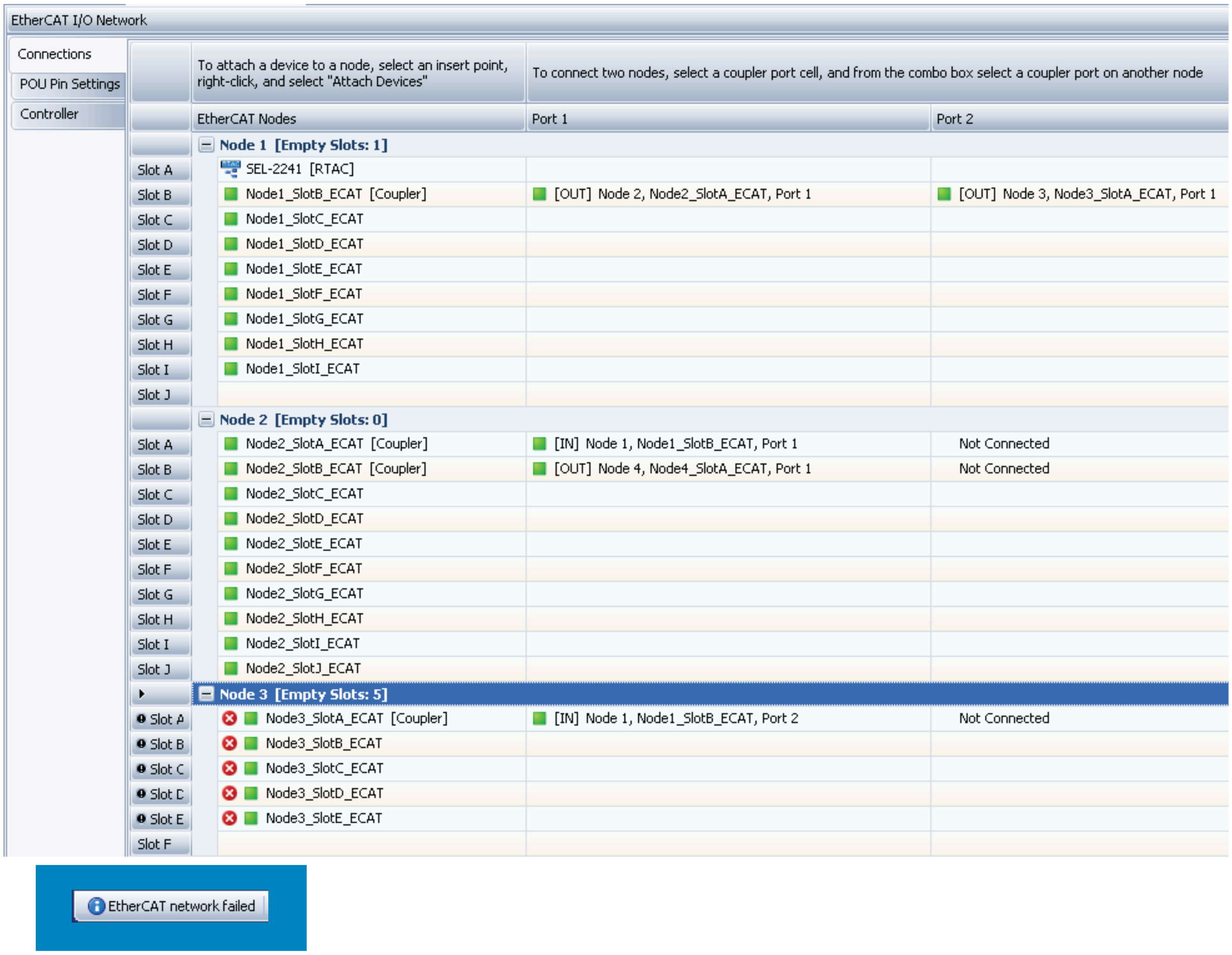


Figure 11.15 Connections Status for Network Cable Fault Between Node 1 and Node 3

Additionally, the individual modules have diagnostics that can be helpful for continuous monitoring and troubleshooting, as described in *Common Diagnostic Tags on page 161*. Referring to *Figure 11.16*, the status of modules in Node 3 will become "Non-Operational." Monitoring of module status and tag quality will provide immediate indication of a network problem.

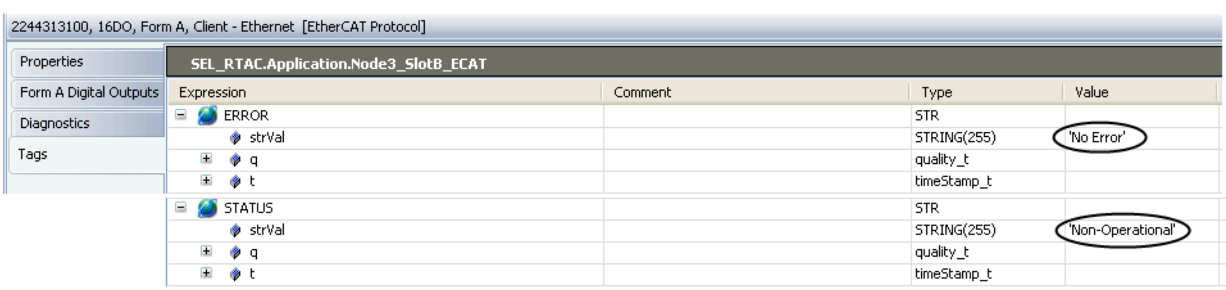


Figure 11.16 Module Status in Node 3

The ERROR tag shown in *Figure 11.16* is "No Error"; which the RTAC reads from the module. Because none of the modules in Node 3 are available on the network once the cable is disconnected, no updated value can be read. The tag retains the last value. The combination of the STATUS and ERROR tags should be used for monitoring. *Figure 11.17* shows the same diagnostics for a module in one of the unfaulted nodes. In the figure you can see that the STATUS tag has the value of "Operational" and the ERROR tag has the value Cable Error. The modules in Nodes 1, 2, and 4 would all have these values. All of these diagnostics agree with the information we saw in the node connection editor; the error is coming from Node 3 or the cable between Node 1 and Node 3.

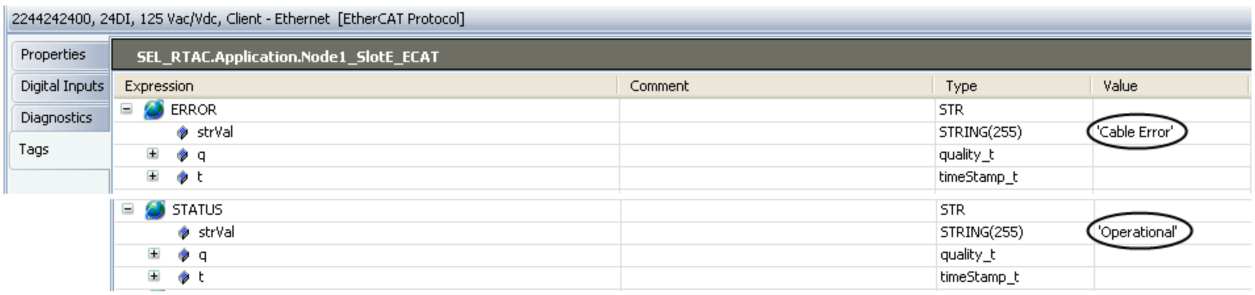


Figure 11.17 Module Status in Unfaulted Nodes

Step 1. At Node 3, verify that the incoming network cable is properly connected to the IN port on the power coupler.

Step 2. At Node 3, verify that the node has incoming power by checking LED status.

Step 3. At Node 1, verify that the outgoing network cable is properly connected to the **OUT** port on the power coupler.

Step 4. Verify continuity in the network cable between Node 1 and Node 3.

Step 5. If needed, verify that all internode network connections match the project settings (listed in the Node Connections Editor).

Step 6. If the "Disable_On_Network_Error" POU pin is FALSE, the network will automatically start in Node 3 once communications return. If the pin is TRUE, manually restart EtherCAT as described in *Replacing a Failed Module With Disabled Network (Disable_On_Network_Error = TRUE)* in the *SEL-2240 Instruction Manual*.

Example 3: Cable Failure to a Node With Downstream Connections

This example also evaluates a cable failure. In this case, the problem is between Node 1 and Node 2. As we will see, the symptoms will be different from Example 2 because Node 2 has an OUT connection to Node 4.

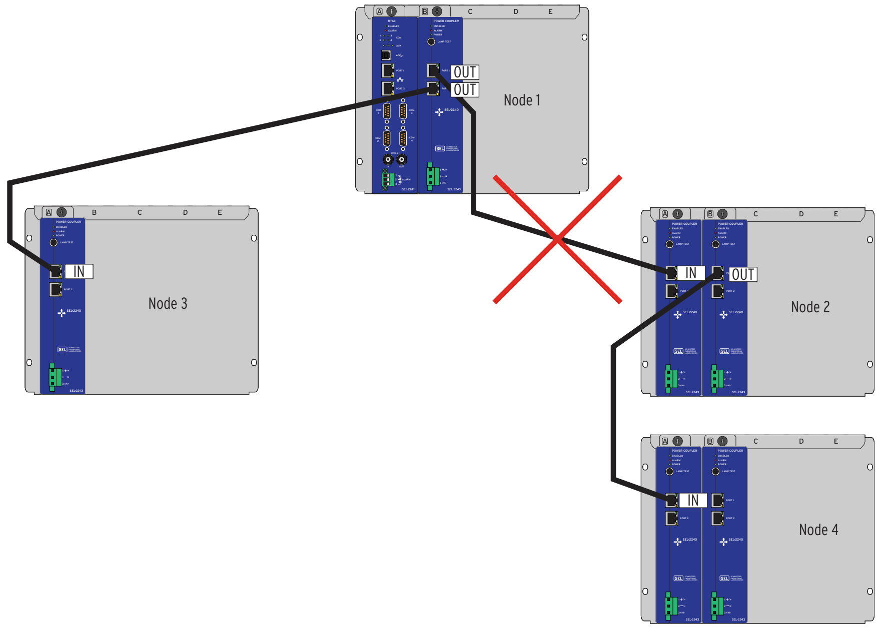


Figure 11.18 Network Cable Fault

As you can see from *Figure 11.19*, the cable fault results in error indications for all of the modules in Node 2 and Node 4. Node 1 and Node 3 would report "Operational" modules and a "Cable Error." The node connections editor in *Figure 11.19* shows that the power coupler in Slot B of Node 2 provides connectivity for Node 4. Using the rule of thumb that the first module with an error is likely the cause, then we would troubleshoot the problem by assuming that the error is coming from Node 2 or the cable between Node 1 and Node 2.

Step 1. At Node 2, verify that the incoming network cable is properly connected to the IN port on the power coupler.

Step 2. At Node 2, verify that the node has incoming power by checking LED status.

Step 3. At Node 1, verify that the outgoing network cable is properly connected to the OUT port on the power coupler.

Step 4. Verify continuity in the network cable between Node 1 and Node 2.

Step 5. If needed, verify that all internode network connections match the project settings (listed in the Node Connections Editor).

Step 6. If the "Disable_On_Network_Error" POU pin is FALSE, the network will automatically start in nodes 2 and 4 once communications return. If the pin is TRUE, manually restart EtherCAT as described in *Replacing a Failed Module With Disabled Network (Disable_On_Network_Error = TRUE)* in the *SEL-2240 Instruction Manual*.

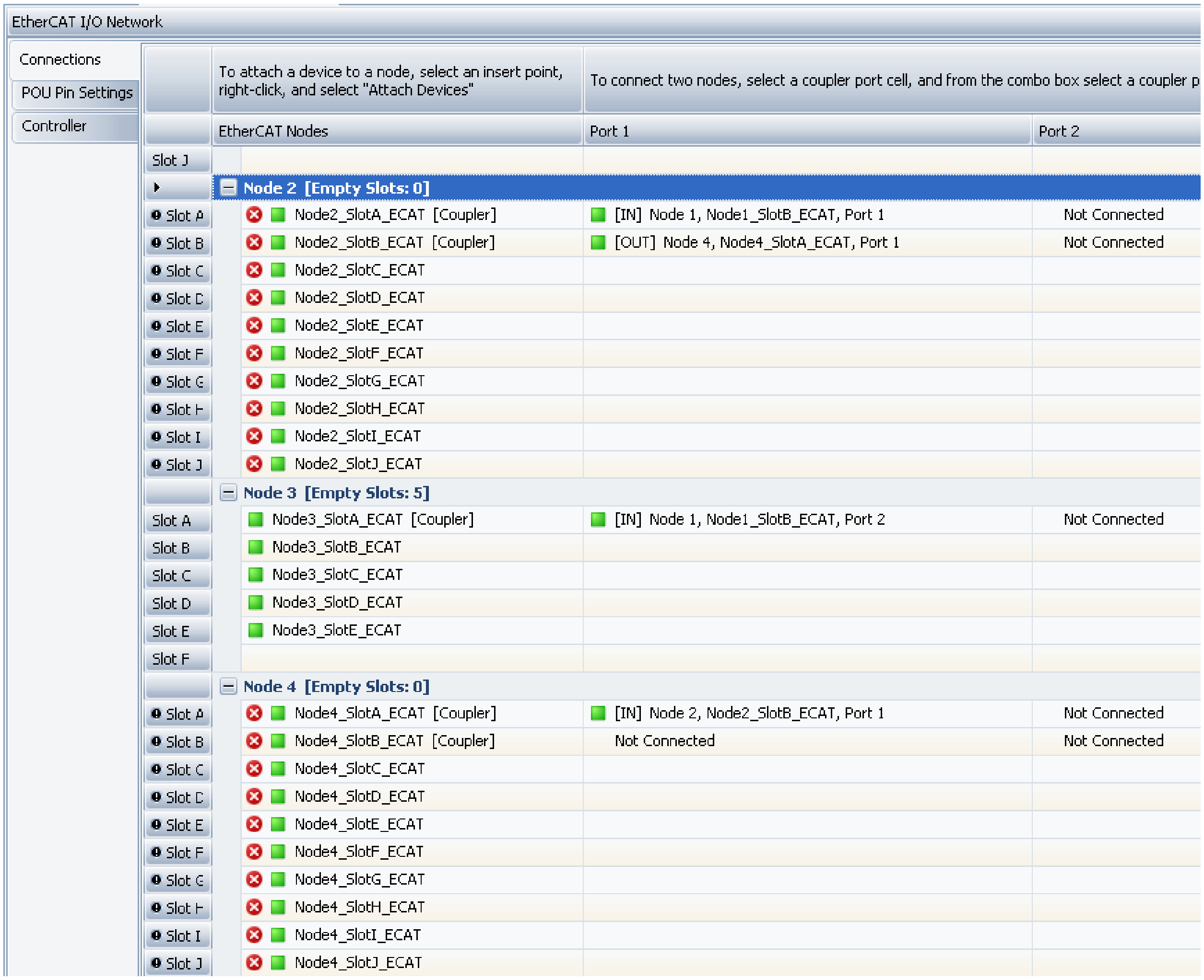


Figure 11.19 Connections Status for Network Cable Fault Between Node 1 and Node 2

Example 4: Module Failure

As a final example, we will evaluate an individual module failure in a network. *Figure 11.20* shows the connection status if the module in Slot C of Node 3 stops functioning. Unlike the network status in Example 1, no modules in any other node have an error indication. Using the troubleshooting rule of thumb, we would start troubleshooting the module in Slot C.

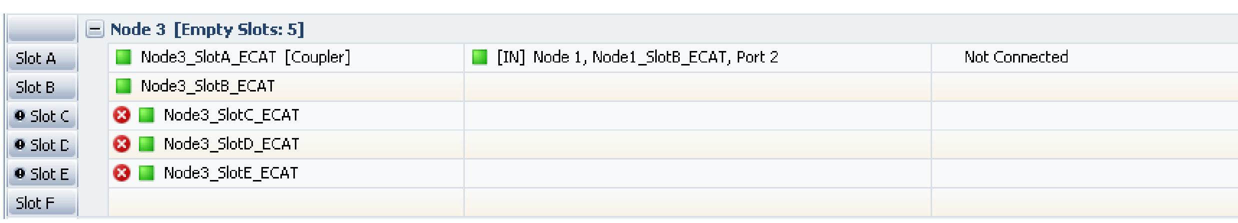


Figure 11.20 Connections Status for an Individual Module Failure

Step 1. Verify that the module is properly inserted into the node and has power by checking that the **ALARM** LED is illuminated.

Step 2. If the problem persists, replace the module and attempt to restart EtherCAT (refer to *Replacing a Failed Module* in the *SEL-2240 Instruction Manual*).

Step 3. If needed, verify that all internode network connections match the project settings (listed in the Node Connections Editor).

This page intentionally left blank

APPENDIX A

Firmware and Manual Versions

Firmware

Determining the Firmware Version

To find the firmware version number in your SEL Real-Time Automation Controller (RTAC) family of products, including the SEL-2240 Axion®, log in to the RTAC web interface and find Firmware Version on the Dashboard display page.

The firmware version will be either a standard release or a point release. A standard release adds new functionality to the firmware beyond the specifications of the existing version. A point release is reserved for modifying firmware functionality to conform to the specifications of the existing version.

A standard release is identified by a change in the R-number of the device FID number.

Existing firmware:

FID=SEL-2241-**R100**-V0

Standard release firmware:

FID=SEL-2241-**R101**-V0

A point release is identified by a change in the V-number of the device FID string.

Existing firmware:

FID=SEL-2241-R100-**V0**

Point release firmware:

FID=SEL-2241-R100-**V1**

Revision History

Table A.1–Table A.8 lists the firmware versions, revision descriptions, and corresponding instruction manual date codes. In the Affected Models column, "All" indicates that the revision note applies to all models and FID numbers listed in the Firmware, Software, and Manual Versions column. The most recent firmware version is listed first.

NOTE
The SEL-3560 uses the same firmware as the SEL-3555. The SEL-3560 will show the SEL-3555 FID string on the web interface. All revision notes which apply to the SEL-3555 firmware are also applicable to the SEL-3560.

The ACSELERATOR RTAC SEL-5033 Software version indicates the minimum version number that you must use for that firmware release. Newer versions of ACSELERATOR RTAC always work with older versions of firmware.

The following list indicates the tables where the RTAC and Axion firmware revision histories can be found:

- RTACs: *Table A.1*
- SEL-2245-2: *Table A.2*
- SEL-2245-22: *Table A.3*
- SEL-2245-221: *Table A.4*
- SEL-2245-3: *Table A.5*
- SEL-2245-4: *Table A.6*
- SEL-2245-411: *Table A.7*
- SEL-2245-42: *Table A.8*

Table A.1 RTAC Firmware Revision History

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R148-V3-Z000012-D20210713 SEL-3530-4-R148-V3-Z000012-D20210713 SEL-2241-R148-V3-Z000012-D20210713 SEL-3505-R148-V3-Z000012-D20210713 SEL-3505-3-R148-V3-Z000012-D20210713 SEL-3532-N-R148-V3-Z000012-D20210713 SEL-3555-R148-V3-Z000012-D20210713 SEL-3350-R148-V3-Z000012-D20210713 ACSELERATOR RTAC Software Version: 1.32.xxx Manual Date Code: 20210713	Includes all the functions of R148-V2 with the following additions:	
	Resolved an issue in Modbus Server where the server disconnects the TCP socket connection after 15 seconds of inactivity.	All
	Resolved an issue in project API and web server where project uploads are not accepted when the project name contains spaces.	All
	Resolved an issue in logic engine API where the response is incomplete when data in the response contain values of `NaN`.	All
	Resolved an incorrect example IP range in Network Audit configuration in the RTAC web interface.	All
	Resolved an issue in C37.118 server where excessive CPU resources are used when using transport mode UDP_S with multicast enabled.	All
	Resolved an issue introduced in R148-V0 where EIA-485/422 does not function when configured.	3350
	Resolved an issue introduced in R148-V0 where +5V port power on serial ports does not function when configured.	3350
Firmware Identification (FID) Numbers: SEL-3530-R148-V2-Z000015-D20210414 SEL-3530-4-R148-V2-Z000015-D20210414 SEL-2241-R148-V2-Z000015-D20210414 SEL-3505-R148-V2-Z000015-D20210414 SEL-3505-3-R148-V2-Z000015-D20210414 SEL-3532-N-R148-V2-Z000015-D20210414 SEL-3555-R148-V2-Z000015-D20210414 SEL-3350-R148-V2-Z000015-D20210414 ACSELERATOR RTAC Software Version: 1.32.xxx Manual Date Code: 20210414	Includes all the functions of R148-V1 with the following additions:	
	Resolved an issue introduced in R148-V0 where SEL Client will occasionally report 0 for analog values momentarily.	3530 3530-4 2241 3505 3505-3
	Resolved an issue introduced in R148-V0 where EtherCAT analog values will occasionally report 0 momentarily.	3530 3530-4 2241
	Resolved an issue where a connected IED page may not always load correctly when using Chromium-based browsers such as Microsoft Edge or Google Chrome.	All
	Resolved an issue where the SNMP Client/Manager incorrectly deasserts to Offline and increments the Message_Received_Count POU pins.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue where SEL Client will not collect events from an SEL-700G when using FTP.	All
	Resolved an issue where RADIUS will not authenticate when null characters are included in the authentication response from the server.	All
Firmware Identification (FID) Numbers: SEL-3530-R148-V1-Z000012-D20210320 SEL-3530-4-R148-V1-Z000012-D20210320 SEL-2241-R148-V1-Z000012-D20210320 SEL-3505-R148-V1-Z000012-D20210320 SEL-3505-3-R148-V1-Z000012-D20210320 SEL-3532-N-R148-V1-Z000012-D20210320 SEL-3555-R148-V1-Z000012-D20210320 SEL-3350-R148-V1-Z000012-D20210320 ACSELERATOR RTAC Software Version: 1.32.xxx Manual Date Code: 20210320	Includes all the functions of R148-V0 with the following additions:	
	Resolved an issue that may prevent users from logging in after restoring User Accounts settings during a project send while using an authenticated LDAP or Radius account.	All
	Resolved an issue introduced in R148-V0 where fileIO may not read events into the logic engine.	All
	Resolved an issue introduced in R148-V0 where a firmware downgrade to R142 or previous may not succeed.	3532 3555
	Resolved an issue introduced in R148-V0 where Ethernet communications may be unavailable after updating firmware to R148-V0.	3555
Firmware Identification (FID) Numbers: SEL-3530-R148-V0-Z001187-D20210208 SEL-3530-4-R148-V0-Z001187-D20210208 SEL-2241-R148-V0-Z001187-D20210208 SEL-3505-R148-V0-Z001187-D20210208 SEL-3505-3-R148-V0-Z001187-D20210208 SEL-3532-N-R148-V0-Z001187-D20210208 SEL-3555-R148-V0-Z001187-D20210208 SEL-3350-R148-V0-Z001187-D20210208 ACSELERATOR RTAC Software Version: 1.32.xxx Manual Date Code: 20210208	Added support for SEL-3350.	3350
	Added support for SNMP Agent/Server.	All
	Added support for RAID 1.	3555
	Added Configuration API to support monitoring for configuration changes using a configuration fingerprint.	All
	Increased support for DNP client/server count to 500.	3555
	Added support for sleep mode on local display.	3532 3555 3350
	Added support to name collected COMTRADE files according to IEEE COMNAME standard.	All
	Enhanced Audit Utilities to include vendor information for MAC addresses in reports.	3532 3555 3350
	Enhanced IEC 60870-103 client to support 16 elements for Measured Value II.	All
	Enhanced Audit Utilities to support returning all TCP and UDP port states instead of only open ports as a part of the audit report.	3532 3555 3350
	Enhanced Modbus Client to support COMTRADE and SOE collection from GE UR firmware versions 7.91 and later.	All
	Updated BIOS for 3555-2 to 2.3.49152.138.	3555
	Updated how SEL Client tracks unsolicited event count statistics.	All
	Updated Logic engine compiler.	All
	Modified COMTRADE folder naming generated from Axion modules to match COMTRADE event file names.	3530 3530-4 2241 3555 3350

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue where subject alternative names are not included in the certificate signing request.	All
	Resolved an issue introduced in R145-V0 where IEC 60870-103 may not send commands.	3530 3530-4 2241 3505 3505-3
	Addressed behavior due to changes in web browsers where users may remain logged in after closing a web browser session when the Web Session Timeout is set to 0.	All
	Resolved an issue where communication ports on the connected IED page may not appear correctly.	3555
	Resolved an issue where IRIG-B output on PCI cards may have incorrect time offset when Enable_IRIG_B_TIME_OUT_DST_OFFSET is set to FALSE.	3555
	Resolved an issue where the sel_file.soe_request_ascending function does not return SOE records if the ReturnAlarmSoeOnly input was set to TRUE.	All
	Increased COMTRADE collection storage count to 10,000.	3555
Firmware Identification (FID) Numbers: SEL-3530-R147-V2-Z000007-D20210318 SEL-3530-4-R147-V2-Z000007-D20210318 SEL-2241-R147-V2-Z000007-D20210318 SEL-3505-R147-V2-Z000007-D20210318 SEL-3505-3-R147-V2-Z000007-D20210318 SEL-3532-N-R147-V2-Z000007-D20210318 SEL-3555-R147-V2-Z000007-D20210318 ACSELERATOR RTAC Software Version: 1.31.xxx Manual Date Code: 20210320	Includes all the functions of R147-V1 with the following additions:	
	Updated BIOS for SEL-3555-2 to 2.3.49152.138.	3355
	Resolved an issue introduced in R145-V0 where IEC 60870-103 may not send commands.	3530 3530-4 2241 3505 3505-3
	Addressed behavior caused by changes in web browsers where users may remain logged in after closing a web browser session when the Web Session Timeout is set to 0.	All
	Enhanced Modbus Client to support COMTRADE and SOE collection from GE UR firmware versions 7.91 and later.	All
	Resolved an issue where the sel_file.soe_request_ascending function does not return SOE records if the ReturnAlarmSoeOnly input was set to TRUE.	All
	Resolved an issue that may prevent users from logging in after restoring User Accounts settings during a project send while using an authenticated LDAP or Radius account.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R147-V1-Z000001-D20201026 SEL-3530-4-R147-V1-Z000001-D20201026 SEL-2241-R147-V1-Z000001-D20201026 SEL-3505-R147-V1-Z000001-D20201026 SEL-3505-3-R147-V1-Z000001-D20201026 SEL-3532-N-R147-V1-Z000001-D20201026 SEL-3555-R147-V1-Z000001-D20201026 ACSELERATOR RTAC Software Version: 1.31.xxx Manual Date Code: 20201026	Includes all the functions of R147-V0 with the following additions:	
	Modified C37.118 server transport scheme UDP_S time to live count to 64.	All
	Resolved an issue in MIRRORED BITS RX where invalid validity may be incorrectly set when system time changes.	All
	Resolved an issue in Audit Utilities where some hosts may not be included in the network audit report.	3532 3555
	Resolved an issue in SEL Client introduced in R147-V0 where COMTRADE events are not successfully collected over serial connections.	All
	Resolved an issue that may cause an erroneous one-hour offset to be introduced into protocol time stamps in the hour leading up to and following the exit of DST.	All
	Modified Audit Utilities to support a configurable ARP request rate.	3532 3555
Firmware Identification (FID) Numbers: SEL-3530-R147-V0-Z000992-D20200820 SEL-3530-4-R147-V0-Z000992-D20200820 SEL-2241-R147-V0-Z000992-D20200820 SEL-3505-R147-V0-Z000992-D20200820 SEL-3505-3-R147-V0-Z000992-D20200820 SEL-3532-N-R147-V0-Z000992-D20200820 SEL-3555-R147-V0-Z000992-D20200820 ACSELERATOR RTAC Software Version: 1.31.xxx Manual Date Code: 20200820	Added support for creating custom URLs on the RTAC that will proxy web traffic to and from an origin web server.	All
	Enhanced Audit Utilities API to manage multiple audit reports.	3555 3532
	Added support to manage network audits via web interface.	3555 3532
	Enhanced Logic Engine API to support grouping of tags.	All
	Enhanced Logic Engine API to support streaming of tags and tag groups.	All
	Added additional support to the Project Web API for activating projects.	All
	Enhanced SEL client to support event collection via FTP.	All
	Enhanced SEL client to save Flex parse responses to a file.	All
	Added IP address of client machine to SOE when users log in, log out, or enter an incorrect password.	All
	Enhanced PRP to support multiple pairs of PRP interfaces.	3555
	Enhanced MMS server to support 100 unbuffered reports and 100 buffered reports.	All
	Added support for Bit mode in sysCom library for serial ports.	All
	Added support for Axion Wave Server.	3530 3530-4 2241 3555
	Updated HMI binz file to version 3.3.1.217.	3530 3530-4 2241 3532 3555
	Increased EtherCAT network bandwidth to support additional modules and tags.	3530 3530-4 2241 3555

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Enhanced EtherCAT network to support 32 frequency groups.	3530 3530-4 2241 3555
	Added support to EthernetIP for PRP.	All
	Added support to SEL client for SEL-400G.	All
	Resolved an issue where GOOSE may not receive messages on Ethernet Port 2. Only affects SEL-3555-2 and SEL-3560.	3555
	Removed session ID and username from web server GET requests.	All
	Web interface changed to prevent web server cross-domain client access.	All
	Resolved an issue in the DNP server where changing deadbands during runtime may cause the server to restart.	All
	Resolved an issue where incoming time stamps from protocols may be shifted by the configured daylight-saving time offset when the transition to and from daylight-saving time occurs. The time-stamp shift behavior lasts for the duration of the configured UTC_offset after the daylight-saving time transition.	All
	Updated IEEE C37.118 clients to use time quality from PMU if available.	All
	Added support to IEEE C37.118 server to choose polar or rectangular formatted phasors.	All
	Resolved an issue where EtherCAT discovery may not return all modules correctly if modules swapped positions after a successful initialization.	3530 3530-4 2241 3555
	Enhanced EtherCAT discovery to provide an accurate message when no modules are connected.	3530 3530-4 2241 3555
	Resolved an issue in SEL client where SEL-321-0 may not complete autoconfiguration.	All
Firmware Identification (FID) Numbers: SEL-3530-R146-V3-Z000006-D20210318 SEL-3530-4-R146-V3-Z000006-D20210318 SEL-2241-R146-V3-Z000006-D20210318 SEL-3505-R146-V3-Z000006-D20210318 SEL-3505-3-R146-V3-Z000006-D20210318 SEL-3532-N-R146-V3-Z000006-D20210318 SEL-3555-R146-V3-Z000006-D20210318 ACSELERATOR RTAC Software Version: 1.30.xxx Manual Date Code: 20210320	Includes all the functions of R146-V2 with the following additions:	
	Updated BIOS for SEL-3555-2 to 2.3.49152.138.	3555
	Resolved an issue introduced in R145-V0 where IEC 60870-103 may not send commands.	3530 3530-4 2241 3505 3505-3
	Addressed behavior caused by changes in web browsers where users may remain logged in after closing a web browser session when the Web Session Timeout is set to 0.	All
	Enhanced Modbus Client to support COMTRADE and SOE collection from GE UR firmware versions 7.91 and later.	All
	Modified C37.118 server transport scheme UDP_S time to live count to 64.	All
	Resolved an issue in MIRRORED BITS RX where invalid validity may be incorrectly set when system time changes.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue where GOOSE may not receive messages on Ethernet Port 2. Only affects SEL-3555-2 and SEL-3560.	3555
	Resolved an issue in the DNP server where changing deadbands during runtime may cause the server to restart.	All
	Removed session id and username from web server GET requests.	All
	Web interface changed to prevent web server cross-domain client access.	All
	Resolved an issue where the sel_file.soe_request_ascending function does not return SOE records if the ReturnAlarmSoeOnly input was set to TRUE.	All
	Resolved an issue that may prevent users from logging in after restoring User Accounts settings during a project send while using an authenticated LDAP or Radius account.	All
	Resolved an issue in Audit Utilities where some hosts may not be included in the network audit report.	3532 3555
Firmware Identification (FID) Numbers: SEL-3530-R146-V2-Z000016-D20200610 SEL-3530-4-R146-V2-Z000016-D20200610 SEL-2241-R146-V2-Z000016-D20200610 SEL-3505-R146-V2-Z000016-D20200610 SEL-3505-3-R146-V2-Z000016-D20200610 SEL-3532-N-R146-V2-Z000016-D20200610 SEL-3555-R146-V2-Z000016-D20200610 ACSELERATOR RTAC Software Version: 1.30.xxx Manual Date Code: 20200610	Includes all the functions of R146-V1 with the following additions:	
	Added a Generate Log button on the Diagnostics page of the RTAC webpage.	All
	Resolved an issue where DNP servers configured to use TLS may experience high CPU utilization after startup.	All
	Removed the option to log out of the desktop manager from the local display.	3532 3555
	Resolved an issue where settings configured in the HMI during run time were lost after a power cycle.	3530 3530-4 2241 3532 3555
Firmware Identification (FID) Numbers: SEL-3530-R146-V1-Z000009-D20200425 SEL-3530-4-R146-V1-Z000009-D20200425 SEL-2241-R146-V1-Z000009-D20200425 SEL-3505-R146-V1-Z000009-D20200425 SEL-3505-3-R146-V1-Z000009-D20200425 SEL-3532-N-R146-V1-Z000009-D20200425 SEL-3555-R146-V1-Z000009-D20200425 ACSELERATOR RTAC Software Version: 1.30.xxx Manual Date Code: 20200425	Includes all the functions of R146-V0 with the following additions:	
	Resolved an issue where a direct firmware update to R146-V0 from R131 or earlier may not succeed.	All
	Resolved an issue in SEL Client where an ASCII request may not access the correct permissions level before issuing the request.	All
	Resolved an issue where event logs may not successfully download when included in a project send.	All
	Resolved an issue where RemoteOutOfBandManagement field may not be returned in the ID API request.	3555
Firmware Identification (FID) Numbers: SEL-3530-R146-V0-Z000002-D20200224 SEL-3530-4-R146-V0-Z000002-D20200224 SEL-2241-R146-V0-Z000002-D20200224 SEL-3505-R146-V0-Z000002-D20200224 SEL-3505-3-R146-V0-Z000002-D20200224 SEL-3532-N-R146-V0-Z000002-D20200224 SEL-3555-R146-V0-Z000002-D20200224 ACSELERATOR RTAC Software Version: 1.30.xxx Manual Date Code: 20200224	Added support for EtherNet/IP.	All
	Added support for REST APIs.	All
	Added support for Network Auditing.	3555
	Added support for multiple pairs of bonded Ethernet interfaces.	All
	Enhanced C37.118 server to support serial communications.	All
	Added new user role permissions to manage sending passwords to SEL relays when connected through SEL server.	All
	Added Remote Out-Of-Band Management status to web interface.	3555

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue where users may not be able to log into the web interface if the web session time-out is set to 0.	All
	Resolved an issue where usernames that contained special characters may not be able to access some parts of the web interface.	All
	Resolved an issue in GOOSE Receive where mapping DPS tag types to INS tag types may cause a logic engine restart.	All
	Resolved an issue where live data coloring may not change for the STR data type.	All
	Resolved an issue where a Modbus client may not resume event collection after losing communications.	All
	Resolved an issue where Modbus servers with direct transparent connections may not accept connections.	All
	Resolved an issue where lower time sync priorities may not return system time to higher time priorities after a time change.	All
	Resolved an issue where partial hour UTC offset from an IRIG-B signal may not be processed correctly.	All
	Resolved an issue where a Modbus server may not transmit ASCII values correctly.	All
	Resolved an issue where an SEL client may log ASCII SER data into the RTAC SOE log more than once.	All
	Resolved an issue where EtherCat CTPT COMTRADE events may lose data while time is synchronizing to a new source.	3530 3530-4 2241 3555
	Resolved an issue where Comm Monitor may not capture all data when EtherCAT is enabled.	3530 3530-4 2241 3555
Firmware Identification (FID) Numbers: SEL-3530-R145-V3-Z000006-D20210318 SEL-3530-4-R145-V3-Z000006-D20210318 SEL-2241-R145-V3-Z000006-D20210318 SEL-3505-R145-V3-Z000006-D20210318 SEL-3505-3-R145-V3-Z000006-D20210318 SEL-3532-N-R145-V3-Z000006-D20210318 SEL-3555-R145-V3-Z000006-D20210318 ACSELERATOR RTAC Software Version: 1.29.xxx Manual Date Code: 20210320	Includes all the functions of R145-V2 with the following additions:	
	Updated BIOS for SEL-3555-2 to 2.3.49152.138.	3555
	Resolved an issue introduced in R145-V0 where IEC 60870-103 may not send commands.	3530 3530-4 2241 3532 3555
	Addressed behavior caused by changes in web browsers where users may remain logged in after closing a web browser session when the Web Session Timeout is set to 0.	All
	Enhanced the Modbus Client to support COMTRADE and SOE collection from GE UR firmware versions 7.91 and later.	All
	Modified the C37.118 server transport scheme UDP_S time to live count to 64.	All
	Resolved an issue in MIRRORED BITS RX where invalid validity may be incorrectly set when system time changes.	All
	Resolved an issue where GOOSE may not receive messages on Ethernet Port 2. Only affects SEL-3555-2 and SEL-3560.	3555

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue in the DNP server where changing deadbands during runtime may cause the server to restart.	All
	Removed session id and username from web server GET requests.	All
	Web interface changed to prevent web server cross-domain client access.	All
	Resolved an issue where the sel_file.soe_request_ascending function does not return SOE records if the ReturnAlarmSoeOnly input was set to TRUE.	All
	Resolved an issue that may prevent users from logging in after restoring User Accounts settings during a project send while using an authenticated LDAP or Radius account.	All
Firmware Identification (FID) Numbers: SEL-3530-R145-V2-Z000014-D20200610 SEL-3530-4-R145-V2-Z000014-D20200610 SEL-2241-R145-V2-Z000014-D20200610 SEL-3505-R145-V2-Z000014-D20200610 SEL-3505-3-R145-V2-Z000014-D20200610 SEL-3532-N-R145-V2-Z000014-D20200610 SEL-3555-R145-V2-Z000014-D20200610 ACSELERATOR RTAC Software Version: 1.29.xxx Manual Date Code: 20200610	Includes all the functions of R145-V1 with the following additions:	
	Resolved an issue where settings configured in the HMI during run time were lost after a power cycle.	3530 3530-4 2241 3532 3555
	Removed the option to log out of the desktop manager from the local display.	3532 3555
	Resolved an issue where event logs may not restore properly.	All
	Resolved an issue in GOOSE Receive where mapping DPS tag types to INS tag types may cause a logic engine restart.	All
	Added a Generate Log button on the Diagnostics page of the RTAC webpage.	All
	Resolved an issue in SEL Client where an ASCII request may not access the correct permissions level before issuing the request.	All
	Resolved an issue where event logs may not successfully download when included in a project send.	All
	Resolved an issue where some RTAC projects may request an encryption password when activated through the RTAC's web interface.	All
	Resolved an issue where DNP servers configured to use TLS may experience high CPU utilization after startup.	All
Firmware Identification (FID) Numbers: SEL-3530-R145-V1-Z000010-D20200122 SEL-3530-4-R145-V1-Z000010-D20200122 SEL-2241-R145-V1-Z000010-D20200122 SEL-3505-R145-V1-Z000010-D20200122 SEL-3505-3-R145-V1-Z000010-D20200122 SEL-3532-N-R145-V1-Z000010-D20200122 SEL-3555-R145-V1-Z000010-D20200122 ACSELERATOR RTAC Software Version: 1.29.xxx Manual Date Code: 20200122	Includes all the functions of R145-V0 with the following additions:	
	Resolved an issue where URL whitelisting may not allow access to a URL.	All
	Resolved an issue where DNP or IEC 60870-5-101/104 servers may report duplicate events after a power cycle of one unit when using SCADA Protocol Redundancy.	All
	Resolved an issue where Live Data may cause a logic engine restart.	All
	Resolved an issue in DNP Server introduced in R144-V0 in which enabling unsolicited messaging at startup may not work.	All
	Resolved an issue where SEL client may go offline while reading large COMTRADE events.	All
	Resolved an issue where the web interface may not be viewable with Microsoft Internet Explorer.	All
	Resolved an issue with Live Data where tags may not reappear on the web interface after EN is set to FALSE.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue where all user accounts may be lost during a firmware change.	All
	Resolved an issue where web interface logins may not succeed when five or more web sessions are active.	All
	Updated BIOS versions: ➤ SEL-3555: 1.9.49152.65 ➤ SEL-3555-2: 2.2.49152.121	3555
Firmware Identification (FID) Numbers: SEL-3530-R145-V0-Z000001-D20190830 SEL-3530-4-R145-V0-Z000001-D20190830 SEL-2241-R145-V0-Z000001-D20190830 SEL-3505-R145-V0-Z000001-D20190830 SEL-3505-3-R145-V0-Z000001-D20190830 SEL-3532-N-R145-V0-Z000001-D20190830 SEL-3555-R145-V0-Z000001-D20190830 ACSELERATOR RTAC Software Version: 1.29.xxx Manual Date Code: 20190830	Enhanced IEEE C37.118 client capabilities.	All
	Enhanced IEEE C37.118 Axion PMU capabilities.	All
	Enhanced IEEE C37.118 server capabilities.	All
	Added support for IEC 60870-5-103 client.	All
	Added support for IEC 60870-5-103 Dynamic Disturbance Recorder collection.	All
	Enhanced SEL server to allow reading files via ASCII.	All
	Enhanced SEL server to manage passwords for local accounts.	All
	Enhanced Modbus server to support multiple servers on the same serial port or IP port.	All
	Added Licensed Features page to RTAC web interface.	All
	Enhanced S/FTP client to support creation of folders.	All
	Enhanced MMS server to support the stSeld attribute for controls.	All
	Expanded CT ratio range for all ac inputs on Axion I/O modules.	3530 3530-4 2241 3555
	Resolved an issue in Live Data where string alarms may not clear.	All
	Resolved an issue on SEL-3555 where communications may not resume after an interruption on Serial Port 2.	3555
	Resolved an issue where ACSELERATOR RTAC may not successfully read projects.	5033
	Resolved an issue that may occur when comparing projects in ACSELERATOR RTAC.	5033
	Resolved an issue in ACSELERATOR RTAC where a project may not be removed from a folder in the tree view.	5033
	Resolved an issue in ACSELERATOR RTAC where a project may always detect settings have changed if IEC 61131 programs or data types are completely commented out.	5033
	Resolved an issue in ACSELERATOR RTAC where the tag processor may not generate time and quality code for the auto-fill setting correctly.	5033

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R144-V8-Z000031-D20200917 SEL-3530-4-R144-V8-Z000031-D20200917 SEL-2241-R144-V8-Z000031-D20200917 SEL-3505-R144-V8-Z000031-D20200917 SEL-3505-3-R144-V8-Z000031-D20200917 SEL-3532-N-R144-V8-Z000031-D20200917 SEL-3555-R144-V8-Z000031-D20200917 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20201026	Includes all the functions of R144-V7 with the following additions:	
	Modified C37.118 server transport scheme UDP_S time to live count to 64.	All
	Resolved an issue in MIRRORED BITS RX where invalid validity may be incorrectly set when system time changes.	All
	Removed session ID and username from web service GET requests.	All
	Web interface changed to prevent web server cross-domain client access.	All
	Resolved an issue in the DNP server where changing deadbands during runtime may cause the server to restart.	All
	Resolved an issue on SEL-3555-2 and SEL-3560 where GOOSE may not receive messages on Ethernet Port 2. Note that this does not affect the SEL-3555.	3555
	Resolved an issue in Live Data introduced in R144-V7 that may prevent tags from being displayed.	All
Firmware Identification (FID) Numbers: SEL-3530-R144-V7-Z000028-D20200610 SEL-3530-4-R144-V7-Z000028-D20200610 SEL-2241-R144-V7-Z000028-D20200610 SEL-3505-R144-V7-Z000028-D20200610 SEL-3505-3-R144-V7-Z000028-D20200610 SEL-3532-N-R144-V7-Z000028-D20200610 SEL-3555-R144-V7-Z000028-D20200610 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20200610	Includes all the functions of R144-V6 with the following additions:	
	Removed the option to log out of the desktop manager from the local display.	3555
	Resolved an issue where settings configured in the HMI during run time were lost after a power cycle.	3530 3530-4 2241 3532 3555
	Resolved an issue in GOOSE Receive where mapping DPS tag types to INS tag types may cause a logic engine restart.	All
	Added a Generate Log button on the Diagnostics page of the RTAC webpage.	All
	Resolved an issue where DNP servers configured to use TLS may experience high CPU utilization after startup.	All
Firmware Identification (FID) Numbers: SEL-3530-R144-V6-Z000022-D20200420 SEL-3530-4-R144-V6-Z000022-D20200420 SEL-2241-R144-V6-Z000022-D20200420 SEL-3505-R144-V6-Z000022-D20200420 SEL-3505-3-R144-V6-Z000022-D20200420 SEL-3532-N-R144-V6-Z000022-D20200420 SEL-3555-R144-V6-Z000022-D20200420 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20200425	Includes all the functions of R144-V5 with the following additions:	
	Resolved an issue in SEL Client where an ASCII request may not access the correct permissions level before issuing the request.	All
	Resolved an issue where some RTAC projects may request an encryption password when activated through the RTAC's web interface.	All
	Resolved an issue where web interface logins may not succeed when five or more web sessions are active.	All
	Resolved an issue where event logs may not successfully download when included in a project send.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R144-V5-Z000003-D20200122 SEL-3530-4-R144-V5-Z000003-D20200122 SEL-2241-R144-V5-Z000003-D20200122 SEL-3505-R144-V5-Z000003-D20200122 SEL-3505-3-R144-V5-Z000003-D20200122 SEL-3532-N-R144-V5-Z000003-D20200122 SEL-3555-R144-V5-Z000003-D20200122 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20200122	Includes all the functions of R144-V4 with the following additions:	
	Resolved an issue where URL whitelisting may not allow access to a URL.	All
	Resolved an issue where DNP or IEC 60870-5-101/104 servers may report duplicate events after a power cycle of one unit when using SCADA Protocol Redundancy.	All
	Updated BIOS versions: ➤ SEL-3555: 1.9.49152.65 ➤ SEL-3555-2: 2.2.49152.121	3555
Firmware Identification (FID) Numbers: SEL-3530-R144-V4-Z000017-D20191015 SEL-3530-4-R144-V4-Z000017-D20191015 SEL-2241-R144-V4-Z000017-D20191015 SEL-3505-R144-V4-Z000017-D20191015 SEL-3505-3-R144-V4-Z000017-D20191015 SEL-3532-N-R144-V4-Z000017-D20191015 SEL-3555-R144-V4-Z000017-D20191015 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20191015	Includes all the functions of R144-V3 with the following additions:	
	Resolved an issue where BIOS may not have updated after loading R144-V3.	3555
	Resolved an issue where CDC client may send more than one select during a control operation.	All
	Resolved an issue where SEL client may go offline while reading large COMTRADE events.	All
	Resolved an issue on SEL-3555 where communications may not resume after an interruption on Serial Port 2.	3555
	Resolved an issue in DNP Server introduced in R144-V0 in which enabling unsolicited messaging at startup may not work.	All
Firmware Identification (FID) Numbers: SEL-3530-R144-V3-Z000013-D20190508 SEL-3530-4-R144-V3-Z000013-D20190508 SEL-2241-R144-V3-Z000013-D20190508 SEL-3505-R144-V3-Z000013-D20190508 SEL-3505-3-R144-V3-Z000013-D20190508 SEL-3532-N-R144-V3-Z000013-D20190508 SEL-3555-R144-V3-Z000013-D20190508 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20190508	Includes all the functions of R144-V2 with the following additions:	
	Resolved an issue in IEC-60870-5-101 server where an incorrect response for link status with function code 9 may be returned.	All
	Resolved an issue where user-created Live Data filters did not persist through firmware upgrades.	All
	Resolved an issue with Live Data performance for configurations that have many tags and aliases.	All
	Resolved an issue with SEL client where relays that do not support Fast Meter may not complete autoconfiguration.	All
	Resolved an issue introduced in R144-V2 where SEL-3555 BIOS may not update.	3555
	Added BIOS version to web interface dashboard.	3555
	Resolved issue where not all GOOSE messages were processed on SEL-3390-E4 cards.	3555

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R144-V2-Z000010-D20190216 SEL-3530-4-R144-V2-Z000010-D20190216 SEL-2241-R144-V2-Z000010-D20190216 SEL-3505-R144-V2-Z000010-D20190216 SEL-3505-3-R144-V2-Z000010-D20190216 SEL-3532-N-R144-V2-Z000010-D20190216 SEL-3555-R144-V2-Z000010-D20190216 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20190216	Includes all the functions of R144-V1 with the following additions:	
	Updated BIOS versions: ➤ SEL-3555: 1.8.49152.54 ➤ SEL-3555-2: 2.1.49152.52	3555
	Enhanced Dynamic Disturbance Recorder extension for projects created in or converted to R144 and later.	5033
	Resolved an issue introduced in R144-V0 where changes to the system UTC offset may not be applied to output time until a power cycle occurs.	All
	Resolved an issue introduced in R144-V0 where DNP clients may restart when a DNP client receives a packet while enabling.	All
	Resolved an issue introduced in R143-V0 where DNP server may use excessive CPU resources when enabling.	All
Firmware Identification (FID) Numbers: SEL-3530-R144-V1-Z000002-D20190111 SEL-3530-4-R144-V1-Z000002-D20190111 SEL-2241-R144-V1-Z000002-D20190111 SEL-3505-R144-V1-Z000002-D20190111 SEL-3505-3-R144-V1-Z000002-D20190111 SEL-3532-N-R144-V1-Z000002-D20190111 SEL-3555-R144-V1-Z000002-D20190111 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20190111	Includes all the functions of R144-V0 with the following addition:	
	Resolved an issue introduced in R144-V0 in DNP server where the RTAC may become unresponsive when DNP server communications are disrupted and re-established.	All
Firmware Identification (FID) Numbers: SEL-3530-R144-V0-Z000376-D20181217 SEL-3530-4-R144-V0-Z000376-D20181217 SEL-2241-R144-V0-Z000376-D20181217 SEL-3505-R144-V0-Z000376-D20181217 SEL-3505-3-R144-V0-Z000376-D20181217 SEL-3532-N-R144-V0-Z000376-D20181217 SEL-3555-R144-V0-Z000376-D20181217 ACSELERATOR RTAC Software Version: 1.28.xxx Manual Date Code: 20181217	Added support for SEL-2245-221 LEA module.	3530 3530-4 2241 3555
	Added support for Radius.	All
	Added support for CDC Type 2 Client.	All
	Added support in ACSELERATOR RTAC to perform database maintenance for improved performance.	5033
	Added support for PTP Power Profile.	All
	Enhanced MMS client to retain entryID values for buffered reports through power cycles.	3530 3530-4 2241 3532 3555
	Improved IEC 61850 configuration import time.	5033
	Added ability to acknowledge all SOE records from logic engine.	All
	Added support for non-English characters in Tag Processor.	5033
	Added support for new Modbus Server Function Codes.	All
	Increased the rate at which system time will synchronize with incoming IRIG-B time.	All
	Added support for vector_t in Live Data.	All
	Added support for processing PTP packets with VLAN tags.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Added support for the Automation Thread to log data to the RTAC's SOE.	All
	Enhanced server IP port range on Modbus Client to 1–65535.	All
	Downloaded CEV events are no longer combined into a single CEV file.	All
	Resolved an issue in bonding failover where failover did not occur faster than one second.	All
	Resolved an issue introduced in R139 in which DNP client may not resume communications after a loss of communications.	All
	Resolved an issue in DNP server that may stop communications.	All
	Resolved an issue where MMS server may stop communications.	All
	Resolved an issue where PTP may not work on SEL-3555 Ethernet expansion cards.	3555
	Resolved an issue where GOOSE messages may not work on SEL-3555 Ethernet expansion cards.	3555
	Resolved an issue in the SEL client that may prevent collection of COMTRADE events from an SEL-735.	All
	Resolved an issue in the SEL client that may prevent collection of COMTRADE events from an SEL-751.	All
	Resolved an issue that prevented gateway communications when PRP is enabled.	All
	Resolved an issue in Tag Processor where Live Data did not compile correctly when referencing variables declared in IEC 61131 programs and global variable lists.	5033
	Resolved an issue in SEL client where Compressed ASCII message failures were incorrectly incremented after closing a transparent connection.	All
	Resolved an issue where receive MIRRORED BITS variable names could not be renamed.	All
	Resolved an issue in CDC Type 2 server where controls may not be processed.	All
	Resolved an issue in DNP server where static and event data may not report in the correct order.	All
Firmware Identification (FID) Numbers: SEL-3530-R143-V1-Z000002-D20200117 SEL-3530-4-R143-V1-Z000002-D20200117 SEL-2241-R143-V1-Z000002-D20200117 SEL-3505-R143-V1-Z000002-D20200117 SEL-3505-3-R143-V1-Z000002-D20200117 SEL-3532-N-R143-V1-Z000002-D20200117 SEL-3555-R143-V1-Z000002-D20200117 ACSELERATOR RTAC Software Version: 1.26.xxx Manual Date Code: 20200122	Includes all the functions of R143-V0 with the following additions:	
	Resolved an issue where a watchdog time-out may occur after a network loop occurs with many ARP messages.	3530 3530-4 2241 3505 3505-3
	Resolved an issue where an Ethernet interface may become unresponsive after a network loop occurs with many ARP messages.	3530 3530-4 2241 3505 3505-3
	Includes all the functions of R143-V0 with no additions.	3532 3555

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Number: SEL-3555-R143-V0-Z000210-D20180710 ACSELERATOR RTAC Software Version: 1.26.xxx Manual Date Code: 20180928	Added support for Developer Mode.	5033
	Enhanced version resolution for IEC 61131 libraries.	5033
	Added support for refactoring tag names.	5033
	Added support for logging tag type, DPS, in the tag processor.	5033
	Resolved an issue introduced in RTAC software 1.26.143.15566 where projects with CP2179 clients did not compile.	5033
	DNP Client setting, server IP port, no longer auto-increments when adding multiple DNP clients in the RTAC software.	5033
Firmware Identification (FID) Numbers: SEL-3530-R143-V0-Z000210-D20180710 SEL-3530-4-R143-V0-Z000210-D20180710 SEL-2241-R143-V0-Z000210-D20180710 SEL-3505-R143-V0-Z000210-D20180710 SEL-3505-3-R143-V0-Z000210-D20180710 SEL-3532-N-R143-V0-Z000210-D20180710 SEL-3555-R143-V0-Z000210-D20180710 **Note**: The SEL-3555 is not available for order after July 1st, 2019. The SEL-3555-2 is the replacement for the SEL-3555. The SEL-3555-2 does not support firmware versions prior to R143-V0. ACSELERATOR RTAC Software Version: 1.26.xxx Manual Date Code: 20180710	Added support for SEL-2245-411 4 CT/4 LEA Monitoring Module.	3530 3530-4 2241 3555
	Added support for EtherCAT discovery of connected Axion modules.	3530 3530-4 2241 3555
	Added support for single-mode fiber on the SEL-2243 Power Coupler.	3530 3530-4 2241 3555
	Added support for DNP secure authentication on DNP server.	All
	Added support for PTP on bonded and PRP Ethernet interfaces.	All
	Added URL whitelist to RTAC web interface.	All
	Added support for the RTAC software command line interface to send projects that are password protected.	5033
	Added support for 115200 baud rate for millisecond MIRRORED BITS.	All
	Added support for Kiosk mode via the local display.	3532 3555
	Added support for virtual keyboard on the local display.	3532 3555
	Enhanced **<Ctrl+X>** functionality in the Tag Processor of the RTAC software.	5033
	Added support for SEL-3555-2.	3555
	Removed support for TLS 1.0.	All
	Resolved an issue introduced in RTAC software 1.25.xxx.xxxx in which R124 projects may not download.	5033
	Resolved an issue in which DNP multi-drop servers may stop responding if configured in failover mode.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R142-V1-Z001001-D20200124 SEL-3530-4-R142-V1-Z001001-D20200124 SEL-2241-R142-V1-Z001001-D20200124 SEL-3505-R142-V1-Z001001-D20200124 SEL-3505-3-R142-V1-Z001001-D20200124 SEL-3532-N-R142-V1-Z001001-D20200124 SEL-3555-R142-V1-Z001001-D20200124 ACSELERATOR RTAC Software Version: 1.25.xxx Manual Date Code: 20200122	Includes all the functions of R142-V0 with the following additions:	
	Resolved an issue where a watchdog time-out may occur after a network loop occurs with many ARP messages.	3530 3530-4 2241 3505 3505-3
	Resolved an issue where an Ethernet interface may become unresponsive after a network loop occurs with many ARP messages.	3530 3530-4 2241 3505 3505-3
	Includes all the functions of R142-V0 with no additions.	3532 3555
Firmware Identification (FID) Numbers: SEL-3530-R142-V0-Z001001-D20180330 SEL-3530-4-R142-V0-Z001001-D20180330 SEL-2241-R142-V0-Z001001-D20180330 SEL-3505-R142-V0-Z001001-D20180330 SEL-3505-3-R142-V0-Z001001-D20180330 SEL-3532-N-R142-V0-Z001001-D20180330 SEL-3555-R142-V0-Z001001-D20180330 ACSELERATOR RTAC Software Version: 1.25.xxx Manual Date Code: 20180330	Added support for SES-92 Client.	All
	Added support to compare two RTAC configurations in ACSELERATOR RTAC.	5033
	Enhanced IEC 60870-5-101/104 server to order interrogation response by IOA addresses.	All
	Added the ability to modify DNP addresses in DNP servers via the logic engine.	All
	Addressed an issue where IEC 60870-5-101/104 or DNP servers may not send events from one server when SCADA Protocol Redundancy is enabled and both servers are actively communicating with clients.	All
	Modified MMS client so that it does not change configured tags timestamps when validity transitions from Good to Invalid.	3530 3530-4 2241 3532 3555
	Addressed an issue where projects may not be activated through the ACSELERATOR RTAC web interface.	All
Firmware Identification (FID) Numbers: SEL-3530-R141-V0-Z001001-D20171107 SEL-3530-4-R141-V0-Z001001-D20171107 SEL-2241-R141-V0-Z001001-D20171107 SEL-3505-R141-V0-Z001001-D20171107 SEL-3505-3-R141-V0-Z001001-D20171107 SEL-3532-N-R141-V0-Z001001-D20171107 SEL-3555-R141-V0-Z001001-D20171107 ACSELERATOR RTAC Software Version: 1.24.xxx Manual Date Code: 20171201	Added the ability to create custom user roles based upon selected access permissions.	All
	Added the password report to the RTAC web interface.	All
	Added support for SEL-2245-4 AC Metering Module for enabling/disabling module tags individually to conserve EtherCAT bandwidth. **Note**: R141 requires upgrading the module firmware to be compatible.	3530 3530-4 2241 3555
	Added support for SEL-2245-22 AI-ER module to run in AC voltage mode and increased oscillography to 24 kHz. **Note**: R141 requires upgrading the module firmware to be compatible.	3530 3530-4 2241 3555
	Added UDP-only transportation scheme to IEEE C37.118 Server protocol.	All
	Enhanced the web session time-out to include an infinite timeout option.	All
	Resolved an issue that could prevent auto-configuration from completing with some SEL-200 series relays.	All
	Resolved an issue introduced in R139-V0 in which the state number in the GOOSE protocol would not increment on the first data change.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R140-V0-Z001001-D20170728 SEL-3530-4-R140-V0-Z001001-D20170728 SEL-2241-R140-V0-Z001001-D20170728 SEL-3505-R140-V0-Z001001-D20170728 SEL-3505-3-R140-V0-Z001001-D20170728 SEL-3532-N-R140-V0-Z001001-D20170728 SEL-3555-R140-V0-Z001001-D20170728 ACSELERATOR RTAC Software Version: 1.23.xxx Manual Date Code: 20170728	Added support for the SEL-2245-42 AC Protection Module for enabling/disabling module tags individually to conserve EtherCAT bandwidth.	3530 3530-4 2241 3555
	Added support for IEEE C37.118 Server for the extended configuration message (CFG3).	All
	Added support for recording groups for recording RTAC logic engine analog quantities.	3530 3530-4 2241 3555
	Added a setting to recording groups to allow time aligning the trigger line to the RTAC logic assertion that triggered the recording group.	3530 3530-4 2241 3555
Firmware Identification (FID) Numbers: SEL-3530-R139-V2-Z001001-D20180202 SEL-3530-4-R139-V2-Z001001-D20180202 SEL-2241-R139-V2-Z001001-D20180202 SEL-3505-R139-V2-Z001001-D20180202 SEL-3505-3-R139-V2-Z001001-D20180202 SEL-3532-N-R139-V2-Z001001-D20180202 SEL-3555-R139-V2-Z001001-D20180202 ACSELERATOR RTAC Software Version: 1.22.xxx Manual Date Code: 20180202	Includes all the functions of R139-V1 with the following additions:	
	Addressed an issue where IEC 60870-5-101/104 or DNP servers may not send events out one server when SCADA Protocol Redundancy is enabled and both servers are actively communicating with clients.	All
	Resolved an issue introduced in R139-V0 in which the state number in the GOOSE protocol would not increment on the first data change.	All
Firmware Identification (FID) Numbers: SEL-3530-R139-V1-Z001001-D20170728 SEL-3530-4-R139-V1-Z001001-D20170728 SEL-2241-R139-V1-Z001001-D20170728 SEL-3505-R139-V1-Z001001-D20170728 SEL-3505-3-R139-V1-Z001001-D20170728 SEL-3532-N-R139-V1-Z001001-D20170728 SEL-3555-R139-V1-Z001001-D20170728 ACSELERATOR RTAC Software Version: 1.22.xxx Manual Date Code: 20170728	Includes all the functions of R139-V0 with the following additions:	
	Enhanced web server to modify DNP server addresses.	All
	Enhanced MMS client to support 129-character rptID for report control blocks.	All
	Resolved an issue introduced in R139 in which the last general interrogation report was not saved in the BRCB buffer.	All
Firmware Identification (FID) Numbers: SEL-3530-R139-V0-Z001001-D20170505 SEL-3530-4-R139-V0-Z001001-D20170505 SEL-2241-R139-V0-Z001001-D20170505 SEL-3505-R139-V0-Z001001-D20170505 SEL-3505-3-R139-V0-Z001001-D20170505 SEL-3532-N-R139-V0-Z001001-D20170505 SEL-3555-R139-V0-Z001001-D20170505 ACSELERATOR RTAC Software Version: 1.22.xxx Manual Date Code: 20170505	Added FTP and SFTP server.	All
	Added support for the SEL-2245-42 AC Protection Module.	3530 3530-4 2241 3555
	Added support for configuring recording groups to combine events from multiple SEL-2245-42 AC Protection modules, SEL-2244 Digital I/O modules, or digital IEC 61131 tags.	3530 3530-4 2241 3555
	Added support for Dynamic Disturbance Recording (DDR) Extension.	3530 3530-4 2241 3555
	Enhanced the RTAC webpage to support firmware updates.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Added the ability to support IP aliasing on Ethernet interfaces from the logic engine.	All
	Enhanced X.509 certificates to have more options for the valid time period.	All
	Enhanced SEL Server to support direct transparent connections.	All
	Enhanced SEL Server to notify ACSELERATOR TEAM of uncollected SOE events.	All
	Enhanced transparent connections with SEL Client to preserve quality of meter tags.	All
	Enhanced Live Data to support filtering.	All
	Enhanced Live Data to have descriptive labels to filter on.	All
	Enhanced Live Data to support the enable/disable of forcing Live Data per point.	All
	Enhanced DNP server to allow individual DNP points to be excluded from a DNP server class 0 response.	All
	Enhanced IEC 60870-5-101/104 Client and Server to allow per-point control selection of Direct/SBO type.	All
	Enhanced image quality on the local display for certain resolutions.	3532 3555
	Updated IEC 61850 MMS Client/Server and GOOSE for Edition 2 Certification.	3530 3530-4 2241 3555
	Resolved an issue introduced in R137 which would temporarily show SOE time stamps in UTC rather than local time when a new UTC offset was applied.	All
	Resolved an issue in SEL Client where an incorrect offset may be applied to SOE time stamps collected from ASCII SER reports.	All
	Resolved an issue introduced in R137 where attempting to go online with a project could incorrectly detect a settings change.	All
Firmware Identification (FID) Numbers: SEL-3530-R138-V0-Z001001-D20170220 SEL-3530-4-R138-V0-Z001001-D20170220 SEL-2241-R138-V0-Z001001-D20170220 SEL-3505-R138-V0-Z001001-D20170220 SEL-3505-3-R138-V0-Z001001-D20170220 SEL-3532-N-R138-V0-Z001001-D20170220 SEL-3555-R138-V0-Z001001-D20170220 ACSELERATOR RTAC Software Version: 1.21.xxx Manual Date Code: 20170220	Enhanced Live Data to support forcing from the web interface.	All
	Added Comm Monitor to Connected IED page on web interface.	All
	Added Ping to Connected IED page on web interface.	All
	Added Alstom Event Collection.	All
	Resolved an issue in LG 8979 Server where Analog Groups might not report assigned points correctly.	All
	Resolved an issue in IEC 61850 MMS Client that could prevent communications from starting.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R137-V0-Z001001-D20170109 SEL-3530-4-R137-V0-Z001001-D20170109 SEL-2241-R137-V0-Z001001-D20170109 SEL-3505-R137-V0-Z001001-D20170109 SEL-3505-3-R137-V0-Z001001-D20170109 SEL-3532-N-R137-V0-Z001001-D20170109 SEL-3555-R137-V0-Z001001-D20170109 ACSELERATOR RTAC Software Version: 1.20.xxx Manual Date Code: 20170109	Added Simple Network Manager Protocol (SNMP) Manager/Client.	All
	Added Parallel Redundancy Protocol (PRP).	All
	Added support for CDC Type II Server Protocol.	3532 3555
	Enhanced the SEL Client to support COMTRADE event collection.	All
	Enhanced COMTRADE Collection Count from 256 to 1024.	All
	Enhanced LG 8979 Client to support configurable timer count per control point.	All
	Resolved an issue with eDNA client where time stamps were not updated to the current time when sending data sync messages.	3532 3555
	Resolved an issue where time stamps on SOE webpage may differ by one second relative to SOE time stamps shown on the HMI SOE viewer.	All
	Resolved an issue introduced in R136 where the SEL-3555 may not synchronize the year correctly when connected to IRIG-B000.	3555
	Resolved an issue with SCADA Protocol Redundancy where duplicate events may be reported during a power cycle condition.	All
	Resolved an issue introduced in R136 where COMTRADE events may not be collected via Modbus Client.	All
	Resolved an issue in LG 8979 Server where Analog Groups may not report assigned points correctly.	All
	Resolved an issue which could prevent a project read process from completing if the SOE log was included.	All
Firmware Identification (FID) Numbers: SEL-3530-R136-V3-Z001001-D20180131 SEL-3530-4-R136-V3-Z001001-D20180131 SEL-2241-R136-V3-Z001001-D20180131 SEL-3505-R136-V3-Z001001-D20180131 SEL-3505-3-R136-V3-Z001001-D20180131 SEL-3532-N-R136-V3-Z001001-D20180131 SEL-3555-R136-V3-Z001001-D20180131 ACSELERATOR RTAC Software Version: 1.19.xxx Manual Date Code: 20180202	Includes all the functions of R136-V2 with the following addition:	
	Addressed an issue where projects may not be activated through the RTAC web interface.	All
Firmware Identification (FID) Numbers: SEL-3530-R136-V2-Z001001-D20170109 SEL-3530-4-R136-V2-Z001001-D20170109 SEL-2241-R136-V2-Z001001-D20170109 SEL-3505-R136-V2-Z001001-D20170109 SEL-3505-3-R136-V2-Z001001-D20170109 SEL-3532-N-R136-V2-Z001001-D20170109 SEL-3555-R136-V2-Z001001-D20170109 ACSELERATOR RTAC Software Version: 1.19.xxx Manual Date Code: 20170108	Includes all the functions of R136-V1 with the following additions:	
	Resolved an issue in IEC 61850 MMS Server where time stamps may be reported incorrectly when status value changes occur less than the configured bufTime apart.	All
	Resolved an issue introduced in R134 that potentially allows unauthorized access via the web interface.	All
	Resolved a potential issue introduced in R134 that could cause memory issues if user logic creates or modifies many files.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R136-V1-Z001001-D20161026 SEL-3530-4-R136-V1-Z001001-D20161026 SEL-2241-R136-V1-Z001001-D20161026 SEL-3505-R136-V1-Z001001-D20161026 SEL-3505-3-R136-V1-Z001001-D20161026 SEL-3532-N-R136-V1-Z001001-D20161026 SEL-3555-R136-V1-Z001001-D20161026 ACSELERATOR RTAC Software Version: 1.19.xxx Manual Date Code: 20161026	Includes all the functions of R136-V0 with the following addition:	
	Resolved an issue introduced in R136-V0 in the IEC 60870-5-104 server that could prevent data transfer after a TCP keep-alive time-out condition.	All
Firmware Identification (FID) Numbers: SEL-3530-R136-V0-Z001001-D20160624 SEL-3530-4-R136-V0-Z001001-D20160624 SEL-2241-R136-V0-Z001001-D20160624 SEL-3505-R136-V0-Z001001-D20160624 SEL-3505-3-R136-V0-Z001001-D20160624 SEL-3532-N-R136-V0-Z001001-D20160624 SEL-3555-R136-V0-Z001001-D20160624 ACSELERATOR RTAC Software Version: 1.19.xxx Manual Date Code: 20160624	Added IEC 60870-5-101/104 Client Protocol.	All
	Enhanced the IEC 61850 MMS client to collect COMTRADE files.	All
	Enhanced the IEC 61850 MMS server to serve files through MMS file transfer.	All
	Added support to synchronize time from Precise Time Protocol (PTP).	All
	Added support for SCADA Protocol Redundancy for DNP server and IEC 60870-5-101/104 server to support redundant RTAC systems.	All
	Added the ability to upload RTAC project files through the web interface.	All
	Added the ability to configure network interfaces through IEC 61131.	All
	Enhanced SEL transparent communications to support YMODEM file transfer.	All
	Enhanced the SEL server to accept multiple simultaneous connections for engineering access.	All
	Added support for Axion I/O modules on SEL-3555.	3555
	Enhanced SEL-3555 to support a combination of 256 DNP clients and servers.	3555
	Enhanced SEL-3555 to support a combination of 256 Modbus clients and servers.	3555
	Enhanced SEL-3555 to support a combination of 254 SEL clients and servers.	3555
	Enhanced SEL-3555 projects to support 100,000 tags.	3555
	Added eDNA client on SEL-3555 and SEL-3532.	3532 3555
	Enhanced ACSELERATOR RTAC to add password protection to projects.	5033
	Enhanced ACSELERATOR RTAC to encrypt and obfuscate passwords used for IED communications.	5033
	Enhanced ACSELERATOR RTAC to import XML files into existing project configurations.	5033
	Enhanced the IEC 60870-5-101 server to support failover.	All
	Resolved an issue in the LG 8979 client where the oscillatory bit was not updated correctly when the MCD bit was set in a change report.	All
	Resolved an issue that could prevent SEL protocol communications with SEL-700GT relays.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue introduced in R134 that could prevent retained variables from persisting through a power cycle.	All
	Updated SSH client and server to address CVE-2013-4421 and CVE-2013-4434.	All
Firmware Identification (FID) Numbers: SEL-3530-R135-V3-Z001001-D20170109 SEL-3530-4-R135-V3-Z001001-D20170109 SEL-2241-R135-V3-Z001001-D20170109 SEL-3505-R135-V3-Z001001-D20170109 SEL-3505-3-R135-V3-Z001001-D20170109 SEL-3532-N-R135-V3-Z001001-D20170109 SEL-3555-R135-V3-Z001001-D20170109 ACSELERATOR RTAC Software Version: 1.18.xxx Manual Date Code: 20170108	Includes all the functions of R135-V2 with the following additions:	
	Resolved an issue in IEC 61850 MMS Server where time stamps may be reported incorrectly when status value changes occur less than the configured bufTime apart.	All
	Resolved an issue introduced in R134 that potentially allows unauthorized access via the web interface.	All
	Resolved a potential issue introduced in R134 that could cause memory issues if user logic creates or modifies many files.	All
Firmware Identification (FID) Numbers: SEL-3530-R135-V2-Z001001-D20160212 SEL-3530-4-R135-V2-Z001001-D20160212 SEL-2241-R135-V2-Z001001-D20160212 SEL-3505-R135-V2-Z001001-D20160212 SEL-3505-3-R135-V2-Z001001-D20160212 SEL-3532-N-R135-V2-Z001001-D20160212 SEL-3555-R135-V2-Z001001-D20160212 ACSELERATOR RTAC Software Version: 1.18.xxx Manual Date Code: 20160212	Includes all the functions of R135-V1 with the following additions:	
	Removed TLS cipher support for RC4.	All
	Upgraded default X.509 certificate digital hashing algorithm from SHA1 to SHA2.	All
	Upgraded default X.509 certificate RSA public key from 1024 to 2048 bits.	All
	Resolved an issue introduced in R134-V1 where communications to a DNP server could be disrupted after receiving an invalid DNP request.	All
	Resolved potential issue in MMS Client which could prevent dataset requests from being issued.	3530 3530-4 2241 3532 3555
Firmware Identification (FID) Numbers: SEL-3530-R135-V1-Z001001-D20160115 SEL-3530-4-R135-V1-Z001001-D20160115 SEL-2241-R135-V1-Z001001-D20160115 SEL-3505-R135-V1-Z001001-D20160115 SEL-3505-3-R135-V1-Z001001-D20160115 SEL-3532-N-R135-V1-Z001001-D20160115 SEL-3555-R135-V1-Z001001-D20160115 ACSELERATOR RTAC Software Version: 1.18.xxx Manual Date Code: 20160201	Resolved issue in ACSELERATOR RTAC version 1.18.7222.1758 where IEC 60870 server sector settings were not retained.	5033
Firmware Identification (FID) Numbers: SEL-3530-R135-V1-Z001001-D20160115 SEL-3530-4-R135-V1-Z001001-D20160115 SEL-2241-R135-V1-Z001001-D20160115 SEL-3505-R135-V1-Z001001-D20160115 SEL-3505-3-R135-V1-Z001001-D20160115 SEL-3532-N-R135-V1-Z001001-D20160115 SEL-3555-R135-V1-Z001001-D20160115 ACSELERATOR RTAC Software Version: 1.18.xxx Manual Date Code: 20160115	Includes all the functions of R135-V0 with the following additions:	
	Resolved password monitoring issue in SEL client with password change detection and storage.	All
	Resolved issue that could cause incorrect IRIG time distribution when multiple serial cards are in use.	3555

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R135-V0-Z001001-D20150904 SEL-3530-4-R135-V0-Z001001-D20150904 SEL-2241-R135-V0-Z001001-D20150904 SEL-3505-R135-V0-Z001001-D20150904 SEL-3505-3-R135-V0-Z001001-D20150904 SEL-3532-N-R135-V0-Z001001-D20150904 SEL-3555-R135-V0-Z001001-D20150904 ACSELERATOR RTAC Software Version: 1.18.xxx Manual Date Code: 20150904	Added IEC 61850 MMS Server.	3530 3530-4 2241 3505 3505-3 3532
	Added docking and undocking of editor panels in ACSELERATOR RTAC.	5033
	Enhanced ACSELERATOR RTAC start menu to display RTAC type and enable right-click convert and rename of RTAC projects.	5033
	Added a setting in SEL server to disable passthrough connections.	3530 3530-4 2241 3505 3505-3 3532
	Added support for accessing SOE and event files in the logic engine by using the FileIO library.	3530 3530-4 2241 3505 3505-3 3532
	Added a setting to LG 8979 to skip startup initialization of the protocol communications.	3530 3530-4 2241 3505 3505-3 3532
	Modified LG 8979 Server to assert the MCD bit when oscillatory is asserted.	3530 3530-4 2241 3505 3505-3 3532
	Enhanced Alarm Summary with automatic updates and alarm color changes.	3530 3530-4 2241 3505 3505-3 3532
	Modified Live Data on the Web interface to use view filters, display Asserted/Deasserted instead of True/False for the status value, and change entry color when point is in alarm.	3530 3530-4 2241 3505 3505-3 3532

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Modified IED Summary Report on the RTAC web interface to show bytes sent and received on each channel.	3530 3530-4 2241 3505 3505-3 3532
	Resolved potential vulnerability in firmware versions prior to R135 in which an affected DNP3 UDP or C37.118 UDP communications channel could be disrupted or terminated.	3530 3530-4 2241 3505 3505-3 3532
Firmware Identification (FID) Numbers: SEL-3530-R134-V4-Z001001-D20170109 SEL-3530-4-R134-V4-Z001001-D20170109 SEL-2241-R134-V4-Z001001-D20170109 SEL-3505-R134-V4-Z001001-D20170109 SEL-3505-3-R134-V4-Z001001-D20170109 SEL-3532-N-R134-V4-Z001001-D20170109 SEL-3555-R134-V4-Z001001-D20170109 ACSELERATOR RTAC Software Version: 1.17.xxx Manual Date Code: 20170108	Includes all the functions of R134-V3 with the following additions:	
	Resolved an issue introduced in R134 that potentially allows unauthorized access via the web interface.	All
	Resolved a potential issue introduced in R134 that could cause memory issues if user logic creates or modifies many files.	All
Firmware Identification (FID) Numbers: SEL-3530-R134-V3-Z001001-D20160212 SEL-3530-4-R134-V3-Z001001-D20160212 SEL-2241-R134-V3-Z001001-D20160212 SEL-3505-R134-V3-Z001001-D20160212 SEL-3505-3-R134-V3-Z001001-D20160212 SEL-3532-N-R134-V3-Z001001-D20160212 SEL-3555-R134-V3-Z001001-D20160212 ACSELERATOR RTAC Software Version: 1.17.xxx Manual Date Code: 20160212	Includes all the functions of R134-V1 with the following additions:	
	Removed TLS cipher support for RC4.	All
	Upgraded default X.509 certificate digital hashing algorithm from SHA1 to SHA2.	All
	Upgraded default X.509 certificate RSA public key from 1024 to 2048 bits.	All
	Resolved potential vulnerability in firmware versions prior to R134-V2 in which an affected DNP3 UDP or C37.118 UDP communications channel could be disrupted or terminated.	All
	Resolved potential issue in MMS Client which could prevent dataset requests from being issued.	All
	Resolved an issue introduced in R134-V1 where communications to a DNP server could be disrupted after receiving an invalid DNP request.	All
	Resolved password monitoring issue in SEL client with password change detection and storage.	All
	Resolved issue that could cause incorrect IRIG time distribution when multiple serial cards are in use.	3555
Firmware Identification (FID) Number: SEL-3532-N-R134-V2-Z001001-D20150504 ACSELERATOR RTAC Software Version: 1.17.xxx Manual Date Code: 20150504	Includes all the functions of R134-V1 with the following addition:	
	Optimized HMI processing to improve performance.	3532

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Number: SEL-3555-R134-V1-Z001001-D20150417 ACSELERATOR RTAC Software Version: 1.17.xxx Manual Date Code: 20150504	Includes all the functions of R134-V0 with the following additions:	
	Added support for ACSELERATOR TEAM software to acknowledge SOE and collected events through a direct database connection.	3555
	Resolved potential issue with Modbus client communications if the remote server is repeatedly disconnected.	3555
	Resolved an issue with web Live Data updates.	3555
	Removed support for SSL 3.0.	3555
Firmware Identification (FID) Numbers: SEL-3530-R134-V1-Z001001-D20150417 SEL-3530-4-R134-V1-Z001001-D20150417 SEL-2241-R134-V1-Z001001-D20150417 SEL-3505-R134-V1-Z001001-D20150417 SEL-3505-3-R134-V1-Z001001-D20150417 SEL-3532-N-R134-V1-Z001001-D20150417 **Note**: To address component obsolescence, Flash memory components were replaced on the SEL-3530 RTAC units manufactured after December 1, 2015. Because of compatibility issues, R134 and older firmware will only function on SEL-3530 RTACs manufactured prior to December 1, 2015. ACSELERATOR RTAC Software Version: 1.17.xxx Manual Date Code: 20150504	Added exe-GUARD anti-malware and SEL Secure Linux.	All
	Added support for HTML5 in the optional HMI.	3530 3530-4 2241 3532
	Added password monitoring to SEL server.	All
	Added IED configuration change monitoring through SEL protocol.	All
	Added Flex Parse protocol.	All
	Added transparent connections and **WHO** commands through SEL server.	All
	Removed support for SSL 3.0 to address the POODLE exploit.	All
	Added the ability to collect and integrate ASCII SER data from SEL Client devices.	All
	Added Firmware Checksum and Project ID tags for web server and logic engine monitoring/reporting.	All
	Added Double Transmission of Events, Pulse On/Off commands, and UDP communication to IEC 60870 server.	All
	Added Double Point Binary Inputs and DataSets to DNP3 Client/Server and Object 50 Variation 4 to the DNP3 Server for AN-2013 Advanced Photovoltaic Profile.	All
	Added the ability to collect COMTRADE and SER events from GE relays with Modbus.	All
	Added Apply_System_Time_UTC_Offset_ Globally setting to allow all protocols to observe system time settings.	All
	Added SEL Fast Message read support for the SEL-487V.	All
	Added support to the SEL Client to communicate with SEL-267D Relays.	All
	Added the name of the current project to the Power_Up_Description tag.	All
	Added support for ACSELERATOR TEAM software to acknowledge SOE and collected events through a direct database connection.	All
	Enhanced DNP3 Client to improve control operations with devices that have intermittent communication.	All
	Resolved an issue with web Live Data updates.	All
	Resolved an issue in CP2179 Client that could cause the POU to go offline after an SBO operation.	All
	Resolved an issue that prevented the USB_B_Link diagnostic from functioning.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved an issue introduced in R132 that could cause transparent Ethernet tunneled connections to SEL-400 Series Relays to become unresponsive.	All
	Resolved an issue introduced in R133 that caused the quality and status bits to indicate incorrectly when EtherCAT modules were disconnected and Disable_On_Network_Error was set to FALSE.	3530 3530-4 2241
	Resolved a potential issue introduced in R126 that could cause memory issues if there are more than 150 SEL IED events on the RTAC.	All
	Resolved an issue in EtherCAT time stamps where UTC offset was adjusted by seconds rather than by minutes.	3530 3530-4 2241
Firmware Identification (FID) Number: SEL-3555-R134-V0-Z001001-D20150311 ACSELERATOR RTAC Software Version: 1.17.xxx Manual Date Code: 20150311	Initial version.	3555
Firmware Identification (FID) Numbers: SEL-3530-R133-V1-Z001001-D20160212 SEL-3530-4-R133-V1-Z001001-D20160212 SEL-2241-R133-V1-Z001001-D20160212 SEL-3505-R133-V1-Z001001-D20160212 SEL-3505-3-R133-V1-Z001001-D20160212 SEL-3532-N-R133-V1-Z001001-D20160212 ACSELERATOR RTAC Software Version: 1.16.xxx Manual Date Code: 20160212	Includes all the functions of R133-V0 with the following additions:	
	Removed TLS cipher support for RC4.	All
	Upgraded default X.509 certificate digital hashing algorithm from SHA1 to SHA2.	All
	Upgraded default X.509 certificate RSA public key from 1024 to 2048 bits.	All
	Resolved potential vulnerability in firmware versions prior to R133-V1 in which an affected DNP3 UDP or C37.118 UDP communications channel could be disrupted or terminated.	All
	Resolved potential issue in MMS Client which could prevent dataset requests from being issued.	3530 3530-4 2241 3532
	Resolved password monitoring issue in SEL client with password change detection and storage.	All
Firmware Identification (FID) Numbers: SEL-3530-R133-V0-Z001001-D20141103 SEL-3530-4-R133-V0-Z001001-D20141103 SEL-2241-R133-V0-Z001001-D20141103 SEL-3505-R133-V0-Z001001-D20141103 SEL-3505-3-R133-V0-Z001001-D20141103 SEL-3532-N-R133-V0-Z001001-D20141103 ACSELERATOR RTAC Software Version: 1.16.xxx Manual Date Code: 20141103	Added SEL-2245-22 dc analog input extended range module.	3530 3530-4 2241
	Updated C37.118 server implementation for IEEE C37.118.1a-2014 compliance with the SEL-2245-4 CT/PT module.	3530 3530-4 2241
	Enhanced C37.118 server when using SEL-2245-4 CT/PT modules to support as many as 64 phasor quantities at 60 messages per second.	3530 3530-4 2241
	Allowed tunneled DNP server connections to use primary and failover channels with matching TCP port numbers.	All
	Added additional error messages in EtherCAT to display for invalid module settings or firmware.	3530 3530-4 2241
	Added UDP as a connection option for Access Points.	All
	Added option for 200 and 600 data rates to IEC 60870-5-101 protocol.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Added an option to respond with a dump response to an exception request in SES-92 server.	All
	Resolved an issue introduced in R125 that could cause the IRIG out signal to differ from the RTAC system time by one second for a short period.	All
	Resolved a vulnerability in OpenSSL that could allow unauthorized access to the web interface.	All
Firmware Identification (FID) Numbers: SEL-3530-R132-V1-Z001001-D20160212 SEL-3530-4-R132-V1-Z001001-D20160212 SEL-2241-R132-V1-Z001001-D20160212 SEL-3505-R132-V1-Z001001-D20160212 SEL-3505-3-R132-V1-Z001001-D20160212 SEL-3532-N-R132-V1-Z001001-D20160212 ACSELERATOR RTAC Software Version: 1.15.xxx Manual Date Code: 20160212	Includes all the functions of R132-V0 with the following additions:	
	Removed TLS cipher support for RC4.	All
	Upgraded default X.509 certificate digital hashing algorithm from SHA1 to SHA2.	All
	Upgraded default X.509 certificate RSA public key from 1024 to 2048 bits.	All
	Resolved potential vulnerability in firmware versions prior to R132-V1 in which an affected DNP3 UDP or C37.118 UDP communications channel could be disrupted or terminated.	All
	Resolved potential issue in MMS Client which could prevent dataset requests from being issued.	3530 3530-4 2241 3532
	Resolved password monitoring issue in SEL client with password change detection and storage.	All
Firmware Identification (FID) Numbers: SEL-3530-R132-V0-Z001001-D20140616 SEL-3530-4-R132-V0-Z001001-D20140616 SEL-2241-R132-V0-Z001001-D20140616 SEL-3505-R132-V0-Z001001-D20140616 SEL-3505-3-R132-V0-Z001001-D20140616 SEL-3532-N-R132-V0-Z001001-D20140616 ACSELERATOR RTAC Software Version: 1.15.xxx Manual Date Code: 20140714	Added Syslog protocol.	3530 3530-4 2241 3505
	Added CP2179 protocol.	3530 3530-4 2241 3505
	Added support for SEL-2245-2 DC AI Waveform Recording and Filter C. SEL-2245-2 module firmware must be upgraded to R102 or higher to use these features.	3530 3530-4 2241
	Added support in DNP3 to allow control actions with non-standard optypes.	3530 3530-4 2241 3505
	Added support in DNP3 to allow serial tunneled connections for primary and secondary failover ports.	3530 3530-4 2241 3505
	Added configurable TCP keep-alive message rates.	3530 3530-4 2241 3505
	Added SSL/TLS or SSH tunneling for Ethernet protocols.	3530 3530-4 2241 3505
	Added support for converting a project for another RTAC type.	5033

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Modified ACSELERATOR RTAC installation process to further protect database integrity during upgrade process.	5033
	Added explanation in ACSELERATOR RTAC to indicate projects saved in R132 software version cannot be opened in earlier software versions. Changes also require user to resend entire project if the pre-R132 project is opened in R132 ACSELERATOR RTAC.	5033
	Enhanced software debugging utilities and library support.	5033
	Enhanced ladder logic implementation to include rung comments.	5033
	Resolved an issue in LG 8979 that could cause the server to stop responding after receiving certain malformed packets.	3530 3530-4 2241 3505
	Revised C37.118 client to as many as 32 inputs at 1 message per second.	3530 3530-4 2241 3505
	Revised SOE time-stamping to update the time stamp when the quality changes to invalid.	3530 3530-4 2241 3505
	Revised alarm acknowledge so that only unacknowledged alarms can be acknowledged.	3530 3530-4 2241 3505
	Resolved an issue with the SEL Client where the New Filtered Event Fault Location could reset when more than two events occur during the lockout period.	3530 3530-4 2241 3505
	Initial version.	3532
	Initial version.	3505-3
Firmware Identification (FID) Numbers: SEL-3530-R131-V0-Z001001-D20140428 SEL-3530-4-R131-V0-Z001001-D20140428 SEL-2241-R131-V0-Z001001-D20140428 SEL-3505-R131-V0-Z001001-D20140428 ACSELERATOR RTAC Software Version: 1.14.xxx Manual Date Code: 20140428	Added SEL-2245-3 dc analog output module.	3530 3530-4 2241
Firmware Identification (FID) Numbers: SEL-3530-R130-V0-Z001001-D20140225 SEL-3530-4-R130-V0-Z001001-D20140225 SEL-2241-R130-V0-Z001001-D20140225 SEL-3505-R130-V0-Z001001-D20140225 ACSELERATOR RTAC Software Version: 1.14.xxx Manual Date Code: 20140225	Resolved an issue introduced in R126 where the SEL Client protocol could limit execution of multiple contemporaneous controls to one per processing cycle.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R129-V0-Z001001-D20140210 SEL-3530-4-R129-V0-Z001001-D20140210 SEL-2241-R129-V0-Z001001-D20140210 SEL-3505-R129-V0-Z001001-D20140210 ACSELERATOR RTAC Software Version: 1.14.xxx Manual Date Code: 20140210	Resolved an issue introduced in R126 that could prevent Axion modules from enabling after a power cycle.	3530 3530-4 2241
	Resolved an issue that could prevent a 61850 MMS device from reconnecting to the RTAC MMS Client following a cable disconnect/ reconnect sequence when operating with a large number of IEDs.	3530 3530-4 2241
	Resolved an issue that could cause the date/time on an SEL-2241 RTAC to reset to midnight of 1/1/2000 if the SEL-2241 is rebooted on a Sunday and is not connected to IRIG. SEL-3530, SEL-3530-4, and SEL-3505 RTACs are not affected.	2241
Firmware Identification (FID) Numbers: SEL-3530-R128-V0-Z001001-D20140113 SEL-3530-4-R128-V0-Z001001-D20140113 SEL-2241-R128-V0-Z001001-D20140113 SEL-3505-R128-V0-Z001001-D20140113 ACSELERATOR RTAC Software Version: 1.14.xxx Manual Date Code: 20140113	Resolved an issue introduced in R126 that could cause Axion digital I/O tags to be incorrectly mapped if the tags are enabled or disabled from the default configuration.	3530 3530-4 2241
Firmware Identification (FID) Numbers: SEL-3530-R127-V0-Z001001-D20131216 SEL-3530-4-R127-V0-Z001001-D20131216 SEL-2241-R127-V0-Z001001-D20131216 SEL-3505-R127-V0-Z001001-D20131216 ACSELERATOR RTAC Software Version: 1.14.xxx Manual Date Code: 20131216	Resolved potential issue in which asserted contact inputs may not be correctly reported directly following settings download. Inputs are reported correctly following power cycle of the RTAC. This issue impacts only onboard contact inputs; Axion I/O modules are unaffected.	All
	Resolved an issue where the MMS Client could toggle offline or have a memory leak with a large number of IEDs.	3530 3530-4 2241
Firmware Identification (FID) Numbers: SEL-3530-R126-V0-Z001001-D20131206 SEL-3530-4-R126-V0-Z001001-D20131206 SEL-2241-R126-V0-Z001001-D20131206 SEL-3505-R126-V0-Z001001-D20131206 ACSELERATOR RTAC Software Version: 1.14.xxx Manual Date Code: 20131206	Modified download and restart method to greatly accelerate settings change time.	All
	Added online edit capability for IEC 61131 programming.	All
	Added pulse output support for contact output control points.	All
	Extended range of supported characters for complex passwords.	All
	Increased precision on contact input time stamps, especially when using ac input.	All
	Added SSH mode for SEL server access.	All
	Added ability to select default web or HMI homepage on web interface.	All
	Modified DNP server to generate a freeze event if the freeze time is changed.	All
	Added synchronization of RTE to top of second if EtherCAT is connected.	3530 3530-4 2241
	Added DNP support for G20V1, immediate freeze.	All
	Resolved potential EIA-232 control line issue with DNP Server if channel failover feature is used.	All
	Resolved potential issue with LDAP server failover.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved potential issue when using static routes on a bridged interface.	All
	Resolved an issue where the offset for IRIG input was not being applied.	All
Firmware Identification (FID) Numbers: SEL-3530-R125-V0-Z001001-D20130827 SEL-3530-4-R125-V0-Z001001-D20130827 SEL-2241-R125-V0-Z001001-D20130827 SEL-3505-R125-V0-Z001001-D20130827 ACSELERATOR RTAC Software Version: 1.13.xxx Manual Date Code: 20130827	Added SEL-2245-4 4CT/4PT ac metering module.	3530 3530-4 2241
	Added IEEE C37.118 server protocol.	All
	Corrected an issue where DNP would continue switching RTS on the primary interface and fail to switch on the secondary interface.	All
	Enhanced EtherCAT connection editor to allow inserting 4-slot chassis with associated module rules.	3530 3530-4 2241
	Enhanced ACSELERATOR RTAC to allow collapsing sections of settings and include a check box for Advanced Settings.	5033
	Enhanced project export features to allow exporting and importing with XML files.	5033
	Improved onboard hardware inputs time-stamp accuracy to 2 ms.	All
	Increased accuracy of Task Usage statistics displayed in the web server.	All
Firmware Identification (FID) Numbers: SEL-3530-R124-V0-Z001001-D20130628 SEL-3530-4-R124-V0-Z001001-D20130628 SEL-2241-R124-V0-Z001001-D20130628 SEL-3505-R124-V0-Z001001-D20130628 ACSELERATOR RTAC Software Version: 1.12.xxx Manual Date Code: 20130628	Added IEC 60870-5-101 and IEC 60870-5-104 server protocols.	All
	Added SES-92 server protocol.	All
	Added advanced-development startup switch to decrease download times for sending test projects.	All
	Added setting to DNP client to disable startup messages.	All
	Added interframe delay setting for DNP server.	All
	Added ability to select 65534 as a DNP address.	All
	Improved HMI communications to reduce CPU burden and improve HMI load time.	3530 3530-4 2241
	Enhanced passwords to allow embedded spaces.	All
	Enhanced ACSELERATOR RTAC to allow 20000 characters in a project description.	5033
	Enhanced web server performance by instructing PC web browser to cache pages.	All
	Modified web interface to show percent of real-time usage instead of percent of CPU usage.	All
	Removed divide by zero error with 64-bit number math.	All
	Resolved issue in DNP server where an Object 12, Var 3 with many points could cause a buffer overflow.	All
	Resolved issue where Modbus strings are not encoded properly.	All
	Resolved issue that could cause MIRRORED BITS communications to not show invalid state even when the cable is disconnected.	All
	Resolved issue in DNP server where quality change did not generate events.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved time-synch issue in DNP where first time-synch message is not accurate.	All
	Resolved issue to allow IEC 61850 GOOSE to work more reliably over bridged interfaces.	All
Firmware Identification (FID) Numbers: SEL-3530-R123-V0-Z002001-D20130117 SEL-3530-4-R123-V0-Z002001-D20130117 SEL-2241-R123-V0-Z002001-D20130117 SEL-3505-R123-V0-Z002001-D20130117 ACSELERATOR RTAC Software Version: 1.11.41xx Manual Date Code: 20130531	Resolved issue with communications counters that could cause the unit to reset.	5033
Firmware Identification (FID) Numbers: SEL-3530-R123-V0-Z001001-D20130117 SEL-3530-4-R123-V0-Z001001-D20130117 SEL-2241-R123-V0-Z001001-D20130117 SEL-3505-R123-V0-Z001001-D20130117 ACSELERATOR RTAC Software Version: 1.11.41xx Manual Date Code: 20130117	Resolved issue with upgrading ACSELERATOR RTAC from pre-R122 versions to R123.	5033
	Resolved issue in which AI dead bands in DNP were reset to default values if a pre-R122 project is converted to R123.	5033
Firmware Identification (FID) Numbers: SEL-3530-R123-V0-Z001001-D20130117 SEL-3530-4-R123-V0-Z001001-D20130117 SEL-2241-R123-V0-Z001001-D20130117 SEL-3505-R123-V0-Z001001-D20130117 ACSELERATOR RTAC Software Version: 1.11.41xx Manual Date Code: 20130117	Increased DNP/IP UDP listener ports from 32 to 128 layers.	All
	Added support for IEC 61850 GOOSE on bridged Ethernet interfaces.	All
	Modified SEL client event collection to accept relay events with nonstandard checksums.	All
	Resolved issue that causes SEL server to become nonresponsive if unsolicited event reporting is enabled but **Allow Anonymous** is set to TRUE.	All
	Resolved issue in which primary gateway was not configured if one of the interfaces were bridged or bonded to another interface.	All
	Resolved issue in DNP client that may cause out-of-memory condition.	All
	Resolved issue in DNP Server that may cause failure to start under certain setting conditions.	All
Firmware Identification (FID) Numbers: SEL-3530-R122-V0-Z001001-D20121121 SEL-3530-4-R122-V0-Z001001-D20121121 SEL-2241-R122-V0-Z001001-D20121121 SEL-3505-R122-V0-Z001001-D20121121 ACSELERATOR RTAC Software Version: 1.11.xxx Manual Date Code: 20121121	Added LG 8979 Client protocol.	All
	Added LG 8979 Server protocol.	All
	Increased valid DNP server point numbers to span 0–65535.	All
	Added INS as an analog data type in DNP client and DNP server.	All
	Added support for control point tagging in the RTAC HMI.	3530 3530-4 2241
	Added support for IEC 61850 indexed buffered reports.	3530 3530-4 2241
	Added POU pins to IEC 61850 to indicate control states.	3530 3530-4 2241

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Enhanced DNP server to report only enabled tags in a sparse map.	All
	Resolved issue with C37.118 protocol on the SEL-700G Generator Protection Relay.	All
	Enhanced error checking in IEC 61850 for invalid CID files and configuration mismatch.	3530 3530-4 2241
	Updated IEC 61850 MMS Client and GOOSE for KEMA certification.	3530 3530-4 2241
	Resolved software issue that sometimes resulted in the error message, "One or more library references failed to load."	5033
	Resolved issue that sometimes caused incorrect logging of watchdog timeout message.	All
	Resolved issue resulting in incorrect saving of Ethernet settings on the web interface if the page is closed prematurely.	All
	Resolved issue with matching IEC 61850 report names if some are only one character long.	3530 3530-4 2241
Firmware Identification (FID) Numbers: SEL-3530-R121-V0-Z001001-D20121031 SEL-3530-4-R121-V0-Z001001-D20121031 SEL-2241-R121-V0-Z001001-D20121031 SEL-3505-R121-V0-Z001001-D20121031 ACSELERATOR RTAC Software Version: 1.10.xxx Manual Date Code: 20121031	Resolved issue that may cause intermittent connection failures with ACSELERATOR RTAC.	All
Firmware Identification (FID) Numbers: SEL-3530-R120-V0-Z001001-D20120924 SEL-3530-4-R120-V0-Z001001-D20120924 SEL-2241-R120-V0-Z001001-D20120924 SEL-3505-R120-V0-Z001001-D20120924 ACSELERATOR RTAC Software Version: 1.10.xxx Manual Date Code: 20121010	Resolved issue with ACSELERATOR RTAC (v1.10.3427.1314) that caused update issues with INC and APC data types.	5033
	Updated web server to address potential denial of service (DOS) attacks.	All
Firmware Identification (FID) Numbers: SEL-3530-R120-V0-Z001001-D20120924 SEL-3530-4-R120-V0-Z001001-D20120924 SEL-2241-R120-V0-Z001001-D20120924 SEL-3505-R120-V0-Z001001-D20120924 ACSELERATOR RTAC Software Version: 1.10.xxx Manual Date Code: 20120924	Resolved issue that may cause IRIG output to drift as much as one second.	All
	Resolved issues with IEC 61850 MMS client that may cause polling and report collection to become disabled.	3530 3530-4 2241

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R119-V0-Z001001-D20120720 SEL-3530-4-R119-V0-Z001001-D20120720 SEL-2241-R119-V0-Z001001-D20120720 SEL-3505-R119-V0-Z001001-D20120720 ACSELERATOR RTAC Software Version: 1.10.xxx Manual Date Code: 20120720	Added support for Live Data visualization on web interface.	3530 3530-4 2241
	Added advanced filtering feature for SOE log.	3530 3530-4 2241
	Added support for as many as 100 DNP servers and DNP shared maps.	3530 3530-4 2241
	Added authentication string to IEC 61850 client.	3530 3530-4 2241
	Increased max SEL flex parse messages from 30 to 50.	3530 3530-4 2241
	Modified DNP/IP Server to reply to the most recently connected polling client.	3530 3530-4 2241
	Modified SYS_TIME() function so you can use it in the automation and main tasks.	3530 3530-4 2241
	Added DCD option to DNP modem settings to allow DCD for carrier detect instead of CTS (applicable to COM_01 on RTACs or COM_05 on SEL-3505).	3530 3530-4 2241
	Modified DNP Client to interrupt an outstanding poll to issue a control operation.	3530 3530-4 2241
	Resolved potential issue in SEL Client that may trigger an out of memory condition if Flex Parse messages are configured and a port is constantly offline.	3530 3530-4 2241
	Upgraded operating system real-time kernel.	3530 3530-4 2241
	Modified SEL Client Ethernet communications to prevent repeated auto-configuration when Fast SER messaging is enabled in settings but unsupported by the IED.	3530 3530-4 2241
	Resolved potential issue with dropping MMS buffered reports in SEL-351 communications.	3530 3530-4 2241
	Modified Modbus/TCP to purge any pending controls when the channel transitions to an on-line state from an off-line state.	3530 3530-4 2241
	Initial version.	3505

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R118-V0-Z001001-D20120427 SEL-3530-4-R118-V0-Z001001-D20120427 SEL-2241-R118-V0-Z001001-D20120427 ACSELERATOR RTAC Software Version: 1.9.xxx Manual Date Code: 20120427	Revised for addition of 16 analog input and 10 FHC control output EtherCAT modules.	All
Firmware Identification (FID) Numbers: SEL-3530-R117-V0-Z001001-D20120316 SEL-3530-4-R117-V0-Z001001-D20120316 SEL-2241-R117-V0-Z001001-D20120316 ACSELERATOR RTAC Software Version: 1.8.xxx Manual Date Code: 20120423	Enhanced ACSELERATOR RTAC to include the following:	5033
	Improved retaining of alias tags and POU pin settings after resetting IEC 61850 configuration.	5033
	Improved consistency of renaming and adding folders actions.	5033
	Modified project description editing to retain formatting.	5033
	Improved error checking to flag cross-assignment errors between automation and main task.	5033
	Modified to decrease the frequency of auto-save/compile operations on an open project.	5033
	Updated several device definition files.	5033
	Added the FIND feature in Tags grid views.	5033
Firmware Identification (FID) Numbers: SEL-3530-R117-V0-Z001001-D20120316 SEL-3530-4-R117-V0-Z001001-D20120316 SEL-2241-R117-V0-Z001001-D20120316 ACSELERATOR RTAC Software Version: 1.8.xxx Manual Date Code: 20120316	Resolved an issue introduced in R113 where digital input tags may be erroneously re-assigned when a user changes any SEL-2242-2 digital input setting.	All
Firmware Identification (FID) Numbers: SEL-3530-R116-V0-Z001001-D20120301 SEL-3530-4-R116-V0-Z001001-D20120301 SEL-2241-R116-V0-Z001001-D20120301 ACSELERATOR RTAC Software Version: 1.8.xxx Manual Date Code: 20120301	Resolved issue that may prevent Ethernet communications if the last octet of the subnet mask is a non-zero value.	All
	Modified the firmware upgrade to select the **Primary Gateway** check box if a default gateway was previously assigned.	All
Firmware Identification (FID) Numbers: SEL-3530-R115-V0-Z001001-D20120208 SEL-3530-4-R115-V0-Z001001-D20120208 SEL-2241-R115-V0-Z001001-D20120208 ACSELERATOR RTAC Software Version: 1.8.xxx Manual Date Code: 20120208	Added IEC 61850 MMS Client support.	All
	Added Network Global Variables (NGVLs) protocol.	All
	Provided ability to place GOOSE in automation task.	All
	Added ability to capture strings from access point router source receive messages.	All
	Added remediation for denial of service (DOS) attacks.	All
	Added default gateway for each Ethernet interface.	All
	Resolved issue where function block names did not show up in input assistant.	All
	Added ability to resolve DNP controls regardless of Tag Processor location in task list.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved issue to allow download of all 30,000 SOE logs via CSV.	All
	Added installation integrity check to assist resolving ACSELERATOR RTAC installation issues.	5033
	Corrected out of memory conditions that may occur with excessive database access.	All
	Resolved issues caused by special characters in usernames.	All
	Resolved issue with restoring >500 event collection logs when sending a project with the Event Logs advanced option selected. The project would have to have been previously read with the Event Logs advanced read option selected from an RTAC that had event collection enabled and over 500 event logs stored.	All
	Provided ability to configure IN101 for 12 volt input.	All
	Resolved issue where DNP or Modbus may not reply to polls if they have no tags defined.	All
	Added min time for GOOSE retransmission.	All
	Resolved issue where Modbus server may not reconnect after settings change.	All
Firmware Identification (FID) Numbers: SEL-3530-R114-V0-Z001001-D20111010 SEL-3530-4-R114-V0-Z001001-D20111010 SEL-2241-R114-V0-Z001001-D20111010 ACSELERATOR RTAC Software Version: 1.7.xxx Manual Date Code: 20111010	Added ability to assign multiple DNP servers to the same serial port.	All
	Enhanced web interface to speed up webpage loading.	All
	Added HMI Operator user role.	All
	Enhanced web interface to address loading, communication, and stale session issues.	All
	Resolved issue with DNP shared map name changing during a save operation.	All
	Resolved issue with GOOSE MAC addressing to allow broader range of addresses.	All
	Resolved issue in DNP server failover with tunneled serial connections.	All
	Resolved issue with importing R108 project settings.	5033
	Resolved autoconfiguration issue with the SEL-587.	All
Firmware Identification (FID) Numbers: SEL-3530-R113-V0-Z001001-D20110721 SEL-3530-4-R113-V0-Z001001-D20110721 SEL-2241-R113-V0-Z001001-D20110721 ACSELERATOR RTAC Software Version: 1.6.xxx Manual Date Code: 20110721	Added EtherCAT client protocol.	3530 3530-4
	Added failover interface on DNP server.	3530 3530-4
	Added shared DNP maps.	3530 3530-4
	Added project templates to ACSELERATOR RTAC.	5033
	Enhanced DNP event buffer settings so that the buffer size can be defined based on the buffer type.	3530 3530-4
	Added a communications offline timer for DNP server.	3530 3530-4
	Enhanced DNP server frozen counters to include a separate event class.	3530 3530-4
	Added DDF for SEL-735.	5033

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Resolved issue with new event tags resetting unexpectedly.	3530 3530-4
	Resolved issue with AC input detection on input status points.	3530 3530-4
	Initial version.	2241
Firmware Identification (FID) Numbers: SEL-3530-R112-V0-Z001001-D20110603 SEL-3530-4-R112-V0-Z001001-D20110603 **Note**: This version of firmware is only compatible with versions of Diagram Builder greater than 1.1.0.1960. ACSELERATOR RTAC Software Version: 1.5.xxx Manual Date Code: 20110603	Updated ACSELERATOR RTAC to support Windows 7 64-bit operating systems.	5033
	Resolved issue where flex parse messages do not recognize level 2 passwords.	All
	Added ability to support bursts of 5,000 events in logging function.	All
	Enhanced to support much faster project downloads.	All
	Resolved issue where TXMB.status is not always updated correctly.	All
	Resolved issue with detecting device reset jumper position.	All
	Added SEL-751 support.	All
	Added bridging support on Ethernet ports.	All
	Added SEL client password pins to provide run-time password modifications.	All
	Resolved issue with occasional event collection delays.	All
	Increased DNP and Modbus clients to 100 instances.	All
	Added automation task cycle time for high-speed applications.	All
	Added Level_1 and Level_2 password POU pins for SEL client.	All
	Increase maximum log entries to 30,000.	All
Firmware Identification (FID) Numbers: SEL-3530-R111-V0-Z001001-D20110406 SEL-3530-4-R111-V0-Z001001-D20110406 ACSELERATOR RTAC Software Version: 1.4.xxx Manual Date Code: 20110406	Resolved communications issues with some versions of SEL-501 IEDs.	All
	Resolved update issues with the status of transmit SCmirroredbits.	All
	Improved initialization procedure of multi-drop DNP clients when devices are not connected during startup.	All
	Resolved possible issue with loss of SEL Protocol SER data when SER data are generated at a very fast rate.	All
	Resolved potential issues with SEL client if autoconfiguration is triggered while Flex Parse messaging is in process.	All
Firmware Identification (FID) Numbers: SEL-3530-R110-V0-Z001001-D20110201 SEL-3530-4-R110-V0-Z001001-D20110201 ACSELERATOR RTAC Software Version: 1.4.xxx Manual Date Code: 20110201	Resolved issues with read/import/export of tags and Tag Processor statements introduced in R109.	5033
	Resolved issue with import/export of new event and event collection POU pin settings.	5033
	Resolved issues manifested when using partial download option on projects that were converted from previous releases to R109 release.	All

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
Firmware Identification (FID) Numbers: SEL-3530-R109-V0-Z001001-D20101230 SEL-3530-4-R109-V0-Z001001-D20101230 ACSELERATOR RTAC Software Version: 1.4.xxx Manual Date Code: 20101230	Added IEC 61850 GOOSE messaging protocol.	All
	Added byte and register swapping for Modbus client and server.	All
	Resolved issue with autoconfig on SEL-587 for PEAK and DEMAND METER values.	All
	Modified software to allow contact outputs to function without any contact inputs defined.	All
	Resolved issue with UTC offset on incoming IRIG signal.	All
	Resolved issue in ACSELERATOR RTAC where on/off LED colors were reversed in virtual tags.	All
	Added support for partial settings download.	All
	Added support for as many as 32 DNP servers.	All
	Increased DNP server capacity to 32 simultaneous connections.	All
	Added local event collection, event archive, automatic event notification and transport to ACSELERATOR TEAM software.	All
	Added fault location/event processing to filter reclose sequences and provide fault location reset intervals.	All
	Support both odd and even parity with IRIG-B signals.	All
Firmware Identification (FID) Numbers: SEL-3530-R108-V0-Z001001-D20101001 SEL-3530-4-R108-V0-Z001001-D20101001 ACSELERATOR RTAC Software Version: 1.3.xxx Manual Date Code: 20101001	Increased analog, counter, and binary event buffer size to 10000 entries.	All
	Added SER label recognition functionality.	All
	Added support for MOT upgrades.	All
	Added support for web-based HMI.	All
	Resolved issue with detecting password defeat jumper position.	All
	Added support for virtual tags.	All
Firmware Identification (FID) Numbers: SEL-3530-R107-V0-Z001001-D20100805 SEL-3530-4-R107-V0-Z001001-D20100805 ACSELERATOR RTAC Software Version: 1.2.xxx Manual Date Code: 20100805	Changed DNP/IP clients to always use unique channels.	3530
	Extended database keep-alive timer to compensate for low-bandwidth connections.	3530
	Forced echo on in transparent connections only when legacy or authentication modes are TRUE and destination is disconnected.	3530
	Changed autoconfigure to support SEL-251 Relay.	3530
	Changed quality attributes of control points (operSPC, operINC, operAPC) to initialize as "good" on startup.	3530
	Initial version.	3530-4
Firmware Identification (FID) Number: SEL-3530-R106-V0-Z001001-D20100604 ACSELERATOR RTAC Software Version: 1.2.xxx Manual Date Code: 20100604	Added flexible parsing for SEL protocol.	3530
	Added support for central authentication using LDAP.	3530
	Add support for as many as 10 DNP/IP clients for one DNP server.	3530
	Added support for Ladder Logic Diagram (LD).	3530
	Added support for SEL protocol over IP.	3530
	Increased number of DNP block writes to 200 AO and 157 BO.	3530

Firmware, Software, and Manual Versions	Summary of Revisions	Affected Models
	Increased upper limit on SEL client poll period setting.	3530
	Added DDFs for SEL relays that do not support binary messaging.	5033
Firmware Identification (FID) Number: SEL-3530-R105-V0-Z001001-D20100422 ACSELERATOR RTAC Software Version: 1.1.xxx Manual Date Code: 20100422	Enhanced performance of project download and restart.	3530
	Added full support for 17-port serial expansion card.	3530
Firmware Identification (FID) Number: SEL-3530-R104-V0-Z001001-D20100305 ACSELERATOR RTAC Software Version: 1.1.xxx Manual Date Code: 20100305	Corrected firmware issue with DNP tunneled serial.	3530
Firmware Identification (FID) Number: SEL-3530-R103-V0-Z001001-D20100218 ACSELERATOR RTAC Software Version: 1.1.xxx Manual Date Code: 20100218	Added SEL-100 and SEL-200 Series Relay support.	3530
	Added insert in Tag Processor.	5033
	Simplified project tree in ACSELERATOR RTAC.	5033
	Added dial-out for DNP server.	3530
	Resolved issue with autoconfigure of certain relays.	3530
	Disabled RTAC before sending project by default.	5033
	Added option to create and convert pre-R103 compatible projects.	5033
Firmware Identification (FID) Number: SEL-3530-R102-V0-Z001001-D20091214 ACSELERATOR RTAC Software Version: 1.0.xxx Manual Date Code: 20091214	Resolved issues in ACSELERATOR RTAC with DDFs and DDF updates.	5033
	Resolved issue with unsolicited write compatibility with SEL-203x devices.	3530
	Resolved issue with SEL-387A autoconfig.	3530
	Added character echo to legacy command transparent connections.	3530
	Resolved issues related to system time and IRIG output signal.	3530
Firmware Identification (FID) Number: SEL-3530-R101-V0-Z001001-D20091102 ACSELERATOR RTAC Software Version: 1.0.xxx Manual Date Code: 20091102	Revised for compatibility with lead-free components.	3530
Firmware Identification (FID) Number: SEL-3530-R100-V0-Z001001-D20090915 ACSELERATOR RTAC Software Version: 1.0.xxx Manual Date Code: 20090915	Initial version.	3530

Table A.2 SEL-2245-2 Firmware Revision History

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-2-R102-V0-Z001001-D20140612	SEL-2245-2-R102 firmware is only compatible with RTAC firmware R132 or higher. ➤ Added support for COMTRADE event reports. ➤ Added Filter C as the default filter.	1.15.xxx	20140714
SEL-2245-2-R101-V0-Z001001-D20121030	➤ Enhanced firmware to provide higher accuracy during electromagnetic disturbances.	1.9.xxx	20130531
SEL-2245-2-R100-V0-Z001001-D20120427	➤ Initial version.	1.9.xxx	20120427

Table A.3 SEL-2245-22 Firmware Revision History

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-22-R101-V0-Z001001-D20171107	➤ Added support for AC Mode metering. ➤ Added 2, 4, 8, 24 kHz options to oscillography capture rates. **Note**: RTAC firmware R141 requires upgrading module firmware to R101. **Note**: SEL-2245-22 modules manufactured with firmware R100 did not receive calibration for AC Mode, and must be returned to the factory to meet the accuracy specifications for ac metering.	1.24.xxx	20171107
SEL-2245-22-R100-V0-Z001001-D20141006	➤ Initial version.	1.16.xxx	20141103

Table A.4 SEL-2245-221 Firmware Revision History

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-22-R102-V0-Z001001-D20181207	➤ Initial version.	1.28.xxx	20181217

Table A.5 SEL-2245-3 Firmware Revision History

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-3-R101-V0-Z001001-D20150123	➤ Enhanced User-Defined Safe Output capability for better performance in certain failure modes.	1.16.xxx	20150123
SEL-2245-3-R100-V0-Z001001-D20140428	➤ Initial version.	1.14.xxx	20140428

Table A.6 SEL-2245-4 Firmware Revision History

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-4-R103-V1-Z001001-D20180629	Includes all the functions of SEL-2245-4-R103-V0 with the following addition: ➤ Enhanced factory calibration procedure.	1.24.xxx	20180710
SEL-2245-4-R103-V0-Z001001-D20171107	➤ Added support for enabling/disabling module tags individually to conserve EtherCAT bandwidth. **Note**: RTAC firmware R141 requires upgrading module firmware to R103.	1.24.xxx	20171107
SEL-2245-4-R102-V0-Z001001-D20150807	➤ Resolved issue which in previous firmware caused the RMS power calculation to be incorrect for delta-connected voltages. ➤ Revised the reactive power quantity to include the sign.	1.17.xxx	20150807
SEL-2245-4-R101-V0-Z001001-D20141006	➤ Updated for IEEE C37.118.1a-2014 compliance.	1.16.xxx	20141103
SEL-2245-4-R100-V0-Z001001-D20130827	➤ Initial version.	1.13.xxx	20130827

Table A.7 SEL-2245-411 Firmware Revision History

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-4-R105-V0-Z001001-D20181207	➤ Added support for Phase/Gain Compensation on LEA inputs.	1.28.xxx	20181217
SEL-2245-4-R104-V0-Z001001-D20180629	➤ Added support for SEL-2245-411 4 CT/4 LEA variant.	1.26.xxx	20180710

Table A.8 SEL-2245-42 Firmware Revision History

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-42-R104-V0-Z001001-D20200731	**Note**: RTAC firmware R147 requires upgrading module firmware to R104-V0. ➤ Added support for Axion Wave Server. ➤ Resolved an issue that prevents back-to-back event recording after 300 event triggers. ➤ Resolved an issue that may prevent event data retrieval after triggering a large number of back-to-back event recordings.	1.31.xxx	20200820
SEL-2245-42-R103-V0-Z001001-D20181207	➤ Improved speed of Event Report collection.	1.28.xxx	20181217
SEL-2245-42-R102-V1-Z001001-D20180523	Includes all the functions of SEL-2245-42-R102-V0 with the following addition: ➤ Resolved an issue that caused the synchrophasor quantity VCA_PM to report as half of the magnitude.	1.24.xxx	20180523
SEL-2245-42-R102-V0-Z001001-D20171107	➤ Enhanced module ADC self-diagnostics.	1.24.xxx	20171107

Firmware Identification (FID) Number	Summary of Revisions	ACSELERATOR RTAC Software Version	Manual Date Code
SEL-2245-42-R101-V0-Z001001-D20170728	➤ Added support for enabling/disabling module tags individually to conserve EtherCAT bandwidth. ➤ Resolved an issue that could cause THD tag to always read zero if the PT or CT ratios were not set to 1.	1.23.xxx	20170728
SEL-2245-42-R100-V1-Z001001-D20170505	➤ Initial version.	1.20.xxx	20170505

ACSELERATOR RTAC

ACSELERATOR RTAC configures the project sent to the RTAC. This software works with all RTAC hardware variants and is backwards compatible with all RTAC firmware versions.

Table A.9 SEL-5033 Software Revision History

ACSELERATOR RTAC Software Version	Summary of Revisions	Manual Date Code
1.32.148.9000	➤ Updated SEL-5033 instruction manual.	20210713
1.32.148.8000	➤ Updated SEL-5033 instruction manual.	20210414
1.32.148.7500	➤ Resolved an issue where projects with nine or more frequency groups cannot be read from an RTAC. ➤ Updated the command line interface to include settable ports for logic engine and project send/read. ➤ Resolved an issue where importing a C37.118 client with multiple PMUs from XML may not succeed.	20210320
1.32.148.7000	➤ Added Simple Tag Mapper Extension. ➤ Added CtPt Monitor Extension. ➤ Added Indirect Tagging Extension. ➤ Enhanced C37.118 PMU Station name to accept non-complaint IEC 61131 names. ➤ Added support to specify a port for logic engine. ➤ Added support to specify a port for project send/read. ➤ Added support for copy and paste operations on the IEC 60870-103 client. ➤ Enhanced password fields to allow a user to view the password while editing before obfuscating the password after the setting is no longer selected. ➤ Resolved an issue where Default Gateway may be incorrect after reading a project that includes interface settings. ➤ Resolved an issue where projects R132 and older with advanced settings included may not be able to open.	20210208
1.31.147.4000	➤ Updated SEL-5033 instruction manual.	20201130
1.31.147.3847	➤ Updated SEL-5033 instruction manual. ➤ Updated Library Extension package to 3.20.10.0.	20201026

ACSELERATOR RTAC Software Version	Summary of Revisions	Manual Date Code
1.31.147.3541	➤ Enhanced Connection Directory to support import and export. ➤ Enhanced the Preferences menu of ACSELERATOR RTAC to improve usability. ➤ Added Radius and Web Proxy to Advanced Settings when reading a project. ➤ When reading a project, the user is now prompted to rename the RTAC project if the name already exists in the database. ➤ Advanced Settings are now editable in RTAC projects. Available settings include Ethernet, Hosts, and Website Description. ➤ Added support for SEL-T401L DDF. ➤ Resolved an issue that may prevent GOOSE POU moving to the Automation thread. ➤ Added licensing information to the Help > About section for WinSCP. ➤ Added improved messaging when software is opened while FIPS is enabled. ➤ Resolved an error that may occur when attempting to install libraries. ➤ Resolved an issue where IEC 61850 configuration information was lost while re-importing IEC 61850 configurations if an IEC 61850 configuration already existed. ➤ Resolved an error message that appeared when attempting to commission a module that did not match the project configuration. ➤ Resolved an error message that appeared when the Ethernet IP Explicit Message client poll period was set to 0. ➤ Resolved an issue where the serial IEEE C37.118 server may not flag an invalid configuration when the number of phasors and message rate exceeded the amount of data that can be transmitted via the configured baud rate. ➤ Resolved an issue where projects could not be sent to RTACs with firmware version R124 or R125.	20200820
1.30.146.4019	➤ Duplicate time sources are now flagged as an error in projects R144 and later. ➤ The IedType attribute is no longer considered when importing IEC 61850 configurations. ➤ Resolved an issue introduced in version 1.30.146.xxx where firmware updates may not succeed for devices with current firmware versions R119–R131.	20200610
1.30.146.3928	➤ Resolved an issue where projects that contain IEC 61850 configurations could cause software version updates to not complete. ➤ Resolved an issue where importing IEC 61850 configurations that contain 64-bit integers may not successfully complete.	20200425
1.30.146.3665	➤ Resolved an issue where station name changes for C37.118 tags did not automatically refactor the previous tag name. ➤ Resolved an issue that could prevent additional users from being added to ACSELERATOR RTAC software. ➤ Resolved an issue where cross task errors could occur when GOOSE transmit messages were in the automation task.	20200325
1.30.146.3437	➤ Updated the ACSELERATOR RTAC software visual interface. ➤ Enhanced the RTAC database to support 64 bits. ➤ Added hot key **<Ctrl+R>** to refactor a tag name. ➤ Enhanced importing IEC 61850 configurations to include SCD and SSD file extensions. ➤ Enhanced importing IEC 61850 process to include interface IP address and MMS poll period and to enable COMTRADE collection as settings from an SCL file. ➤ Added an auto-fill source time and quality column button in the Tag Processor. ➤ Resolved an issue where the clean flag may not remove all files during installation or uninstallation. ➤ Resolved an issue where the Backup Projects process could encounter an error. ➤ Resolved an issue in XML export where BRCB and URCB tags were not exported. ➤ Resolved an issue where Extensions may not work correctly. ➤ Resolved an issue where a project could not be read successfully if an apostrophe was in the RTAC web interface Device Description.	20200224

ICD File Revisions

The ICD file for the RTAC works for all RTAC hardware options. Configurations that use older class file versions can be used in newer firmware versions. This allows for configurations to be updated to the latest firmware without requiring an upgrade of the class file version. To input the new features that are added in newer ICD files, you must reconfigure the RTAC ICD file by using ACSELERATOR Architect® SEL-5032 Software.

Table A.10 ICD Revision Table

Class File Version	Summary of Revisions	Minimum RTAC Firmware
006	➤ Added support for MMS server, added support for Edition 2 compliance.	R135
003	➤ Made modifications to ICD file for Edition 1 certification for MMS client and GOOSE TX/RX.	R119
002	➤ Added support for MMS client.	R115
001	➤ Initial CID file. Support for GOOSE TX and RX only.	R108

Instruction Manual

The date code at the bottom of each page of this manual reflects the creation or revision date.

Table A.11 lists the instruction manual versions and revision descriptions. The most recent instruction manual version is listed first.

Table A.11 Instruction Manual Revision History

Date Code	Summary of Revisions
20210713	Appendix A ➤ Updated for RTAC firmware versions R148-V3 ➤ Added software version information for 1.32.148.9000.
20210414	Section 10 ➤ Updated *Table 10.11: Monitored Channel Settings*. ➤ Updated *Table 10.14: CtPt Monitor Status Structure (struct_CTPTStatusOutput)*. ➤ Updated *Figure 10.42: POU Output Status Indicators*. ➤ Updated *Available Sources Tab* and *Indirect Tagging Extension Example* in *Indirect Tagging*. Appendix A ➤ Updated for RTAC firmware version R148-V2. ➤ Added software version information for 1.32.148.8000.
20210320	Appendix A ➤ Updated for RTAC firmware versions R145-V3, R146-V3, R147-V2, and R148-V1. ➤ Revised R148-V0 firmware summaries for all RTAC models. ➤ Added software version information for 1.32.148.7500.
20210208	General ➤ Updated manual format. Section 2 ➤ Added R148 information to *Overview*. ➤ Added *Event Collection File Naming*.

Date Code	Summary of Revisions
	Section 5 ➤ Added *RAID Support and Configuration*. Section 10 ➤ Added *Simple Tag Mapper*. ➤ Added *CtPt Monitor*. ➤ Added *Indirect Tagging*. Section 11 ➤ Updated *Table 11.2: Self-Test System Tags* with RAID information. Appendix A ➤ Combined the RTAC firmware revision history tables into one (Table A.1). ➤ Updated for RTAC firmware version R148-V0. ➤ Updated for RTAC software version 1.32.xxx. ➤ Added software version information for 1.30.146.4019, 1.31.147.3541, 1.31.147.3847, and 1.31.147.4000.
20201130	Appendix A ➤ Updated for RTAC software version 1.31.xxx. ➤ Revised R147-V1 firmware summaries for all RTAC models. ➤ Revised R144-V8 firmware summaries for all RTAC models. ➤ Revised R143-V0 firmware summary for the SEL-3555. ➤ Revised R104-V0 firmware summary for the SEL-2245-42.
20201026	Appendix A ➤ Updated for RTAC firmware versions R144-V8 and R147-V1.
20200820	Section 2 ➤ Updated *SEL Client in SEL Protocol*. ➤ Added *Axion Wave Server*. ➤ Updated *EtherNet/IP*. Section 5 ➤ Added *Web Proxy*. Section 10 ➤ Updated *FTPSync*. Appendix A ➤ Updated for RTAC firmware version R147-V0.
20200610	Appendix A ➤ Updated for RTAC firmware version R146-V2. ➤ Updated for RTAC firmware version R145-V2. ➤ Updated for RTAC firmware version R144-V7.
20200425	Appendix A ➤ Updated for RTAC firmware version R146-V1. ➤ Updated for RTAC firmware version R144-V6.
20200224	Section 2 ➤ Added *Web API Communications*. Section 7 ➤ Updated *Table 7.6: Available Permissions*. Section 10 ➤ Added *Email Plus*. Appendix A ➤ Updated for RTAC firmware version R146-V0.
20200122	Appendix A ➤ Updated for RTAC firmware version R145-V1. ➤ Updated for RTAC firmware version R144-V5. ➤ Updated for RTAC firmware version R143-V1. ➤ Updated for RTAC firmware version R142-V1.

Date Code	Summary of Revisions
20191015	Appendix A ➤ Updated for RTAC firmware version R144-V4.
20190830	Section 2 ➤ Added *Password Command and Managing User Accounts in SEL Protocol*. ➤ Added *IEEE C37.118 System Configuration* and *IEEE C37.118 Axion PMU* in *IEEE C37.118 Synchrophasors*. ➤ Updated *IEEE C37.118 Client Configuration*. ➤ Added *IEC 60870-5-103*. Section 7 ➤ Added *Static Routes in Ethernet Security*. Section 10 ➤ Added *Report Generator*. ➤ Added *FTPSync*. Appendix A ➤ Updated for RTAC firmware version R145-V0. Appendix B ➤ Updated *Table B.23: INS Attributes*.
20190508	Appendix A ➤ Updated for RTAC firmware version R144-V3.
20190315	Appendix A ➤ Revised R144-V2 firmware summary for SEL-3555 and SEL-3560.
20190216	Appendix A ➤ Updated for RTAC firmware version R144-V2.
20190111	Appendix A ➤ Updated for RTAC firmware version R144-V1.
20181217	Section 1 ➤ Added *Software Maintenance*. Section 2 ➤ Added *SEL-2245-221 Low-Voltage (LEA) Monitoring Module* under *EtherCAT*. ➤ Added *CDC Type 2 Client Configuration* under *CDC Type 2*. Section 7 ➤ Added *Centralized User Accounts With RADIUS*. Section 8 ➤ Updated *Table 8.1: Profile Comparison and RTAC Support*. Appendix A ➤ Updated for RTAC firmware version R144.
20180928	Appendix A ➤ Updated for RTAC software version 1.26.xxx.
20180710	General ➤ Added *Section 11: Testing and Troubleshooting*. ➤ Added *Appendix C: Firmware Upgrade Instructions*. Section 2 ➤ Added *Secure Authentication to DNP3*. ➤ Added *SEL-2245-411 Standard Current and Low-Voltage (LEA) Monitoring Module* under *EtherCAT*. Appendix A ➤ Updated for RTAC firmware version R143.
20180330	Appendix A ➤ Updated for RTAC firmware version R142.

Date Code	Summary of Revisions
20180202	Appendix A ➤ Updated for RTAC versions R139-V2 and R136-V3.
20171201	Appendix A ➤ Added information for RTAC firmware version R141.
20171107	Section 5 ➤ Updated *Web HMI*. Section 6 ➤ Renamed *RTAC Web Interface, ODBC, and Transparent Connections Passwords* to *User Accounts in R100–R140* and moved to *RTAC Web Interface User Accounts*. ➤ Added *Password Report*. ➤ Added *RTAC Web Interface User Accounts*. Appendix A ➤ Updated for RTAC firmware version R141.
20170728	Section 1 ➤ Added *Table 1.1: Typical RTAC Startup Time*. Section 2 ➤ Updated *EtherCAT Recording Group* for support of Custom Analogs. Section 9 ➤ Added *Extensions*. Appendix A ➤ Updated for RTAC firmware versions R139-V1 and R140.
20170531	Section 2 ➤ Updated *Configuring Axion Recording Groups*. Appendix D ➤ Updated *Overview*. ➤ Added *Table D.1: Dynamic Disturbance and Recording System Performance*.
20170505	Section 2 ➤ Updated *SER Server Commands*. ➤ Added *Configuring Axion Recording Groups*. ➤ Added *File Transfer Protocol*. Section 3 ➤ Updated *Table 3.1: Available Columns in the Tag Processor*. Section 5 ➤ Updated *Live Data*. Section 6 ➤ Updated *RTAC Web Interface, ODBC, and Transparent Connections Passwords*. Appendix A ➤ Updated for RTAC firmware version R139. Appendix D ➤ Added new *Section D: Configuring Dynamic Disturbance and Fault Recording Systems*.
20170220	Section 2 ➤ Added *Alstom Event Collection*. Section 5 ➤ Changed section name from *Web HMI and Logging* to *Web HMI and Reports*. ➤ Updated *Overview*. ➤ Renamed *Connecting IED Report* to *Viewing and Troubleshooting Connected IEDs* and updated. ➤ Added *Live Data*. Section 7 ➤ Updated *Table 7.1: Profile Comparison and RTAC Support*.

Date Code	Summary of Revisions
	Appendix A ➤ Updated for RTAC firmware version R138-V0. Appendix B ➤ Updated *Table B.12: BCR Attributes*. ➤ Updated *Table B.37: timeStamp_t Attributes*.
20170109	Section 2 ➤ Updated *Configure Event Collection*. ➤ Updated *Viewing Waveforms and Event Files*. ➤ Added *SEL-2245-42 AC Protection Module*. ➤ Added *CDC Type 2 Server Configuration*. ➤ Added *SNMP Manager/Client Configuration*. Section 6 ➤ Updated *Table 6.1: IP Security Settings*. ➤ Added *Parallel Redundancy Protocol*. Appendix A ➤ Updated for RTAC firmware version R137-V0.
20170108	Appendix A ➤ Updated for RTAC firmware versions R136-V2, R135-V3, and R134-V4.
20161026	Appendix A ➤ Updated for RTAC firmware version R136-V1.
20160624	Section 1 ➤ Updated *Creating a New Project* for the addition of **Import Items** and **Export** to the Quick Access Toolbar. ➤ Added *Upload Projects From the Web Interface* and *Creating and Augmenting Projects With XML*. ➤ Removed *Creating a New Project From XML*. Section 2 ➤ Added *SCADA Protocol Redundancy*, *Interoperability Statement for IEC 60870-5-101 Client for the SEL RTAC*, *Interoperability Statement for IEC 60870-5-104 Client for the SEL RTAC*, and *eDNA*. ➤ Updated *DNP3*, *Modbus*, *IEEE C37.118 Synchrophasors*, *EtherCAT*, *Interoperability Statement for IEC 60870-5-101 Server for the SEL RTAC*, and *Interoperability Statement for IEC 60870-5-104 Server for the SEL RTAC*. ➤ Updated *SEL Protocol* and added the following subsections: *Remote Access and File Transfer*, *Multiple Simultaneous Connections Support*, *Primary SEL Protocol Connection Configuration*, and *Unsolicited Event and SOE Notification*. ➤ Updated *IEC 61850* and added the following subsections: *MMS Client File Services*, *COMTRADE Collection*, and *IEC 61850 MMS Server File Services*. ➤ Added *IEC 60870 101/104 Client Configuration* to *IEC 60870-5-101 and -104*. Section 6 ➤ Updated *Passwords and User Accounts* and added the following subsections: *Protocol Passwords*, *Project Passwords*, and *Project Export Encryption*. Section 7 ➤ Added *Precision Time Protocol (PTP)*. ➤ Updated *Time Source Settings*. Appendix A ➤ Updated for RTAC firmware version R136-V0. Appendix B ➤ Updated *IEC Operators and Extending Functions* and added the following subsections: *Custom Conversions Functions* and *Time Functions*. ➤ Added information to *Predefined Function Blocks*.
20160212	Appendix A ➤ Updated for RTAC firmware versions R135-V2, R134-V3, R133-V1, and R132-V1.
20160201	Appendix A ➤ Updated for IEC 60870 issue in ACSELERATOR RTAC.

Date Code	Summary of Revisions
20160115	Appendix A ➤ Updated for RTAC firmware version R135-V1. ➤ Updated to include note about hardware compatibility with firmware versions R134 and earlier.
20150904	Section 2 ➤ Updated *IEC 61850* section to include *MMS server*. Section 3 ➤ Updated information on the Alarm Summary and IED Report. Section 5 ➤ Improved the description for web session time-out usage. Appendix A ➤ Updated for RTAC firmware version R135-V0.
20150807	Appendix A ➤ Updated for SEL-2245-4 firmware version R102.
20150504	Section 1 ➤ Added references to SEL-3555. ➤ Updated information related to the **Tools** menu item. Section 2 ➤ Added references to SEL-3555. ➤ Replaced *Figure 2.41–Figure 2.44* and *Figure 2.86–Figure 2.87* to include more detail. Appendix A ➤ Updated for RTAC firmware version R134-V1 (R134-V2 for SEL-3532). Appendix C ➤ Replaced *Figure C.3: Direct-Acting and Reverse-Acting Processes*, *Figure C.8: Cascade Control Example*, *Figure C.9: Configuration Example of Primary PID Controller*, and *Table C.4: Cascade Operating Modes* to include more detail.
20150311	Section 2 ➤ Updated maximum size of tag names to 100. Section 2 ➤ Updated IEC 60870-5-101 and -104 interoperability tables to reflect double transmission and new data rates. ➤ Added *Flex Parse Protocol*. Section 7 ➤ Added new Global UTC time offset setting that provides more simple time configuration. Appendix A ➤ Updated for RTAC firmware version R134. ➤ Added *Table A.11: SEL-3555 Firmware Revision History*.
20150123	Section 2 ➤ Updated Safe Output information in *Configuring the DC Analog Output Module* in *EtherCAT*. Appendix A ➤ Updated *Table A.4: SEL-2245-3 Firmware Revision History*.
20141103	Section 2 ➤ Updated *EtherCAT* for addition of 4AI-ER module. Appendix A ➤ Updated for RTAC firmware version R133. ➤ Added *Table A.3: SEL-2245-22 Firmware Revision History*.
20140908	Appendix A ➤ Updated ACSELERATOR RTAC enhancements.

Date Code	Summary of Revisions
20140714	Section 2 ➤ Added note about correct usage of power coupler EtherCAT ports. ➤ Added EtherCAT Filter C information. ➤ Added *Waveform Recording* and *Inputs* subsections to *SEL-2245-2 16 DC Analog Input Module*. Appendix A ➤ Updated for RTAC firmware version R132.
20140616	Section 1 ➤ Updated *Creating a New Project* subsection. ➤ Updated *Figure 1.3: ACSELERATOR RTAC Project Creation Interface*. ➤ Updated *Figure 1.5: Application Button and Quick Access Toolbar*. ➤ Removed *Figure 1.7: New From XML Window*. ➤ Updated *Startup Switches* subsection. Section 2 ➤ Added *CP2179* subsection. ➤ Updated *Table 2.56: RTU Templates With Included Modules*. ➤ Added *Interoperability Statement for IEC 60870-5-101 Server for the SEL Real-Time Automation Controller (RTAC)* and *Interoperability Statement for IEC 60870-5-104 Server for the SEL Real-Time Automation Controller (RTAC)* subsections. Section 6 ➤ Updated *Table 6.1: IP Security Settings*. ➤ Updated *Figure 6.2: Ethernet Settings* and *Figure 6.3: Routes and Certificates*. ➤ Added *Syslog* subsection. Section 7 ➤ Updated *Table 7.2: Pertinent System POU Pin Settings*. Appendix A ➤ Added *Table A.6: SEL-3505-3 Firmware Revision History* and *Table A.9: SEL-3532 Firmware Revision History*.
20140428	Section 1 ➤ Added *Table 1.1: Advanced Debugging Tools*. Section 2 ➤ Updated *EtherCAT* for addition of AO module. ➤ Updated *Figure 2.83: New Pulse Config offDur = 0*. Appendix A ➤ Updated for RTAC firmware version R131. ➤ Added *Table A.3: SEL-2245-3 Firmware Revision History*. Appendix B ➤ Updated *Table B.7: Integer Data Types* and *Table B.16: DPS Attributes*. Appendix C ➤ Added new *Appendix C: Proportional-Integral-Derivative (PID) Function Block*.
20140225	Appendix A ➤ Updated for RTAC firmware version R130.
20140210	Appendix A ➤ Updated for RTAC firmware version R129.
20140113	Appendix A ➤ Updated for RTAC firmware version R128.
20131216	Appendix A ➤ Updated for RTAC firmware version R127.

Date Code	Summary of Revisions
20131206	Section 1 ➤ Updated *Figure 1.5: Application Button and Quick Access Toolbar*. ➤ Added **Export Items** option to *Application Button*. ➤ Updated *Save and Download a Project* to include IEC 61131 logic information. ➤ Updated export instructions in *Creating a New Project From XML*. ➤ Removed *Test Mode* section in *Startup Switches*. Section 2 ➤ Added *Output Contact Pulse* section to *EtherCAT*. ➤ Updated *Figure 2.84: Digital output Module Tag Attributes* and *Figure 2.85: Tag Processor for Trip-Close Pair Example*. ➤ Updated *Example 2.8: Configure a Transmit and a Receive NGVL on two RTACs*. Section 6 ➤ Added maximum length specification for password in *Ethernet Security*. ➤ Added information about X.509 certificate activation in *Ethernet Security*. Section 8 ➤ Added new_project_version information in *Variables* section in *IEC 61131-3 Programming Overview*. Appendix A ➤ Updated for RTAC firmware version R126. Appendix B ➤ Added Q1 output information in RS and SR function block sections in *Predefined Function Blocks*. ➤ Added origin attribute to *Table B.8: APC Attributes*, *Table B.14: DNPC Attributes*, *Table B.19: INC Attributes*, *Table B.24: MDBC Attributes*, *Table B.25: MRBC Attributes*, *Table B.27: SPC Attributes*, *Table B.28: SBRC Attributes*, and *Table B.30: RBC Attributes*. ➤ Added *DPC (Controllable Double Point)* section in *Data Types*. ➤ Added operPulse attribute to *Table B.21: IOC Attributes*.
20130827	Section 2 ➤ Added *IEEE C37.118 Server* and *IEEE C37.118 Server PMU* to protocols section. ➤ Added *SEL-2245-4 CT/PT Module* to the *EtherCAT* protocol section. Appendix A ➤ Updated for RTAC firmware version R125.
20130628	Section 1 ➤ Updated project view descriptions in *Creating a New Project*. ➤ Added *Startup Switches* section that discusses advanced startup switches to facilitate library support and advanced development options. Section 2 ➤ Update max tag count to 5000 for SEL-3505. ➤ Added *SES-92* server protocol section. ➤ Added *IEC 60870-5-101 and -104* server protocols section. Section 4 ➤ Added APR commands and controls information in *Configuration*. ➤ Added *Table 4.1: APR Commands Summary*. ➤ Added *POU Pin Settings* section. Section 5 ➤ Added clarifications for logging various data types in *Log Settings*. Section 7 ➤ Added Force_System_Time_Source to *Table 7.2: Pertinent System Time POU Pin Settings*. Section 8 ➤ Added *Cross References* in *IEC 61131-3 Programming Overview* with information about using cross referencing to find all instances of a tag or variable in the project.

Date Code	Summary of Revisions
	Appendix A ➤ Updated for RTAC firmware version R124. **Appendix B** ➤ Added several new data types in *ACSELERATOR RTAC Data Types* and Data Classes for IEC 60870-5-101 protocol.
20130531	**Appendix A** ➤ Updated for RTAC firmware version R123 Z002. ➤ Updated for SEL-2245-2 firmware version R101.
20130207	**Appendix A** ➤ Updated for ACSELERATOR RTAC revision 1.11.41xx.xxx.
20130117	**Section 2** ➤ Added note regarding anonymous client IP addressing when using unsolicited event reporting in SEL server. **Appendix A** ➤ Updated for RTAC firmware version R123.
20121121	**Section 1** ➤ Added definition for a tag. ➤ Added information on how to create a shared ACSELERATOR RTAC database. **Section 2** ➤ Clarified that the SEL-3505 supports 10000 tags total. ➤ Clarified event collection parameter usage. ➤ Clarified MMS DataSet maximum is per IED. ➤ Modified GOOSE TX and RX maximums to 150 each. ➤ Added MICS and PICS documents in *IEC 61850*. ➤ Added analog input module to *EtherCAT*. ➤ Added *Landis and Gyr LG 8979*. **Section 4** ➤ Modified *Example 4.3* to use an SEL-451 instead of an SEL-351S. **Section 5** ➤ Added ODBC configuration steps for Windows 7. **Section 6** ➤ Fixed typo in LDAP configuration that referred to SEL-5033 instead of RTAC as the LDAP client. ➤ Added configuration information for certificate usage in *LDAP Configuration*. **Section 8** ➤ Added note on referencing function blocks each processing cycle. ➤ Modified figure in *Example 8.2* to reflect example description. **Appendix A** ➤ Added entry for R119 firmware regarding Modbus/TCP controls. ➤ Updated for firmware version R122.
20121031	**Appendix A** ➤ Updated for firmware version R121.
20121010	**Appendix A** ➤ Amended R120 firmware revision history.
20120924	**Appendix A** ➤ Updated for firmware version R120.
20120720	**General** ➤ Included SEL-3505 as newest member of the RTAC family. **Section 1** ➤ Update *Figure 1.3* to include SEL-3505.

Date Code	Summary of Revisions
	Section 2 ➤ Added note to exclude MMS, EtherCAT options from SEL-3505. ➤ Modified max DNP servers to from 32 to 100. **Section 3** ➤ Included information about new web-based tag visualization. **Section 5** ➤ Added note to exclude HMI option from SEL-3505. ➤ Updated figures to show SOE filtering features. **Appendix A** ➤ Updated for firmware version R119.
20120427	**Section 2** ➤ Revised for addition of 16 analog inputs and 10 FHC digital output modules. **Appendix A** ➤ Updated for firmware version R118.
20120423	**Appendix A** ➤ Updated for ACSELERATOR RTAC maintenance release.
20120316	**Appendix A** ➤ Updated for firmware version R117.
20120301	**Appendix A** ➤ Updated for firmware version R116.
20120208	**Section 1** ➤ Added information about configuring connection configurations per RTAC. ➤ Enhanced description of Controller function block. **Section 2** ➤ Enhanced description of DNP failover methods. ➤ Enhanced description of Modbus register variations. ➤ Removed *GOOSE*. ➤ Added *IEC 61850* to include MMS client, GOOSE and IEC 61850 overview. ➤ Inserted *Network Global Variables (NGVLs)*. **Section 3** ➤ Updated figures in *Example 3.1*. **Section 4** ➤ Added *Example 4.3* to illustrate capturing characters in source access point receive messages. **Section 8** ➤ Clarified discussion of GLOBAL RETAIN variables. **Appendix A** ➤ Updated for RTAC firmware version R115. **Appendix B** ➤ Corrected status type in *Table B.14*.
20111010	**Section 1** ➤ Enhanced explanation of task cycle time behavior. **Section 2** ➤ Added explanation of DNP server failover and port sharing. ➤ Added description of DNP server event buffers configuration. ➤ Described SEL client unsolicited write tag quantities. **Section 3** ➤ Added example illustrating how to access individual bits in a tag. **Section 5** ➤ Added logging function block names to logging description.

Date Code	Summary of Revisions
	Section 6 ➤ Added information regarding HMI Operator user role. Appendix A ➤ Updated for RTAC firmware version R114.
20110721	➤ Initial version.

APPENDIX B

IEC 61131-3 Programming Reference

Overview

This section provides details about the following subjects related to IEC 61131-3 programming standards:

- *IEC Operands on page 675*: Inputs or arguments used in logic statements.

  myNumber := **4** + **4**;

- *IEC Operators and Extending Functions on page 681*: Functions used on operands in logic statements to produce a result.

  myNumber := 4 **+** 4;

- *Predefined Function Blocks on page 722*: A library of commonly used routines you can use in custom logic.

  TON() is a turn on delay function block.

- *Instructions on page 737*: Logic statements that perform actions.

  FOR, WHILE, IF, etc.

- *Declarations on page 743*: Statements that define operands and their characteristics.

  VAR, VAR_INPUT...

- *Data Types on page 747*: Standard predefinitions of operand types.

  myNumber : INT;

Other resources are widely available for the IEC 61131-3 programming standard. Note that the Real-Time Automation Controller (RTAC) only supports Structured Text (ST), Ladder Diagram (LD), and Continuous Function Chart (CFC) programming methods.

IEC Operands

An operand is any input or argument in logic statements. Use the following as operands in RTAC-defined logic:

- *Constants on page 676*
- *Variables on page 680*
- *Functions on page 681*

Constants

A constant operand is a value in a logic program that does not change dynamically.

```
myNumber := 2 + x;
```

The number 2 here is a constant. Regardless of what else happens in the logic, 2 will remain 2.

Use the following data types to create constants as operands in IEC 61131-3 logic.

➤ *BOOL Constants on page 676*
➤ *TIME Constants on page 676*
➤ *DATE Constants on page 677*
➤ *TIME_OF_DAY Constants on page 678*
➤ *DATE_AND_TIME Constants on page 678*
➤ *Number Constants on page 678*
➤ *REAL Constants on page 679*
➤ *STRING Constants on page 679*
➤ *Typed Literals on page 680*

BOOL Constants

BOOL (Boolean) constants are the logical values TRUE and FALSE or 1 and 0, respectively. A digital input state is an example of a Boolean value. Although only a single bit is necessary (1 or 0), each BOOL constant uses eight bits of memory space.

```
IF (IED_DNP.BI_0000.stVal = TRUE) THEN
```

IED_DNP.BI_0000.stVal is compared to the constant value of TRUE, or 1 in this example.

See also data type *BOOL on page 748*.

TIME Constants

Generally, you will use TIME constants to operate the standard timer modules. Besides the 32-bit IEC 61131-3 time constant TIME, LTIME is also supported. This extension to the standard provides a time base for high-resolution timers. LTIME is 64 bits with a resolution in nanoseconds.

Syntax for TIME constant:

```
t#<time declaration>
```

You are not limited to using the format t#; you can also use T#, time, or TIME.

Include the following time units in the time declaration. You must use the time units in the following sequence, but not all time units are necessary.

➤ d: days
➤ h: hours

- m: minutes
- s: seconds
- ms: milliseconds

Examples of correct TIME constants in a Structured Text assignment (comments shown in parentheses to the right):

TIME1 := T#14ms;

TIME1 := T#100S12ms; (* The highest component may be allowed to exceed its TIME1 := T#12h34m15s; limit *)

The following would be incorrect:

TIME1 := t#5m68s; (* limit exceeded in a lower component *)

TIME1 := 15ms; (* T# is missing *)

TIME1 := t#4ms13d; (* Incorrect order of entries *)

Syntax for LTIME constant:

LTIME#<time declaration>

In the time declaration, you can include the time units you used previously with the TIME constant and add the following:

- us: microseconds
- ns: nanoseconds

Examples of correct LTIME constants in a Structured Text assignment:

LTIME1 := LTIME#1000d15h23m12s34ms2us44ns

LTIME1 := LTIME#2ns

LTIME1 := LTIME#3445343m3424732874823ns

See also *Time Data Types on page 749*.

DATE Constants

Use these constants to enter dates.

Syntax:

d#<date declaration>

You are not limited to using the format d#; you can also use the following: D#, date, DATE.

Enter the date declaration in the format <year-month-day>.

Internally, DATE values are assigned as DWORD values.

Examples:

DATE#1996-05-06

d#1972-03-29

See also *Time Data Types on page 749*.

TIME_OF_DAY Constants

Use this type of constant to store times of the day.

Syntax:

tod#<time declaration>

You are not limited to using the format tod#; you can also use the following: TOD#, time_of_day, TIME_OF_DAY.

Enter the time declaration in the format <hour:minute:second>.

You can enter seconds as real numbers; you can therefore specify fractions of a second.

Internally, TIME_OF_DAY values are assigned as DWORD values that contain the time in seconds since 00:00.

Examples:

TIME_OF_DAY#15:36:30.123 tod#00:00:00

See also *Time Data Types on page 749*.

DATE_AND_TIME Constants

You can combine DATE constants and TIME_OF_DAY constants to form DATE_AND_TIME constants.

Syntax:

dt#<date and time declaration>

You are not limited to using the format dt#; you can also use the following: DT#, date_and_time, DATE_AND_TIME.

Enter the date and time declaration in the format <year-month-day-hour:minute:second>. You can enter seconds as real numbers; you can therefore specify fractions of a second.

Internally DATE_AND_TIME values are assigned as DWORD values that contain the number of seconds since 01.01.1970, 00:00.

Examples:

DATE_AND_TIME#1996-05-06-15:36:30 dt#1972-03-29-00:00:00

See also *Time Data Types on page 749*.

Number Constants

Number constant values can be in binary, octal, decimal, and hexadecimal format. If an integer value is not in decimal (base-10) format, you must declare the base followed by the number sign (#) in front of the integer constant.

Always represent the values for the decimal numbers 10–15 in hexadecimal format by the letters A–F.

You can include the underscore character within the number.

Table B.1 Examples of Number Constants

14	(decimal or base-10 number)
2#1001_0011	(binary or base-2 number)
8#67	(octal or base-8 number)
16#A	(hexadecimal or base-16 number)

These number values can be of type BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, or LREAL.

The compiler does not permit implicit conversions from larger to smaller variable types. This means that you cannot substitute a DINT variable for an INT variable. You must use a type conversion function such as described in *Type Conversion Functions on page 693*.

REAL Constants

Represent REAL and LREAL constants either as decimal fractions or exponentially. Use a period for the decimal point.

Table B.2 Examples of REAL Constants

7.4	instead of 7,4
1.64E+009	instead of 1,64e009

STRING Constants

A string is a sequence of characters.

Surround all STRING constants with single quotation marks. You may also enter blank spaces and special characters (umlauts, for instance). The compiler will treat these characters similarly to all other characters.

Using the dollar sign "$" in string constants will inform the software that the constant immediately following has special meaning. Note the following combinations and how the string constants change from face value.

Table B.3 Examples of STRING Constants

Entered Combination	Interpretation
$<two hex numbers>	hexadecimal representation of the eight bit character code
$$	Dollar sign
'	Single quotation mark
$L or $l	Line feed
$N or $n	New line
$P or $p	Page feed
$R or $r	Line break
$T or $t	Tab

Examples:

' w1Wüβ?'

' Abby and Craig '

':-)'

'costs ($$)'

Typed Literals

The compiler assigns IEC constants to the smallest possible data type by default. If you must assign a data type explicitly to a constant, then you will use typed literals. In these cases, you must provide the constant with a prefix that determines the type.

Syntax:

<Type>#<Literal>

<Type> specifies the data type you want; possible entries are: BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, LREAL. Identify the type with uppercase letters.

<Literal> specifies the constant. The data you enter must fit within the data type you specify in <Type>.

Example:

var1 := DINT#34;

If the compiler cannot convert the constant to the target type without data loss, it issues an error message.

You can use typed literals wherever you can use normal constants.

Variables

A variable has a unique name and specific data type, and it provides reference to a value stored in memory. Declare variables either locally in the declaration part of a program organizational unit (POU) or in a global variable list (GVL). Use the data type editor (DTE) to declare a structure variable.

You can use variables anywhere the declared type allows for them. You can access available variables through the Input Assistant.

You can reference various types of variables in different ways:

Two-dimensional array components:

<Fieldname>[Index1, Index2]

Structure variables:

<StructureName>.<VariableName>

Function Block and Program variables accessed by another program:

<FunctionBlockName>.<VariableName>

Functions

In Structured Text, a function call can also appear as an operand. The result returned from that function is used in the context of the function call.

Example: (Result is assigned the value of Fct(7) added to 3)

```
Result := Fct(7) + 3;
```

IEC Operators and Extending Functions

ACSELERATOR RTAC® SEL-3530 Software supports all IEC 61131-3 operators. These operators are recognized implicitly throughout the project.

Use operators in a POU similarly to the way that you use a function. See the following categories of operators:

Assignment Operators

On the left side of an assignment there is an operand (variable) to which the assignment operator := assigns a value for the expression on the right side of the assignment.

See the *MOVE* operator, which performs the same function, especially in CFC-type POUs.

Also see the *SET* and *RESET* operators.

Example:

```
var1 := var2 * 10;
```

After completion of this line, var1 has the tenfold value of var2.

MOVE

Operator: Assignment of a variable to another variable of an appropriate type.

MOVE is available as a box in the CFC and LD editor. You can also add or remove the EN/ENO functionality on a variable assignment by right-clicking on the MOVE box and selecting EN/ENO from the list.

Only if en_i is TRUE, var1 will be assigned to var2.

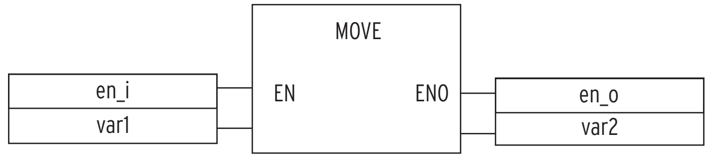


Figure B.1 MOVE Example in CFC or LD in Conjunction With the EN/ENO Function

Example in Structured Text:

```
var2 := MOVE(var1);
```

(you get the same result with: var2 := var1;)

RESET

Standard IEC 61131-3 does not prescribe this operator.

Operator: RESET R forces a RESET or FALSE on an output when an input goes TRUE. The output element will retain the value of FALSE. In LD logic, use the ladder element Reset coil.

Allowed types: BOOLEAN.

Example in Structured Text:

```
var1 R= var2;
```

This logic will reset Boolean element var1 to a value of FALSE if var2 becomes TRUE. Once RESET, var1 will remain FALSE independently of the value of var2.

Example:

```
var1 S= var2 R= fun1(par1,par2);
```

In this case, var2 gets the reset output value of fun1, but var1 gets the set output value of fun1 and does not get the set value of var2.

⚠IMPORTANT
Notice the behavior of the software in case of a multiple assignment. All set and reset assignments refer to the last member of the assignment.

SET

Standard IEC 61131-3 does not prescribe this operator.

Operator: SET S forces a SET or TRUE on an output when an input goes TRUE. The output element will retain the value of TRUE. In LD logic, use the ladder element Set coil.

Allowed types: BOOLEAN.

Example in Structured Text:

```
var1 S= var2;
```

This logic will set Boolean element var1 to a value of TRUE if var2 becomes TRUE. Once SET, var1 will remain TRUE independently of the value of var2.

Arithmetic Operators

The following operators, which the IEC 61131-3 standard prescribes, are available: ADD, MUL, SUB, DIV, MOD.

The extending operator SIZEOF is also available.

ADD

Operator: Addition of variables. In LD, you can also select two or three inputs.

Allowed types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, and LREAL.

You can also add two TIME variables to produce another time data type (e.g., t#45s + t#50s = t#1m35s).

Example in Structured Text:

```
var1 := 7 + 2 + 4 + 7;
```

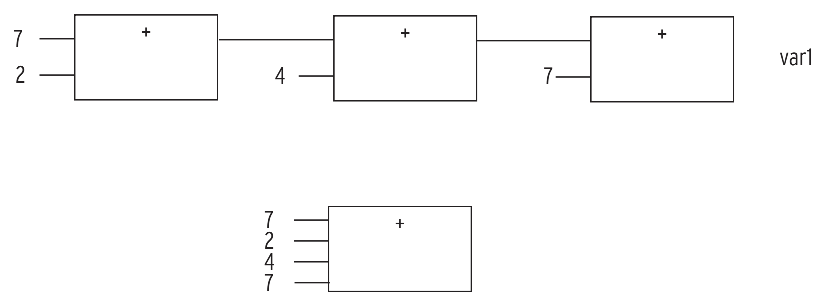


Figure B.2 Two ADD Examples in CFC, LD

DIV

Operator: Division of one variable by another.

Allowed types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, and LREAL.

Example in Structured Text:

```
var1 := 8/2; (* Result in var1 is 4 *)
```

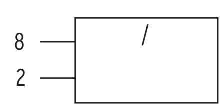


Figure B.3 DIV Example in CFC, LD

MOD

Operator: Modulo Division of one variable by another. The result of this function will be the remainder of the division expressed as a whole number.

Allowed types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT.

Example in Structured Text:

var1 := 9 MOD 2 (* Result in var1 is 1 *)

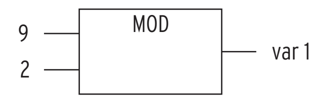


Figure B.4 MOD Example in CFC, LD

MUL

Operator: Multiplication of variables.

Allowed types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, and LREAL.

Example in Structured Text:

var1 := 7 * 2 * 4 * 7;

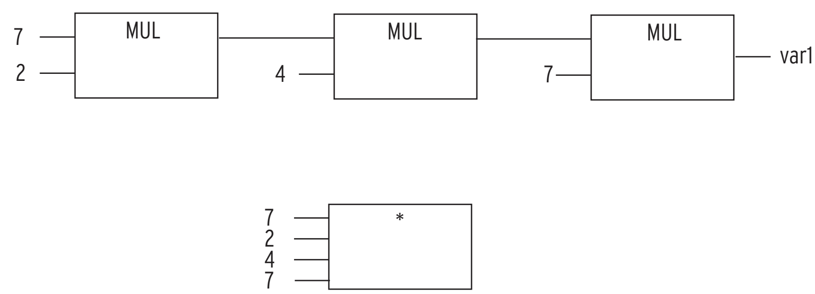


Figure B.5 Two MUL Examples in CFC, LD

SUB

Operator: Subtraction of one variable from another.

Allowed types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, and LREAL.

You can also subtract a TIME variable from another TIME variable to produce a third TIME type variable. Note that negative TIME values are undefined.

Example in Structured Text:

var1 := 7 – 2;

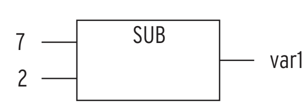


Figure B.6 SUB Example in CFC, LD

Bitstring Operators

The following bitstring operators are available, matching the IEC 61131-3 standard: AND, OR XOR, NOT.

Bitstring operators compare the corresponding bits of two or several operands.

AND

Bitstring Operator: Bitwise AND of bit operands. If the input bits each are 1, then the resulting bit will be 1. Otherwise, the resulting bit will be 0.

Allowed types: BOOL, BYTE, WORD, DWORD.

Example in Structured Text: (* Result in var1 is 2#1000_0010 *) var1

var1 : BYTE;

var1 := 2#1001_0011 AND 2#1000_1010

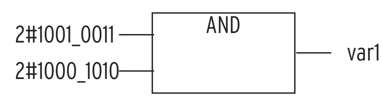


Figure B.7 AND Example in CFC, LD

NOT

Bitstring Operator: Bitwise NOT operation of a bit operand. The resulting bit will be 1, if the corresponding input bit is 0. The resulting bit will be 0, if the corresponding input bit is 1.

Allowed types: BOOL, BYTE, WORD, DWORD.

Example in Structured Text: (* Result in var1 is 2#0110_1100 *)

var1 : BYTE;

var1 := NOT 2#1001_0011

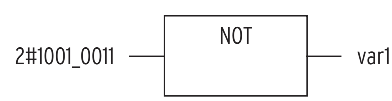


Figure B.8 NOT Example in CFC, LD

OR

Bitstring Operator: Bitwise OR of bit operands. If at least one of the input bits is 1, the resulting bit will be 1. Otherwise, the resulting bit will be 0.

Allowed types: BOOL, BYTE, WORD or DWORD.

Example in Structured Text: (* Result in var1 is 2#1001_1011 *)

var1 : BYTE;

var1 := 2#1001_0011 OR 2#1000_1010

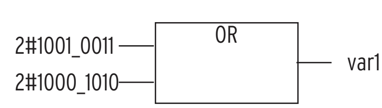


Figure B.9 OR Example in CFC, LD

XOR

Bitstring Operator: Bitwise XOR operation of bit operands. If only one of the input bits is 1, then the resulting bit will be 1. If both or none are 1, the resulting bit will be 0.

Allowed types: BOOL, BYTE, WORD, DWORD.

Example in Structured Text: (* Result is 2#0001_1001 *)

var1 : BYTE;

var1 := 2#1001_0011 XOR 2#1000_1010

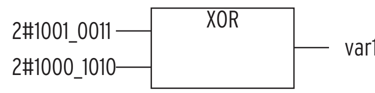


Figure B.10 XOR Example in CFC, LD

NOTE
It is important to understand the behavior of the XOR function in extended form (cases where you have more than two inputs). The logic engine checks the inputs in pairs and then compares the results again in pairs (this complies with the standard, but this behavior may not be anticipated). To ensure intended operation, group input comparisons within parentheses.

Bit-Shift Operators

The following bit-shift operators, matching the IEC 61131-3 standard, are available: SHL, SHR, ROL, ROR.

NOTE
The software derives the number of bits it uses for the bit-shift operation from the data type of the input variable. If the input variable is a constant, the software uses the smallest possible data type. The data type of the output variable has no effect on the arithmetic operation.

The examples of bit-shift operations are given in hexadecimal notation. These examples illustrate how outputs can vary depending on the data type of the input variable, even though the values of the input variables are the same.

ROL

Operator: Bitwise rotation of an operand to the left.

erg := ROL (in, n)

Allowed data types: BYTE, WORD, DWORD.

in will be shifted one bit position to the left *n* times while the bit that is farthest to the left (the one shifted out) will be reinserted from the right.

Example in Structured Text:

PROGRAM rol_st

VAR

in_byte : BYTE := 16#45;

```
in_word : WORD := 16#45;

erg_byte : BYTE;

erg_word : WORD;

n : BYTE := 2;

END_VAR

erg_byte := ROL(in_byte,n); (* Result is 16#15 *)

erg_word := ROL(in_word,n); (* Result is 16#0114 *)
```

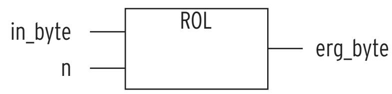


Figure B.11 ROL Example in CFC, LD

ROR

Operator: Bitwise rotation of an operand to the right.

```
erg := ROR (in, n)
```

Allowed data types: BYTE, WORD, DWORD.

in will be shifted one bit position to the right *n* times while the bit that is farthest to the right (the one shifted out) will be reinserted from the left.

Example in Structured Text:

```
PROGRAM ror_st VAR

in_byte : BYTE := 16#45; in_word :

WORD := 16#45;

erg_byte : BYTE;

erg_word : WORD;

n : BYTE := 2;

END_VAR

erg_byte := ROR(in_byte,n); (* Result is 16#51 *) erg_word

:= ROR(in_word,n); (* Result is16#4011 *)
```

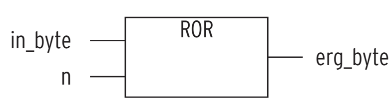


Figure B.12 ROR Example in CFC, LD

SHL

Operator: Bitwise left shift of an operand.

```
erg := SHL (in, n)
```

in: Operand to be shifted to the left.

n: Number of bits by which in gets shifted to the left.

SHL populates least significant bits with zeros for each bit shifted to the left.

Allowed data types: BYTE, WORD, DWORD.

Example in Structure Text:

```
PROGRAM shl_st
VAR
in_byte : BYTE := 16#45;
in_word : WORD := 16#45;
erg_byte : BYTE;
erg_word : WORD;
n : BYTE := 2;
END_VAR
erg_byte := SHL(in_byte,n); (* Result is 16#14 *)
erg_word := SHL(in_word,n); (* Result is 16#0114 *)
```

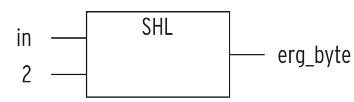


Figure B.13 SHL Example in CFC, LD

SHR

Operator: Bitwise right shift of an operand.

```
erg := SHR (in, n)
```

in: Operand to be shifted to the right.

n: Number of bits by which in will shift to the right.

Allowed data types: BYTE, WORD, DWORD.

For unsigned operands (e.g., BYTE, WORD), SHR populates most significant bits with zeros for each bit shifted right. For operands of signed data types (e.g., INT), SHR replicates the sign bit into the most significant bits for each bit shifted right.

Example in Structured Text:

```
PROGRAM shr_st
VAR
in_byte : BYTE := 16#45;
in_word : WORD := 16#45;
erg_byte : BYTE;
erg_word : WORD;
n : BYTE := 2;
```

END_VAR

erg_byte := SHR(in_byte,n); (* Result is 16#11 *)

erg_word := SHR(in_word,n); (* Result is 16#0011 *)

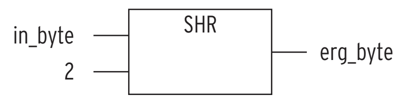


Figure B.14 SHR Example in CFC, LD

Selection Operators

You can perform all selection operations also with variables. For clarity, we limit the following examples to the use of constants as operators: SEL, MAX, MIN, LIMIT, MUX.

LIMIT

Selection Operator: Limiting.

OUT := LIMIT(Min, IN, Max) means:

Max is the upper limit, and *Min* is the lower limit for the result. Should the value *IN* exceed the upper limit *Max*, LIMIT will return *Max*. Should *IN* fall below *Min*, the result will be *Min*.

Allowed data types: IN and OUT can be any type of variable.

Example in Structured Text:

var1 := LIMIT(30,90,80); (* Result is 80 *)

MAX

Selection Operator: Maximum function. Returns the greater of the two values.

OUT := MAX(IN0, IN1)

Allowed data types: IN0, IN1, and OUT can be any type of variable.

Example in Structured Text:

var1 := MAX(30,40); (* Result is 40 *)

var1 := MAX(40,MAX(90,30)); (* Result is 90 *)

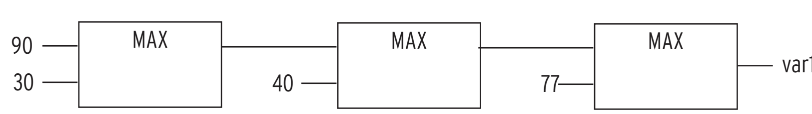


Figure B.15 MAX Example in CFC, LD

MIN

Selection Operator: Minimum function. Returns the lesser of the two values.

OUT := MIN(IN0, IN1)

Allowed data types: IN0, IN1, and OUT can be any type of variable.

Example in Structured Text:

var1 := MIN(90,30); (* Result is 30 *)

var1 := MIN(MIN(90,30),40); (* Result is 30 *)

Figure B.16 MIN Example in CFC, LD

MUX

Selection Operator: Multiplexer

OUT := MUX(*K*, IN0, ..., IN*n*)

MUX selects the *K*th value from among a group of values. MUX counts the first element as element number 0, the second element as number 1, etc.

Allowed data types: IN0, ..., IN*n* and OUT can be any type of variable.

K must be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, or UDINT.

NOTE
Any element in the list that is an expression rather than a variable or constant will only be processed if selected by the value of K.

Examples in Structured Text:

var1 := MUX(0,30,40,50,60,70,80); (* Result is 30 *)

var1 := MUX(3,30,40,50,60,70,80); (* Result is 60 *)

SEL

Selection Operator: Binary Selection. G determines whether IN0 or IN1 is assigned to OUT.

OUT := SEL(G, IN0, IN1) means:

OUT := IN0; if G = FALSE

OUT := IN1; if G = TRUE.

Allowed data types: IN0, IN1, OUT: any type. G must be BOOL.

Example in Structured Text:

var1 := SEL(TRUE,3,4); (* Result is 4 *)

NOTE
If IN0 or IN1 are expressions, rather than simple variables or constants, the software will process only one of the two expressions according to the value of G.

Figure B.17 SEL Example in CFC, LD

Comparison Operators

The following operators, matching the IEC 61131-3 standard, are available: GT, LT, LE, GE, EQ, and NE.

These are Boolean operators, each of which compares two inputs (first and second operand).

EQ

Comparison Operator: Equal to.

A Boolean operator that returns the value TRUE when the operands are equal.

Allowed data types: VAR1 must be BOOL.

Other operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, and STRING.

Example in Structured Text: (* Result is TRUE *)

```
VAR1 := 40 = 40;
```

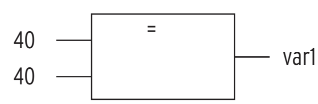


Figure B.18 EQ Example in CFC, LD

GE

Comparison Operator: Greater than or equal to.

A Boolean operator that returns the value TRUE when the value of the first operand is greater than or equal to the value of the second.

Allowed data types: VAR1 must be BOOL.

Other operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, and STRING.

Example in Structured Text: (* Result is TRUE *)

```
VAR1 := 60 >= 40;
```

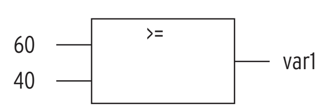


Figure B.19 GE Example in CFC, LD

GT

Comparison Operator: Greater than.

A Boolean operator that returns the value TRUE when the value of the first operand is greater than the value of the second.

Allowed data types: VAR1 must be BOOL.

Other operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, and STRING.

Example in Structured Text: (* Result is FALSE *)

```
VAR1 := 20 > 30;
```

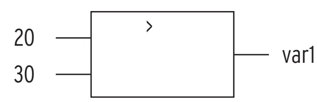


Figure B.20 GT Example in CFC, LD

LE

Comparison Operator: Less than or equal to.

A Boolean operator that returns the value TRUE when the value of the first operand is less than or equal to the value of the second.

Allowed data types: VAR1 must be BOOL.

Other operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, and STRING.

Example in Structured Text: (* Result is TRUE *)

```
VAR1 := 20 <= 30;
```

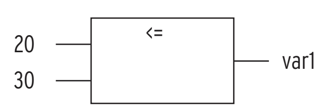


Figure B.21 LE Example in CFC, LD

LT

Comparison Operator: Less than.

A Boolean operator that returns the value TRUE when the value of the first operand is less than the value of the second.

Allowed data types: VAR1 must be BOOL.

Other operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, and STRING.

Example in Structured Text: (* Result is TRUE *)

VAR1 := 20 < 30;

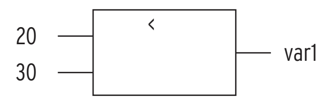


Figure B.22 LT Example in CFC, LD

NE

Comparison Operator: Not equal to.

A Boolean operator that returns the value TRUE when the operands are not equal.

Allowed data types: VAR1 must be BOOL.

Other operands can be BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, and STRING.

Example in Structured Text: (* Result is FALSE *)

VAR1 := 40 <> 40;

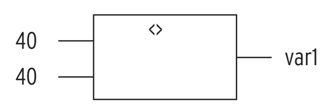


Figure B.23 NE Example in CFC, LD

Type Conversion Functions

You can convert natively from any elementary type to any other elementary type. The system cannot, however, convert from a larger type to a smaller type (for example, from INT to BYTE or from DINT to WORD) implicitly. Use type conversion functions for this.

Syntax:

<elem.Typ1>_TO_<elem.Typ2>

NOTE
When you perform a type conversion from a larger type to a smaller type, you risk losing some information.

Notice that in each _TO_STRING conversion, the string is generated left-justified. If the string is defined too small for the resulting conversion, the resulting string will be truncated from the right.

ACSELERATOR RTAC supports the following type conversions:

➤ *BOOL_TO Conversions on page 694*
➤ *TO_BOOL Conversions on page 695*

- *Conversion Between Integral Number Types on page 696*
- *REAL_TO/LREAL_TO Conversions on page 696*
- *TIME_TO/TIME_OF_DAY Conversions on page 697*
- *DATE_TO/DT_TO Conversions on page 697*
- *STRING_TO Conversions on page 698*
- *TRUNC on page 699*

The supported data types in IEC 61131 are shown in *Table B.4*. Although any type can be converted to any other type, some data may be lost or may no longer be meaningful.

Table B.4 IEC Data Type

BOOL	LREAL	UDINT
BYTE	LTIME	UINT
DATE	LWORD	ULINT
DINT	REAL	USINT
DT	SINT	WORD
DWORD	STRING	WSTRING
INT	TIME	
LINT	TOD	

BOOL_TO Conversions

Operator: Converting from data type BOOL to any other data type.

Syntax for a BOOL_TO conversion operator:

BOOL_TO_<data type>

For number types, the result is 1 when the operand is TRUE; the result is 0 when the operand is FALSE.

For the STRING type, the result is TRUE when the operand is TRUE; the result is FALSE when the operand is FALSE.

Examples in Structured Text:

```
i := BOOL_TO_INT(TRUE); (* Result is 1 *)

str := BOOL_TO_STRING(TRUE); (* Result is "TRUE" *)

t := BOOL_TO_TIME(TRUE); (* Result is T#1ms *)

tof := BOOL_TO_TOD(TRUE); (* Result is TOD#00:00:00.001 *)

dat := BOOL_TO_DATE(FALSE); (* Result is D#1970-01-01 *)

dandt := BOOL_TO_DT(TRUE); (* Result is DT#1970-01-01-00:00:01 *)
```

Examples in CFC, LD:

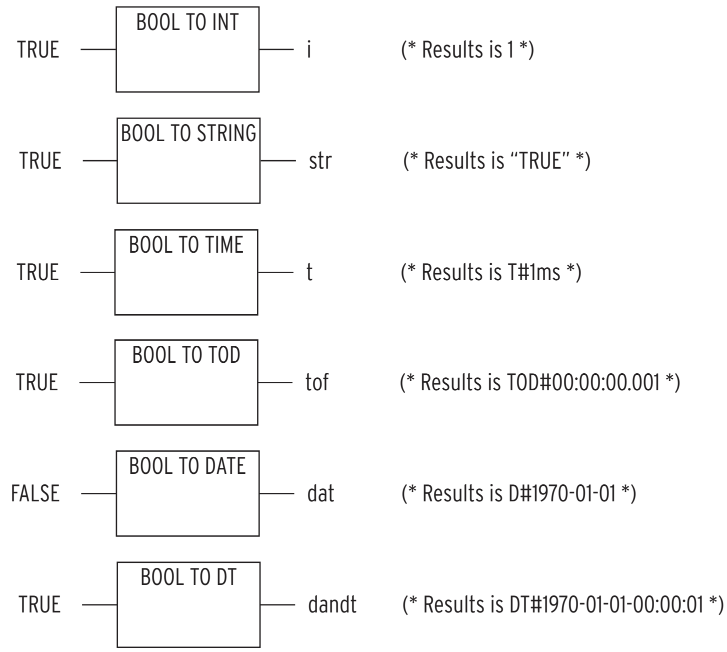


Figure B.24 BOOL_TO Conversions

TO_BOOL Conversions

Operator: Converting from another variable type to BOOL.

Syntax for a TO_BOOL conversion operator:

<data type>_TO_BOOL

The result is TRUE when the operand is not equal to 0. The result is FALSE when the operand equals 0.

The result is TRUE for STRING type variables when the operand is TRUE. Otherwise, the result is FALSE.

Examples in CFC, LD:

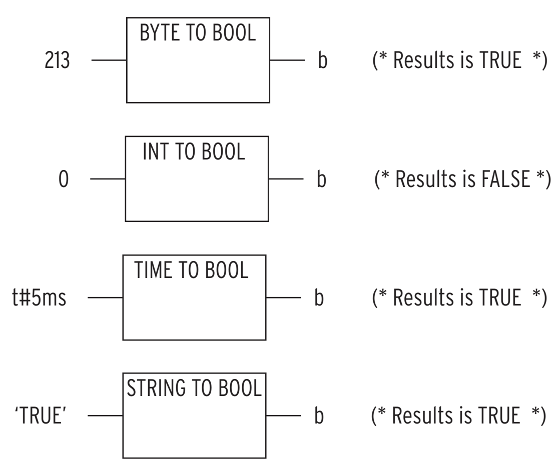


Figure B.25 TO_BOOL Conversions

Examples in Structured Text:

b := BYTE_TO_BOOL(2#11010101); (* Result is TRUE *)

b := INT_TO_BOOL(0); (* Result is FALSE *)

b := TIME_TO_BOOL(T#5ms); (* Result is TRUE *)

b := STRING_TO_BOOL('TRUE'); (* Result is TRUE *)

Conversion Between Integral Number Types

Conversion from an integral number type to another number type.

Syntax for the conversion operator:

<INT data type>_TO_<INT data type>

If the number you are converting exceeds the range limit, the first bytes for the number will be ignored.

Example in Structured Text:

si := INT_TO_SINT(4223); (* Result is 127 *)

If you save the integer 4223 (16#107f represented hexadecimally) as a SINT (short integer) variable, it will appear as 127 (16#7f represented hexadecimally) because the sign bit will be truncated.

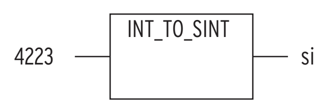


Figure B.26 INT_TO_SINT Example in CFC, LD

REAL_TO/LREAL_TO Conversions

Operator: Converting from the variable type REAL or LREAL to a different type.

The value will be rounded up or down to the nearest whole number and converted into the new variable type. Exceptions to this are the variable types STRING, BOOL, REAL, and LREAL.

Please note that when you convert to type STRING, the total number of digits is limited to 16. If the (L)REAL-number has more digits, then the software rounds the 16th digit. If you define the length of the string too short, it will be truncated from the right.

Example in Structured Text:

i := REAL_TO_INT(1.5); (* Result is 2 *)

j := REAL_TO_INT(1.4); (* Result is 1 *)

i := REAL_TO_INT(-1.5); (* Result is -2 *)

j := REAL_TO_INT(-1.4); (* Result is -1 *)

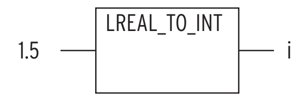


Figure B.27 LREAR_TO_INT Example in CFC, LD

TIME_TO/TIME_OF_DAY Conversions

Operator: Converting from the variable type TIME or TIME_OF_DAY to a different type.

Syntax for the conversion operator:

<time data type>_TO_<data type>

Time is stored internally in a DWORD in milliseconds (beginning with 12:00 a.m. for the TIME_OF_DAY variable). This millisecond value is the one converted.

For the STRING type variable, the result is a time constant.

Examples in Structured Text:

```
str := TIME_TO_STRING(T#12ms); (* Result is T#12ms *)

dw := TIME_TO_DWORD(T#5m); (* Result is 300000 *)

si := TOD_TO_SINT(TOD#00:00:00.012); (* Result is 12 *)
```

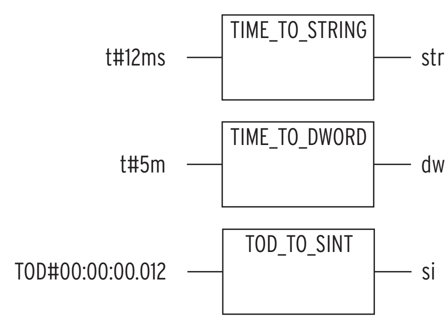


Figure B.28 TIME_TO Examples in CFC, LD

DATE_TO/DT_TO Conversions

Operator: Converting from the variable type DATE or DATE_AND_TIME to a different type.

Syntax for the conversion operator:

<date data type>_TO_<data type>

The date is stored internally in a DWORD in seconds since Jan. 1, 1970. This value in seconds is the converted value.

For STRING type variables, the result is the date constant.

Examples in Structured Text:

```
b := DATE_TO_BOOL(D#1970-01-01); (* Result is FALSE *)

i := DATE_TO_INT(D#1970-01-15); (* Result is 29952 *)
```

byt := DT_TO_BYTE(DT#1970-01-15-05:05:05); (* Result is 129 *)

str := DT_TO_STRING(DT#1998-02-13-14:20); (* Result is 'DT#1998-02-13-14:20' *)

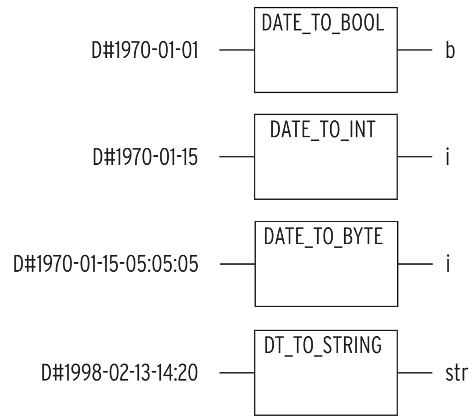


Figure B.29 DATE_TO Examples in CFC, LD

STRING_TO Conversions

Operator: Converting from the variable type STRING to a different type.

Syntax for the conversion operator:

STRING_TO_<data type>

The operand from the STRING type variable must contain a value that is valid in the target variable type, or the result will be 0.

Examples in Structured Text:

b := STRING_TO_BOOL('TRUE'); (* Result is TRUE *)

w := STRING_TO_WORD('abc34'); (* Result is 0 *)

t := STRING_TO_TIME('T#127ms'); (* Result is T#127ms *)

Examples in CFC:

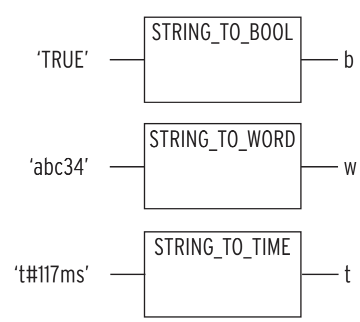


Figure B.30 STRING_TO Examples in CFC, LD

TRUNC

Operator: Converting from REAL to DINT. The whole number portion of the value will be used.

Examples in Structured Text:

i := TRUNC(1.9); (* Result is 1 *)

i := TRUNC(-1.4); (* Result is -1 *)

Custom Conversion Functions

Custom conversions provide a means to perform conversion operations on ACSELERATOR RTAC data types and data classes.

operAPC_TO_TIME

Operator: Converting from the structure operAPC to a TIME data type.

The operAPC.setMag is converted to milliseconds based on the time unit supplied to the function and then converted to a TIME data type. Fractions of milliseconds are truncated. Maximum value is T#49d17h2m47s295ms.

Inputs:

In : operAPC;

TimeUnit : timeUnit_t; (3 = DAY, 4 = HOUR, 5 = MINUTE, 6 = SECOND, 7 = MILLISECOND)

Output: Returns a TIME data type representing a combination of the number specified in the input IN (operAPC.setMag) and the unit of time specified in TimeUnit.

Examples in ST:

VarTime := operAPC_TO_TIME(Var_operAPC, 3); (Returns time in hours)

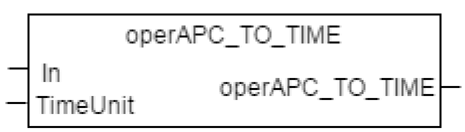


Figure B.31 operAPC_TO_TIME Example in CFC

SPS2_TO_DPS

Operator: Convert two SPS data types to a DPS data type.

Inputs:

SPS1 : SPS;

SPS2 : SPS;

Output: Returns a DPS data type representing a combination of two SPS data types, according to the following table. The quality is assigned based on the input SPS that has the least data quality (for example, if SPS1 has q.validity = invalid then the DPS returns value q.validity = invalid). The time attributes are assigned by the most recently updated input.

Table B.5 Status Value of Data Structures

SPS1	SPS2	DPS
False	False	dbpos_intermediate
False	True	dbpos_off
True	False	dbpos_on
True	True	dbpos_bad

Examples in ST:

VarDPS := SPS2_TO_DPS(SPS1, SPS2);

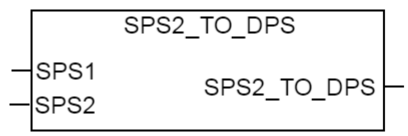


Figure B.32 SPS2_TO_DPS Example in CFC

SPS_TO_DPS

Operator: Convert an SPS data type to a DPS data type

Inputs:

SPS1 : SPS;

Output: Returns a DPS data type representing the state of an SPS data type, according to the following table. All other data attributes are identical to the SPS1.

Table B.6 Status Value of Data Structures

InSPS	DPS
False	dbpos_off
True	dbpos_on

Examples in ST:

VarDPS := SPS_TO_DPS(SPS1);

Figure B.33 SPS_TO_DPS Example in CFC

Numeric Functions

The following numeric IEC operators are available: ABS, SQRT, LN, LOG, ESP, SIN, COS, TAN, ASIN, ACOS, ATAN, EXPT.

ABS

Operator: Returns the absolute value of a number.

ABS(–2) returns 2.

ACSELERATOR RTAC permits the following data type combinations for input and output tags.

Table B.7 Permitted Data Type Combinations for Input and Output Tags

Input Tags	Output Tags
IN	OUT
INT	INT, REAL, WORD, DWORD, DINT
REAL	REAL
BYTE	INT, REAL, BYTE, WORD, DWORD, DINT
WORD	INT, REAL, WORD, DWORD, DINT
DWORD	REAL, DWORD, DINT
SINT	REAL
USINT	REAL
UINT	INT, REAL, WORD, DWORD, DINT, UDINT, UINT
DINT	REAL, DWORD, DINT
UDINT	REAL, DWORD, DINT, UDINT

Example in Structured Text:

```
i := ABS(-2);
```

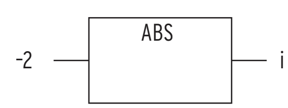


Figure B.34 ABS Example in CFC, LD

ACOS

Operator: Returns the arc cosine (inverse function of cosine) of a number. Calculates the value in radians.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 1.0472 *)

```
q := ACOS(0.5);
```

Figure B.35 ACOS Example in CFC, LD

ASIN

Operator: Returns the arc sine (inverse function of sine) of a number.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 0.523599 *)

```
q := ASIN(0.5);
```

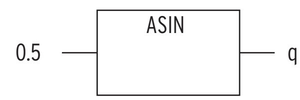


Figure B.36 ASIN Example in CFC, LD

ATAN

Operator: Returns the arc tangent (inverse function of tangent) of a number. Calculates the value in radians.

The operand can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 0.463648 *)

```
q := ATAN(0.5);
```

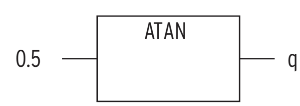


Figure B.37 ATAN Example in CFC, LD

COS

Operator: Returns the cosine of a number. Calculates the value in radians.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 0.877583 *)

```
q := COS(0.5);
```

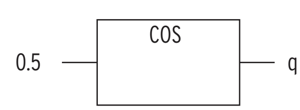


Figure B.38 COS Example in CFC, LD

DEG_TO_RAD

Non-IEC function: Returns the radian equivalent of the degrees given.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 0.08726647 *)

```
q := DEG_TO_RAD(5);
```

Figure B.39 DEG_TO_RAD Example in CFC, LD

EXP

Operator: Returns the exponential function.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 7.389056099 *)

```
q := EXP(2);
```

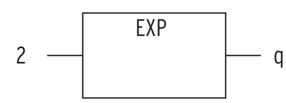


Figure B.40 EXP Example in CFC, LD

EXPT

Operator: Exponentiation of a variable with another variable:

OUT = IN1^IN2.

The two operands can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result is 49 *)

```
var1 := EXPT(7,2);
```

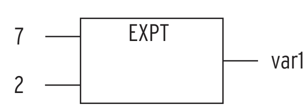


Figure B.41 EXPT Example in CFC, LD

LN

Operator: Returns the natural logarithm of a number.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 3.80666 *)

```
q := LN(45);
```

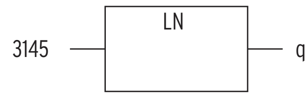


Figure B.42 LN Example in CFC, LD

LOG

Operator: Returns the logarithm of a number in base-10.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 2.49762 *)

```
q := LOG(314.5);
```

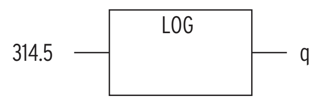


Figure B.43 LOG Example in CFC, LD

RAD_TO_DEG

Non-IEC function: Returns the degrees equivalent of the radians given.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 286.4789 *)

```
q := RAD_TO_DEG(5);
```

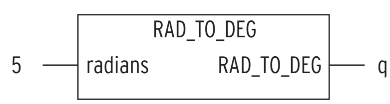


Figure B.44 RAD_TO_DEG Example in CFC, LD

SIN

Operator: Returns the sine of a number.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 0.479426 *)

```
q := SIN(0.5);
```

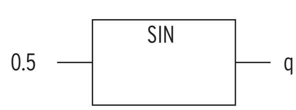


Figure B.45 SIN Example in CFC, LD

SQRT

Operator: Returns the square root of a number.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT.

The output variable must be type REAL.

Example in Structured Text: (* Result in q is 4 *)

```
q := SQRT(16);
```

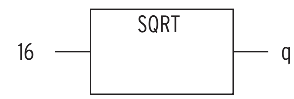


Figure B.46 SQRT Example in CFC, LD

TAN

Operator: Returns the tangent of a number. Calculates the value in radians.

The input variable can be type BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, and UDINT. The output variable must be type REAL.

Example in Structured Text: (* Result in q is 0.546302 *)

```
q := TAN(0.5);
```

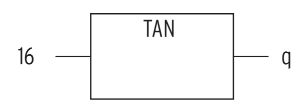


Figure B.47 TAN Example in CFC, LD

String Functions

The following string IEC functions are available: CONCAT, DELETE, FIND, INSERT, LEFT, LEN, MID, REPLACE, RIGHT.

CONCAT

This function is of type STRING and combines two strings into one.

Inputs:

STR1, STR2: STRING; (* strings to be concatenated *)

Return value:

STRING; (* concatenated string *)

CONCAT (STR1, STR2) connects STR1 to STR2 into a single string.

Example in ST: (* Result in VarSTRING1 is Hello world *)

```
VarSTRING1 := CONCAT('Hello', ' world');
```

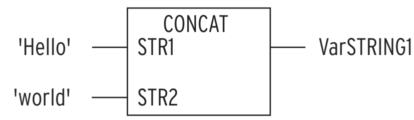


Figure B.48 CONCAT Example in CFC, LD

DELETE

This function is of type STRING and removes a partial string from a larger string at a defined position.

Inputs:

STR : (* STRING; string from which a part should be deleted *)

LEN : INT; (* length of the partial string to be deleted, number of characters *)

POS : INT; (* position in STR where the deletion of LEN characters should start, counted from left *)

Return value:

STRING, string remaining after deletion

DELETE(STR, L, P) means: Delete L characters from STR, beginning with the character in the P position.

Example in ST:

```
var1 := DELETE ('SUXYSI',2,3);
```

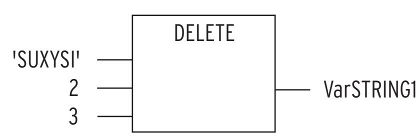


Figure B.49 DELETE Example in CFC, LD

FIND

This function is of type INT and searches for the position of a partial string within a string.

Inputs:

STR1 : STRING; (* string within which STR2 should be searched *)

STR2 : STRING; (* string whose position should be searched in STR1 *)

Return value:

INT (* start position of STR2 in STR1; "0" if STR2 is not found in STR1 *)

FIND(STR1, STR2) means: Find the position of the first character where STR2 appears in STR1 for the first time. If STR2 is not found in STR1, then OUT := 0.

Example in ST:

```
arINT1 := FIND ('abcdef','de');
```

Figure B.50 FIND Example in CFC, LD

INSERT

This function is of type STRING and inserts a string into another string at a defined point.

Inputs:

STR1 : STRING; (* string into which STR2 has to be inserted *)

STR2 : STRING; (* string that has to be inserted into STR1 *)

POS : INT; (* Position in STR1 where STR2 has to be inserted, number of characters counted from left *)

Return value:

STRING; (* resulting string *)

INSERT(STR1, STR2, POS) means: Insert STR2 into STR1 after position POS.

Example in ST:

VarSTRING1 := INSERT ('SUSI','XY',2);

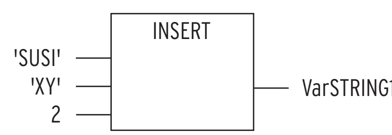


Figure B.51 INSERT Example in CFC, LD

LEFT

This function is of type STRING and returns the left, initial string for a given string.

Inputs:

STR : STRING; (* string to be analyzed *)

SIZE : INT; (* length of left initial string (number of characters) *)

Return value:

STRING; (* initial string *)

LEFT (STR, SIZE) means: Return the first SIZE characters from the left in the string STR.

Example in ST:

VarSTRING1 := LEFT ('SUSI',3);

Figure B.52 LEFT Example in CFC, LD

LEN

This function is of type STRING and returns the length of a string.

Input:

STR : STRING; (* string to be analyzed *)

Return value:

INT, length of string (number of characters)

Example in ST:

VarINT1 := LEN ('SUSI');

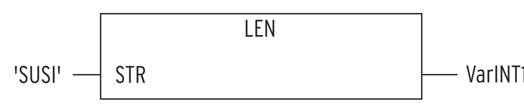


Figure B.53 LEN Example in CFC, LD

MID

This function is of type STRING and returns a partial string from within a string.

Inputs:

STR : STRING; (* string to be analyzed *)

LEN : INT; (* length of the partial string (number of characters) *)

POS : INT; (* start position for the partial string, number of characters counted from the left of STR *)

Return value:

STRING; (* partial string *)

MID(STR, LEN, POS) means: Retrieve LEN characters from the STR string beginning with the character at position POS.

Example in ST:

VarSTRING1 := MID ('SUSI',2,2);

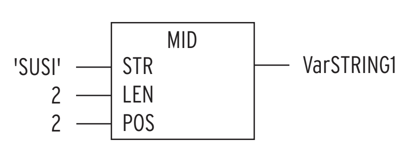


Figure B.54 MID Example in CFC, LD

REPLACE

This function is of type STRING and replaces a partial string from a larger string with a third string.

Inputs:

STR1 : STRING; (* string of which a part should be replaced by string STR2 *)

STR2 : STRING: (* string that should replace a part of STR1 *)

L : INT; (*length of partial string in STR1 that should be replaced *)

P : INT; (*position where STR2 should be inserted instead of the existing L characters *)

Return value:

STRING; (* resulting string *)

REPLACE(STR1, STR2, L, P) means: Replace L characters from STR1 by STR2, beginning with the character in the P position.

Example in ST:

VarSTRING1 := REPLACE ('SUXYSI','K',2,2);

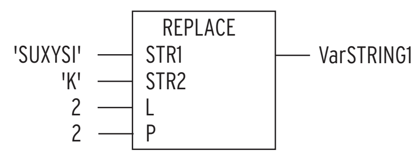


Figure B.55 REPLACE Example in CFC, LD

RIGHT

This function is of type STRING and returns the right, initial string for a given string.

Inputs:

STR : STRING; (* string to be analyzed *)

SIZE : INT; (* number of characters to be counted from the right in string STR *)

Return value:

STRING; (* initial right string *)

RIGHT (STR, SIZE) means: Return the first SIZE character from the right in the string STR.

Example in ST:

VarSTRING1 := RIGHT ('SUSI',3);

Figure B.56 RIGHT Example in CFC, LD

Time Functions

DateCalc

Operator: This function adds a time interval specified by Unit and Number to an existing DT variable.

Inputs:

Unit : timeUnit_t; (0 = YEAR, 1 = MONTH, 2 = WEEK, 3 = DAY, 4 = HOUR, 5 = SECOND, 6 = SECOND)

Number : DINT;

StartDate : DT;

Outputs: Returns a DT data type representing the addition of the "Number" of "Units" added to the "StartDate".

Examples in ST:

VarTimeOut := DateCalc(VarUnit, VarNumber, VarStartDate);

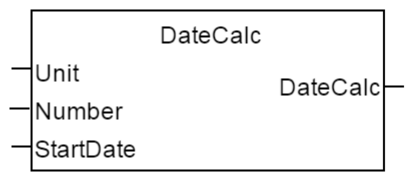


Figure B.57 DateCalc Example in CFC

dateTime_t_TO_DNP48

Operator: Converts a dateTime_t data structure into three separate variables. Each variable contains 16 bits of the original dateTime_t data structure.

Inputs:

Value : dateTime_t;

Outputs:

HighRegister : DINT;T

MidRegister : DINT;

LowRegister : DINT;

Examples in ST:

dateTime_t_TO_DNP48 (VarValue, HighRegister => VarHigh, MidRegister => VarMed, LowRegister => VarLow);

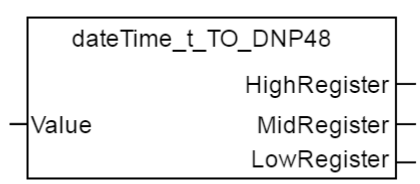


Figure B.58 dateTime_t_TO_DNP48 Example in CFC

DayOfWeek

Operator: When given a DT variable, this function returns the day of week.

Inputs:

InDate : DT; Outputs:

DayString : STRING;

Returns a dayOfWeek_t that can be used interchangeably with a DINT data type (0 = SUNDAY, 1 = MONDAY, 2 = TUESDAY, 3 = WEDNESDAY, 4 = THURSDAY, 5 = FRIDAY, 6 = SATURDAY)

Examples in ST:

Var_dayOfWeek_t := DayOfWeek (VarDT, DayString => VarString);

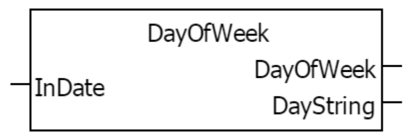


Figure B.59 DayOfWeek Example in CFC

DayOfYear

Operator: When given a DT variable, this function returns the day of year.

Inputs:

InDate : DT;

Outputs: Returns a DINT that represents day of year.

Examples in ST:

VarDINT := DayOfWeek (VarDT);

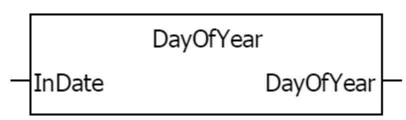


Figure B.60 DayOfYear Example in CFC

DNP48_TO_dateTime_t

Operator: Converts three 16-bit values representing milliseconds since epoch to a DT data type. The three inputs when combined represent milliseconds since Jan 1, 1970.

Inputs:

HighRegister : DINT;

MidRegister : DINT;

LowRegister : DINT;

Outputs: Returns a DT data type representing the combined input register representing a 48-bit time.

Examples in ST:

Var_dateTime_t := DNP48_TO_dateTime_t (VarHigh, VarMed, VarLow);

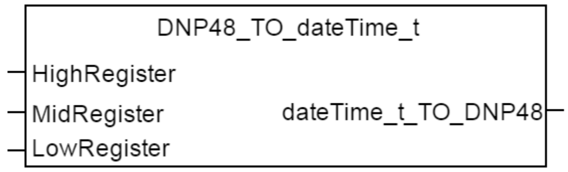


Figure B.61 DNP48_TO_dateTIME_t Example in CFC

dateTime_t_Constructor

Operator: Constructs a dateTime_t data structure from the provided inputs.

Inputs:

Year : DINT;

Month : DINT;

Day : DINT;

Hour : DINT;

Minute : DINT;

Second : DINT;

Millisecond : DINT;

Outputs: Returns a dateTime_t data type representation of the provided time components

Examples in ST:

Var_dateTime_t := dateTime_t_Constructor (VarYear, VarMonth, VarDay, VarHour, VarMinute, VarSecond, VarMillisecond);

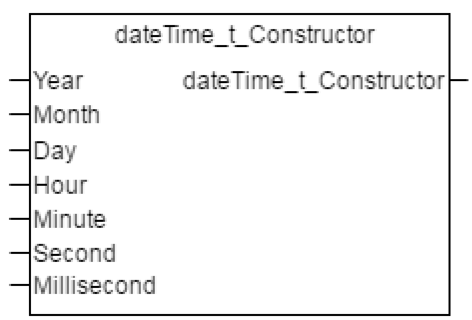


Figure B.62 dateTime_t_Constructor Example in CFC

dateTime_t_Deconstructor

Operator: Deconstructs a dateTime_t data structure into its constituent parts.

Inputs:

Value : dateTime_t;

Outputs:

Year : DINT;

Month : DINT;

Day : DINT;

Hour : DINT;

Minute : DINT;

Sec : DINT;

Ms : DINT;

Examples in ST:

dateTime_t_Deconstructor (Value := Var_dateTime_t, Year => VarYear, Month => VarMonth, Day => VarDay, Hour => VarHour, Minute => VarMinute, Sec => VarSec, Ms => VarMs);

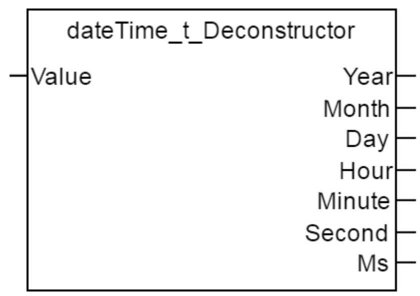


Figure B.63 dateTime_t_Deconstructor Example in CFC

Custom Functions

The following non-IEC standard functions are available.

CMVRangeAndDeadbandCheck

CMVRangeAndDeadbandCheck is used by the Tag Processor to check the absolute value of the CMV tags instCVal.mag to determine if it exceeds the dead band, and if so, assign it to cVal.mag. The cVal.mag is then checked against the zero dead band and the various low and high limits. If a limit is exceeded, the .range is appropriately set. Finally, the .q.validity is set to questionable or good, depending on the instCVal range check and the .q.validity.

Structured Text Example:

CMVRangeAndDeadbandCheck(tCMV := VarCMV);

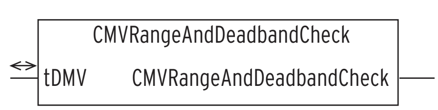


Figure B.64 CMVRangeAndDeadbandCheck Example in CFC, LD

INSRangeCheck

The INSRangeCheck function performs range checking. The rangeC values and the stVal are used to calculate and set the range attribute.

The INS RangeCheck function takes an INS variable as an input/output parameter. The function returns a BOOL but calling methods will ignore it.

Structured Text Example:

```
INSRangeCheck(tINS := VarINS);
```

Figure B.65 INSRangeCheck Example in CFC, LD

MERGE_q

This function performs a logical OR of all the quality bits of the two input qualities.

Inputs:

Qual_1, Qual_2 : quality_t

Output:

Returns modified quality_t structure, where q is the result of an OR operation on each quality bit between the Qual_1 and Qual_2 to reflect the worst case.

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
SCADA1_DNP.AI_0000.q		MERGE_q (Feeder1_DNP.AI_0000.q, Feeder1_DNP.AI_0001.q)	QUALITY_T

Figure B.66 MERGE_q Example in Tag Processor

Example in ST:

```
SCADA1_DNP.AI_0000.q := MERGE_q (Feeder1_DNP.AI_0000.q,
Feeder1_DNP.AI_0001.q)
```

MVRangeAndDeadbandCheck

MVRangeAndDeadbandCheck is used by the Tag Processor to check the absolute value of the MV tags instMag.mag to determine if it exceeds the dead band, and if so, assign it to mag. The mag is then checked against the zero dead band and the various low and high limits. If a limit is exceeded, the .range is appropriately set. Finally, the .q.validity is set to questionable or good, depending on the instaMag range check and the .q.validity.

Structured Text Example:

```
MVRangeAndDeadbandCheck(tMV := VarMV);
```

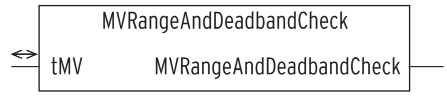


Figure B.67 MVRangeAndDeadbandCheck Example in CFC, LD

SIZEOF

This function returns the number of bytes required by the given variable.

Example in IL: (* Result is 10 *)

arr1:ARRAY[0..4] OF INT;

var1:INT;

Example in ST:

var1 := SIZEOF(arr1);

SYS_TIME

This function returns the full time-stamp structure of the current system time.

The data type is timeStamp_t.

Example in Structured Text:

systime := SYS_TIME();

TIME

This function returns the time (based on milliseconds) that has passed since the last system power cycle.

The data type is TIME.

Example in Structured Text:

systime := TIME(); (* Result e.g., : T#35m11s342ms *)

TRANSLATE

Operator: Translates the input value to a new value by scaling it by the ratio between Range In and Out.

Inputs:

IN : REAL;

Range_Min_In : REAL;

Range_Max_In : REAL;

Range_Min_Out : REAL;

Range_Max_Out : REAL;

Outputs: Returns a REAL value that is the IN input scaled based on the ranges provided.

Examples in ST:

VarReal := TRANSLATE (VarInReal, VarMinIn, VarMaxIn, VarMinOut, VarMaxOut);

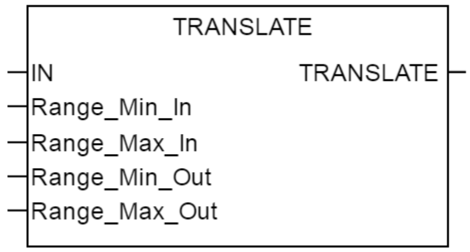


Figure B.68 TRANSLATE Example in CFC

TRANSLATE_MV

Operator: Translates the input value to a new value by scaling it by the ratio between Range In and Out. The quality and time attributes are transferred from the IN data structured to the returned data structure.

Inputs:

IN : MV;

Range_Min_In : REAL;

Range_Max_In : REAL;

Range_Min_Out : REAL;

Range_Max_Out : REAL;

Outputs: Returns a MV data structure in which the instMag of the IN data structure is scaled based on the ranges provided.

Examples in ST:

VarMV := TRANSLATE_MV (VarInMV, VarMinIn, VarMaxIn, VarMinOut, VarMaxOut);

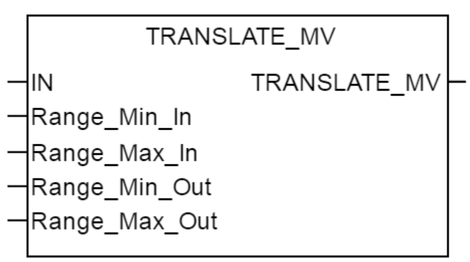


Figure B.69 TRANSLATE_MV Example in CFC

TRANSLATE_INS

Operator: Translates the input value to a new value by scaling it by the ratio between Range In and Out. The quality and time attributes are transferred from the IN data structured to the returned data structure.

Inputs:

IN : INS;

Range_Min_In : DINT;

Range_Max_In : DINT;

Range_Min_Out : DINT;

Range_Max_Out : DINT;

Outputs: Returns an INS data structure in which the stVal of the IN data structure is scaled based on the ranges provided.

Examples in ST:

VarINS := TRANSLATE_INS (VarInINS, VarMinIn, VarMaxIn, VarMinOut, VarMaxOut);

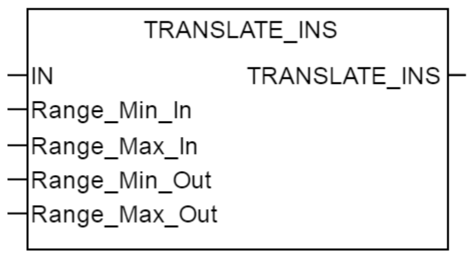


Figure B.70 TRANSLATE_INS Example in CFC

TS_DIFF

This function subtracts two time stamps.

Inputs:

T1_val, T2_val : dateTime_t

Output:

Returns the REAL number result, in seconds, from subtracting T2_val from T1_val

Example in Tag Processor:

Destination Tag Name	DT Data Type	Source Expression	SE Data Type
my_REAL_var	REAL	TS_DIFF (Feeder1_DNP.BI_0000.t.value, Feeder1_DNP.BI_0001.t.value)	REAL

Figure B.71 TS_DIFF in CFC

Example in ST:

my_REAL_var := TS_DIFF (Feeder1_DNP.BI_0000.t.value, Feeder1_DNP.BI_0001.t.value)

ValidityCheck

ValidityCheck is used to examine the detail quality and set validity to good, questionable, or invalid depending on the detail quality attributes.

The ValidityCheck function takes a quality_t variable as an input/output parameter. The function returns a BOOL but calling methods shall ignore it.

Example in ST:

ValidityCheck(tq := VarQuality.q);

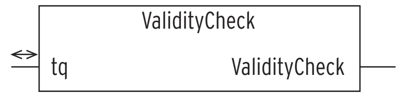


Figure B.72 ValidityCheck Example in CFC, LD

Ethernet Interface Control

The following functions provide online control over enabling and disabling Ethernet interfaces on the RTAC and setting IP addresses and associated gateway addresses. The current state of the given interface can still be accessed while in online mode with the ACSELERATOR RTAC from the **SystemTags > Communications** tab.

These functions are intended for use in advanced Ethernet interface control applications. For general applications, see *Ethernet_Interface_Control on page 725*.

Functions:

Enable_ethernet_link Operator: Enables an Ethernet interface.

Inputs:

interface_id : Enum_interface_id

Input/Output:

response: Enum_config_response

Example in ST:

Enable_ethernet_link (Eth_01, Enum_config_response_1);

Disable_ethernet_link Operator: Disables an Ethernet interface.

Inputs:

interface_id : Enum_interface_id

Input/Output:

response : Enum_config_response

Example in ST:

Disable_ethernet_link (Eth_01, Enum_config_response_1);

Config_IPv4_addr Operator: Configures a static IPv4 address and subnet mask for an Ethernet interface by using CDIR notation; for example, 192.168.0.2/32.

Inputs:

Interface_id : Enum_interface_id

Ipv4_addr : STRING(255)

Input/Output:

Response : Enum_config_response

Example in ST:

Config_IPv4_addr (Eth_01, '192.168.1.4/32', Enum_config_response_1);

Config_default_gateway Operator: Adds a default gateway to the specified interface. Optionally allows gateway to be flagged as "primary default gateway".

Input:

interface_id : Enum_interface_id

default_gw : STRING(255)

is_primary : BOOL

Input/Output:

Response : Enum_config_response

Example in ST:

Config_default_gateway (Eth_01, '192.168.1.1/32', TRUE, Enum_config_response_1);

Get_ethernet_state Operator: Returns the state of configuration change from the logic engine or web interface.

Input:

interface_id : Enum_interface_id

Return:

is_enabled : BOOL

Example in ST:

VarBOOL1 := Get_ethernet_state(Eth_01);

Get_IPv4_addr Operator: Returns the current IPv4 address of the specified Ethernet interface in CIDR notation; for example, 192.168.0.2/32.

Input:

interface_id : Enum_interface_id

Input/Output:

ipv4_addr : STRING(255)

Example in ST:

Get_IPv4_addr(Eth_01, VarSTRING1);

Get_default_gateway Operator: Returns the default gateway of the specified Ethernet interface in CIDR notation; for example, 192.168.0.2/32.

Input:

interface_id : Enum_interface_id

Input/Output:

default_gw : STRING(255)

Example in ST:

Get_default_gateway (Eth_01, VarSTRING2);

ADD_IPv4_addr Operator: This function is used to add an IP address alias to the specified interface.

Input:

interface_id : Enum_interface_id

ipv4_addr : STRING(255)

Input/Output:

Response : Enum_config_response

Example in ST:

ADD_IPv4_addr(Eth_01, '192.168.3.5/24', var_enum_config_response);

DEL_IPv4_addr Operator: This function is used to delete an IP address alias from the specified interface.

Input:

Interface_id : Enum_interface_id

ipv4_addr : STRING(255)

Input/Output:

Response : Enum_config_response

Example in ST:

DEL_IPv4_addr(Eth_01, '192.168.3.5/24', var_enum_config_response);

GET_IPv4_extra_addrs Operator: This function is used to add the additional IP aliases to the specified interface by using the Add_IPv4_addr() function. This function needs to be given a starting address to an array of strings and then given a number of IP addresses to be returned. Each string in the array will contain one IP address. If the last index in the returned values contains an IP address, there are additional IP addresses on that interface. Once a null string is encountered in the array, there are no more additional IP addresses on that interface.

Input:

Interface_id : Enum_interface_id

ipv4_addr : POINTER TO STRING(255)

num_ipv4_addr : UDINT

Example in ST:

get_ipv4_extra_addrs(eth_01, ADR(var_array_of_Strings), 10);

Enumerations:

Enum_interface_id Used to identify a specific Ethernet interface on the device.

Name	Value
Eth_01	0
Eth_02	1
Eth_03	2
Eth_04	3
Eth_05	4
Eth_06	5
Eth_07	6
Eth_08	7
Eth_09	8
Eth_10	9
Eth_F	10

Enum_config_response This enumeration is a union of all return codes that could be provided to the user when the configuration functions are called.

Table B.8 Table B.8 Configuration Return Codes

Name	Value	Description
ERR_INVALID_IP_ADDRESS	0	Indicates that the IP address does not conform to the format *W.X.X.X/Y*. (Where *W* = 0–223; *X* = 0–255; and *Y* = 1–32).
ERR_INTERFACE_NOT_PRESENT	1	Indicates that the interface identifier does not map onto a physical interface on the device.
ERR_COMMAND_REJECTED	2	Indicates that the current function call was unable to successfully complete. This may occur if a similar function call for the same Ethernet interface is in progress. Also indicates any other internal processing errors.
ERR_DHCP_INTERFACE	3	Indicates that the operation cannot be performed as the interface is configured to use DHCP.
OK_IN_PROGRESS	4	Indicates that processing related to the function call has begun.
OK_COMPLETED	5	Indicates that the command was issued successfully.

ACKNOWLEDGE_ALL_EVENTS Operator: This function is used to acknowledge all items in the RTAC SOE log from the IEC 61131 logic engine. There are no inputs.

Example in ST:

```
ACKNOWLEDGE_ALL_EVENTS();
```

Predefined Function Blocks

The RTAC has several predefined function blocks that you can use to accelerate code completion of a custom logic project. To use a predefined function block, you must declare an instance of that function block with a user-defined name, as shown in *Figure B.73*. See *Section 9: Custom Logic* for more information on declaring an instance of a function block.

These predefined function blocks are timers, counters, and other routines where intermediate values must be retained between function block calls.

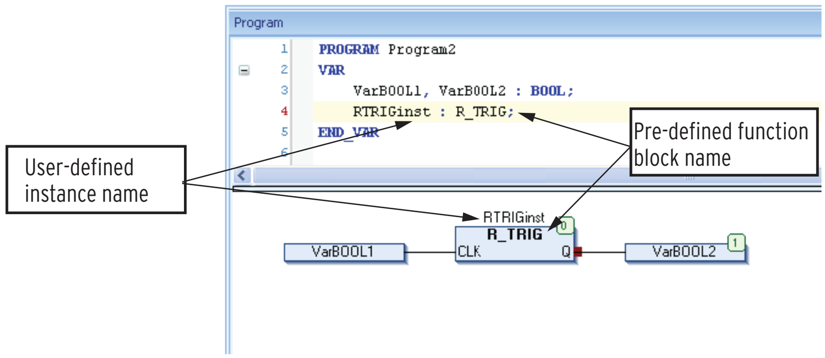


Figure B.73 Creating a Function Block Instance

CTD

This function block is a decrementing counter.

Inputs:

CD : BOOL; (* A rising edge at this input starts the incrementing of CV *)

LOAD : BOOL; (* If TRUE, CV will be reset to the upper limit given by PV *)

PV : WORD; (* Upper limit, i.e., start value for decrementing of CV *)

Outputs:

Q : BOOL; (* Gets TRUE as soon as CV is 0 *)

CV : WORD; (* Value to be decremented by 1, starting with PV until 0 is reached *)

When LOAD becomes TRUE, the counter variable CV is initialized with the upper limit PV. Each time CD has a rising edge change (from FALSE to TRUE), CV will be lowered by 1, provided CV is greater than 0. CV will not count below zero.

Q returns TRUE when CV is equal 0.

Declaration example:

CTDInst : CTD;

Example in ST:

CTDInst(CD := VarBOOL1, LOAD := VarBOOL2, PV := VarINT1);

VarBOOL3 := CTDInst.Q;

VarINT2 := CTDInst.CV;

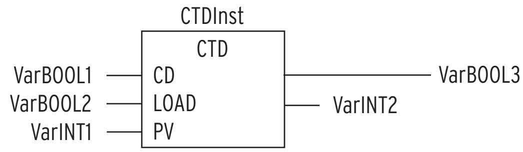


Figure B.74 CTD Example in CFC, LD

CTU

This function block is an incrementing counter:

Inputs:

CU : BOOL; (* A rising edge at this input starts the incrementing of CV *)

RESET : BOOL; (* If TRUE, CV will be reset to 0 *)

PV : WORD; (* Upper limit for the incrementing of CV *)

Outputs:

Q : BOOL; (* Gets TRUE as soon as CV has reached the limit given by PV *)

CV : WORD; (* Value to be counted up until it reaches PV *)

The counter variable CV will be initialized to 0 if RESET is TRUE. If CU has a rising edge from FALSE to TRUE, CV will increase by 1. Q will return TRUE when CV is greater than or equal to the upper limit PV.

Declaration example:

CTUInst : CTU;

Example in ST:

CTUInst(CU := VarBOOL1, RESET := VarBOOL2, PV := VarINT1);

VarBOOL3 := CTUInst.Q;

VarINT2 := CTUInst.CV;

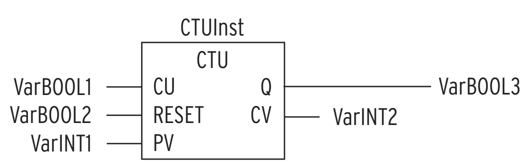


Figure B.75 CTU Example in CFC, LD

CTUD

This function block is an incrementing and decrementing counter.

Inputs:

CU : BOOL; (* If a rising edge occurs at CU, incrementing of CV will be started *)

CD : BOOL; (* If a rising edge occurs at CD, decrementing of CV will be started *)

RESET : BOOL; (* If TRUE, CV will be set to 0 *)

LOAD : BOOL; (* If TRUE, CV will be set to PV *)

PV : WORD; (* Upper limit for incrementing or decrementing CV *)

Outputs:

QU : BOOL; (* Returns TRUE when CV has been incremented to >= PV *)

QD : BOOL; (* Returns TRUE when CV has been decremented to 0 *)

CV : WORD; (* Value to be incremented or decremented *)

If RESET is TRUE, the counter variable CV will initialize to 0. If LOAD is TRUE, CV will initialize to the value of PV.

If CU has a rising-edge change from FALSE to TRUE, CV will increase by 1. If CD has a rising-edge change from FALSE to TRUE, CV will decrease by 1 provided this does not cause the value to fall below 0.

QU returns TRUE when CV has become greater than or equal to PV.

QD returns TRUE when CV has become equal to 0.

Declaration example:

CTUDInst : CUTD;

Example in ST:

CTUDInst(CU := VarBOOL1, CD := VarBOOL2, RESET := VarBOOL3, LOAD := VarBOOL4, PV := VarINT1);

VarBOOL5 := CTUDInst.QU;

VarBOOL6 := CTUDInst.QD;

VarINT2 := CTUDInst.CV;

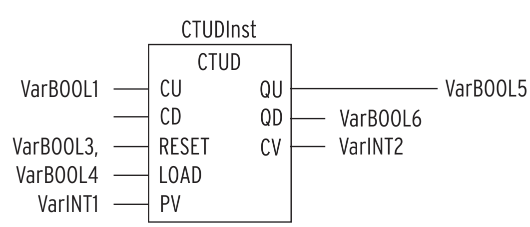


Figure B.76 CTUD Example in CFC, LD

Ethernet_Interface_Control

This function block provides online control over the enabling and disabling and a simple logical interface for control and status monitoring of the physical Ethernet interfaces of the RTAC as well as assignment of IP and default gateway addresses. For applications that require greater control and monitoring flexibility, you can leverage the Ethernet interface control functions described in this section.

Declaration Inputs:

Interface_Name : Enum_interface_id; (*The name of the interface on which to operate. See Enum_interface_id for a description of the Enum_interface_id enumeration*)

Inputs:

Set_Enable_State : BOOL; (*On the rising edge of this input, the New_Enable_State will be applied to the interface specified by Interface_Name*)

New_Enable_State : BOOL; (*The new state of the interface will be set when the operation is triggered by the Set_Enable_Sate input*)

Set_IP_Address : BOOL; (*On the rising edge of this input, the New_IP_Address will be applied to the interface specified by Interface_Name*)

New_IP_Address : STRING; (*This input should be an IP address and subnet mask specified in CIDR notation; for example, 192.168.0.2/32*)

Set_Gateway_IP : BOOL; (*On the rising edge of this input, the New_Gateway_IP will be set on the interface specified by Interface_Name*)

New_Gateway_IP : STRING; (*This input should be an IPv4 address specified in CIDR notation; for example, 192.168.0.1/32*)

Primary_Gateway : BOOL; (*The new gateway IP will be designated as the gateway used for all outgoing connections to remote networks*)

Outputs:

Interface_Enabled : BOOL; (*Indicates whether the interface selected by the Interface_Name input is currently enabled.*)

Set_Enable_State_Pending : BOOL; (*True while the enable state of the interface is being changed, false otherwise.*)

Set_Enable_State_DN : BOOL; (*Pulses for one processing interval when the enable state of the interface completes.*)

IP_Address : STRING; (*Indicates the current IP address configured on the interface specified by the Interface_Name input. This information is only available when the interface is enabled.*)

Set_IP_Address_Pending : BOOL; (*True while the IP address of the interface is being changed, false otherwise.*)

Set_IP_Address_DN : BOOL; (*Pulses for one processing interval when the IP address of the interface completes.*)

Gateway_IP : STRING; (*Indicates the current gateway IP configured on the interface specified by the Interface_Name input. This information is only available when the interface is enabled.*)

Set_Gateway_IP_Pending : BOOL; (*True while the IP address of the interface is being changed, false otherwise.*)

Set_Gateway_IP_DN : BOOL; (*Pulses for one processing interval when an attempt to change the IP address of the interface completes.*)

Interface_Not_Present : BOOL; (*Pulses for one processing interval when the requested operation fails because the interface is not present.*)

Interface_Busy : BOOL; (*Pulses for one processing interval when the requested operation fails because the interface cannot presently be modified.*)

Invalid_IP_Address : BOOL; (*Pulses for one processing interval when the requested operation fails because the provided IP address is not valid.*)

Invalid_Gateway_IP : BOOL; (*Pulses for one processing interval when the requested operation fails because the provided gateway IP is not valid.*)

Enumerations: See *Ethernet_Interface_Control on page 725* for a description of the Enum_interface_id enumeration.

Declaration example:

E1Cont : Ethernet_Interface_Control; (Interface_Name := Eth_01)

Example in ST:

E1Cont(Set_Enable_State := VarBOOL1, New_Enable_State := VarBOOL2,

Set_IP_Address := VarBOOL3, New_IP_Address := '192.168.1.4', Set_Gateway_IP := VarBOOL4,

New_Gateway_IP := '192.168.1.1', Primary_Gateway := VarBOOL5,);

VarBOOL6 := E1Cont.Interface_Enabled;

VarBOOL7 := E1Cont.Interface_Busy;

F_TRIG

This function block detects a falling edge.

Inputs:

CLK : BOOL; (* Incoming Boolean signal to be checked for falling edge *)

Outputs:

Q : BOOL; (* Becomes TRUE if a falling edge occurs at CLK *)

The output Q will remain FALSE as long as the input variable CLK returns TRUE. When CLK changes to FALSE, Q will return TRUE. Each time F_TRIG is called, it must detect a falling edge of CLK followed by a rising edge of CLK to trigger Q to TRUE.

Declaration example:

FTRIGInst : F_TRIG;

Example in ST:

FTRIGInst(CLK := VarBOOL1);

VarBOOL2 := FTRIGInst.Q;

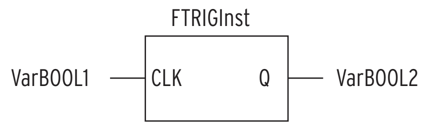


Figure B.77 F_TRIG Example in CFC, LD

PulseDecoder

This function block decodes pulseConfig attributes associated with an operSPC value. The output, Q, pulses as described by the pulseConfig attribute.

Example in Structured Text:

```
Program Test
VAR
PulseDecoderInst : PulseDecoder;
IN : operSPC;
OUT : BOOL;
ElapsedPulseCnt : USINT;
END_VAR
PulseDecoderInst (CTRL := IN);
OUT := PulseDecoderInst.Q;
ElapsedPulseCnt := PulseDecoderInst.EC_PulseCount;
```

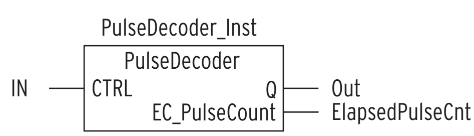


Figure B.78 PulseDecoder Example in CFC, LD

RS

This is a resetting bistable function block.

Inputs:

SET : BOOL;

RESET1 : BOOL;

Outputs:

Q1 : BOOL;

Q1 is set to TRUE only if RESET1 is FALSE and SET becomes TRUE. Q1 becomes FALSE when RESET1 is TRUE, regardless of the value of SET.

Q1 = RS (SET, RESET1) means:

Q1 = NOT RESET1 AND (Q1 OR SET)

Declaration example:

RSInst : RS;

Example in ST:

RSInst(SET := VarBOOL1, RESET1 := VarBOOL2);

VarBOOL3 := RSInst.Q1;

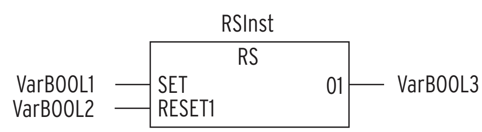


Figure B.79 RS Example in CFC, LD

RTC

The Run-Time Clock function block returns the current date and time at a given date and time.

Inputs:

EN : BOOL; (* At a rising edge starts counting up the time in PDT *)

PDT : DATE_AND_TIME; (* Date and time from which the counting up should be started *)

Outputs:

Q : BOOL; (* Is TRUE as long as CDT is counting up *)

CDT : DATE_AND_TIME; (* Current state of counted date and time *)

VarBOOL2 := RTC(EN, PDT, Q, CDT) means:

When EN is FALSE, the output variables Q and CDT are FALSE and DT#1970-01- 01-00:00:00, respectively.

When EN becomes TRUE (rising edge), CDT is set to the value of PDT and counted up in seconds as long as EN is TRUE. As soon as EN is reset to FALSE, CDT is reset to the initial value DT#1970-01-01-00:00:00.

Example in ST:

RTC(EN := VarBOOL1, PDT := DT#2006-03-30-14:00:00, Q => VarBOOL2, CDT => VarTimeCur);

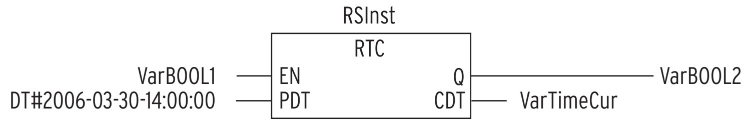


Figure B.80 RTC Example in CFC, LD

R_TRIG

This function block detects a rising edge.

Inputs:

CLK : BOOL; (* Incoming Boolean signal to be checked for rising edge *)

Outputs:

Q : BOOL; (* Becomes TRUE if a rising edge occurs at CLK *)

The output Q will remain FALSE as long as the input variable CLK is FALSE. When CLK changes to TRUE, Q will become TRUE for one processing interval, then return FALSE, even if CLK stays TRUE. Q is only set to TRUE when R_TRIG detects a rising edge on CLK.

Declaration example:

RTRIGInst : R_TRIG;

Example in ST:

RTRIGInst(CLK := VarBOOL1);

VarBOOL2 := RTRIGInst.Q;

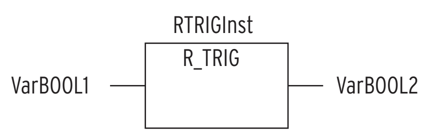


Figure B.81 R_TRIG Example in CFC, LD

SR

This is a dominant bistable function block.

Inputs:

SET1 : BOOL;

RESET : BOOL;

Output:

Q1 : BOOL;

Q1 is set to TRUE only if RESET is FALSE and SET1 becomes TRUE. Q1 remains TRUE until SET1 is FALSE and RESET is TRUE.

Q1 = SR (SET1, RESET) means:

Q1 = (NOT RESET AND Q1) OR SET1

Declaration example:

SRInst : SR;

Example in ST:

SRInst(SET1 := VarBOOL1, RESET := VarBOOL2);

VarBOOL3 := SRInst.Q1;

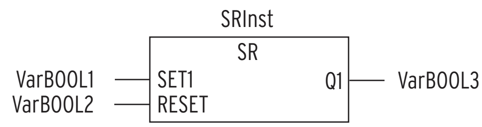


Figure B.82 SR Example in CFC, LD

TI

This function block is an interval timer. Effectively it is a self-initiating TON function block with a Boolean input override.

Timer ET will count up in milliseconds until it reaches the value of PT. At that time, output Q is forced TRUE. The timer is then reset and started again. Once Q is forced TRUE, it is reset to FALSE automatically at the next processing interval.

If IN goes TRUE at any time, Q will become TRUE regardless of the value of ET.

Inputs:

IN : BOOL: (* Rising edge forces Q to TRUE. *)

PT : TIME: (* Upper limit of ET (time interval) *)

Outputs:

Q : BOOL: (* Forced to TRUE as soon as ET has reached the value of PT or on the rising edge of IN *)

ET : TIME: (* Present value of the interval timer *)

Figure B.83 shows Q forced high because the value of timer ET reaches the limit PT. *Figure B.84* shows the rising edge on IN forces Q high. Notice the expiration of the ET timer is overridden by the IN rising-edge event. *Figure B.83* assumes the period PT is greater than the RTAC processing interval.

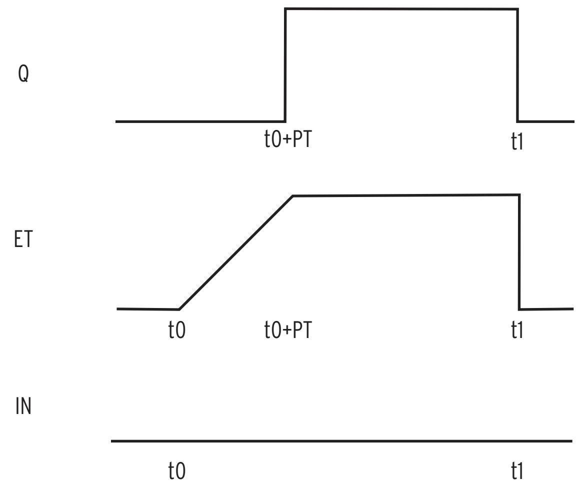


Figure B.83 Display of TI Behavior Over Time

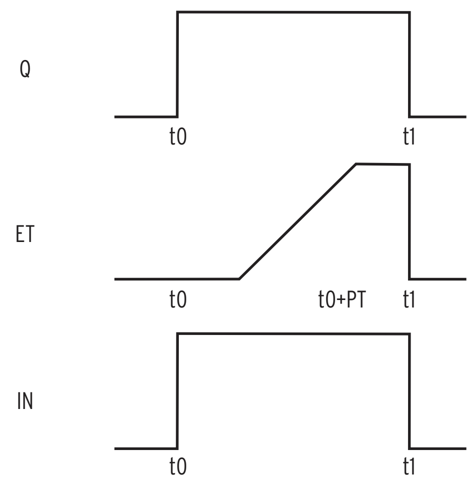


Figure B.84 Display of TI Behavior Triggered by IN

Declaration example:

TIinst : TI;

Example in ST:

TIinst(IN := VarBOOL1, PT := T#5s);

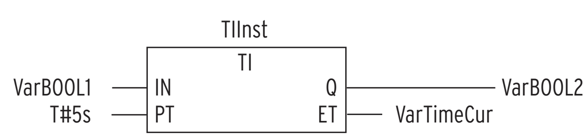


Figure B.85 TI Example in CFC, LD

TIV

This function block is a TI timer that also validates the input values. Functionally, TIV is the same as TI except for the two additional inputs and one additional output. The two inputs provide a range for input PT. If PT is greater than PT_MAX or less than PT_MIN, BOOL output invalid_input is forced TRUE.

Inputs:

IN : BOOL: (* Rising edge forces Q to TRUE *)

PT : TIME: (* Upper limit of ET (time interval) *)

PT_MAX : TIME; (* Maximum allowed value of PT *)

PT_MIN : TIME; (* Minimum allowed value of PT *)

Outputs:

Q : BOOL: (* Forced to TRUE as soon as ET has reached the value of PT *)

Invalid_input : BOOL: (* Forced to TRUE if PT < PT_MIN or PT > PT_MAX *)

Declaration example:

TIinst : TI;

TIVinst(IN := VarBOOL1, PT := T#5s, PT_MAX := 10, PT_MIN := 2);

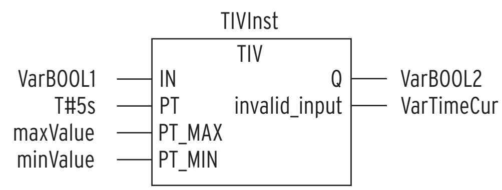


Figure B.86 TIV Example in CFC, LD

TOF

This function block is a turn-off delay.

When the input changes from TRUE to FALSE (falling edge), counter ET will begin counting in milliseconds to PT. When ET = PT, output Q is set FALSE and ET remains a constant value.

Inputs:

IN : BOOL; (* Falling edge starts counting up ET *)

PT : TIME; (* Upper limit for counting up ET (delay time) *)

Outputs:

Q : BOOL; (* Gets a falling edge as soon as ET has reached the upper limit PV (delay time is over) *)

ET : current value of delay time

TOF(IN, PT, Q, ET) means:

If IN is TRUE, the outputs are TRUE and 0, respectively.

Q is FALSE when IN is FALSE and ET equal PT. Otherwise it is TRUE.

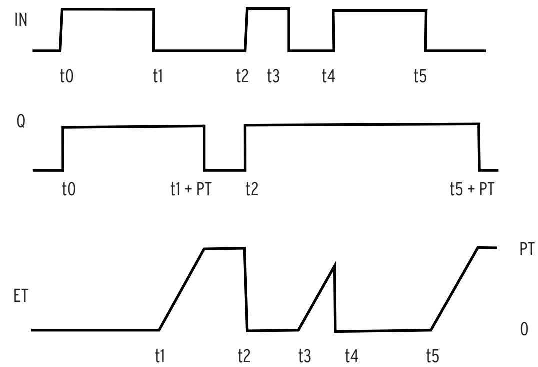


Figure B.87 Display of TOF Behavior Over Time

Declaration example:

TOFInst : TOF;

Example in ST:

TOFInst(IN := VarBOOL1, PT := T#5s);

VarBOOL2 := TOFInst.Q;

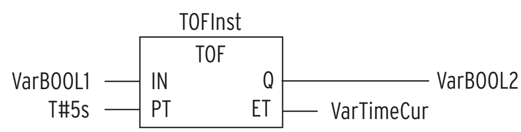


Figure B.88 TOF Example in CFC, LD

TON

This function block is a turn-on delay.

When the input changes from FALSE to TRUE, counter ET will begin counting in milliseconds to PT. When ET = PT, output Q is set TRUE and ET remains a constant value.

Inputs:

IN : BOOL; (* Rising edge starts counting up ET *)

PT : TIME; (* Upper limit for counting up ET (delay time) *)

Outputs:

Q : BOOL; (* Gets a rising edge as soon as ET has reached the upper limit PV (delay time is over) *)

ET : (* Current state of delay time *)

TON(IN, PT, Q, ET) means:

If IN is FALSE, Q is FALSE and ET is 0.

Q is TRUE when IN is TRUE and ET is equal to PT. Otherwise it is FALSE.

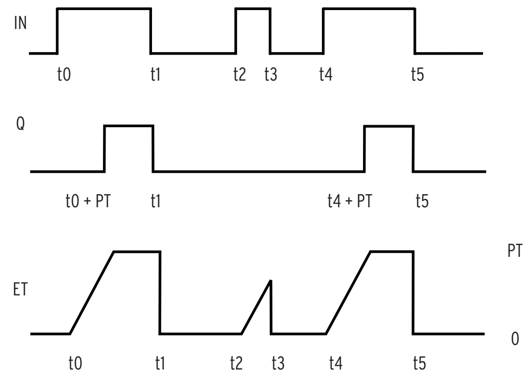


Figure B.89 Display of TON Behavior Over Time

Declaration example:

TONInst : TON;

Example in ST:

TONInst(IN := VarBOOL1, PT := T#5s);

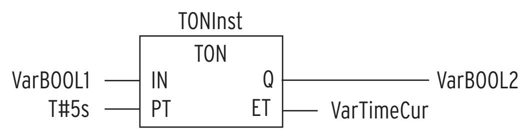


Figure B.90 TON Example in CFC, LD

TP

This function block triggers on a timer value.

A timer is counted up until a given limit is reached. During counting up a pulse variable is TRUE, otherwise it is FALSE.

Inputs:

IN : BOOL; (* A rising edge will start ET to count up in time *)

PT : TIME; (* Upper limit of the time *)

Outputs:

Q : BOOL; (* TRUE as long as the time is being counted up in ET (pulse) *)

ET : TIME; (* Current state of the time *)

TP(IN, PT, Q, ET) means:

If IN is FALSE, Q is FALSE and ET is 0.

When IN becomes TRUE, the time ET will begin to count up in milliseconds until ET = PT. It will then remain constant.

Q is TRUE when IN is TRUE and ET is less than or equal to PT. Otherwise it is FALSE.

Q returns a signal for the time period given in PT.

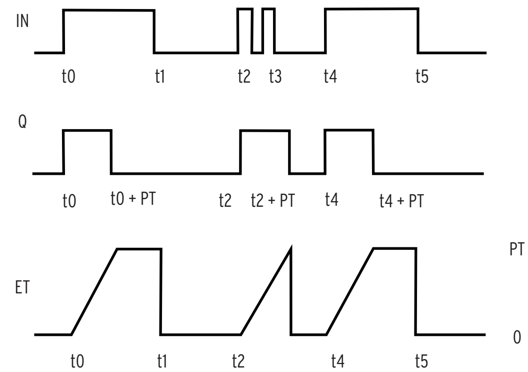


Figure B.91 Display of the TP Time Sequence

Declaration example:

TPInst : TP;

Example in ST:

```
TPInst(IN := VarBOOL1, PT := T#5s);
VarBOOL2 := TPInst.Q;
```

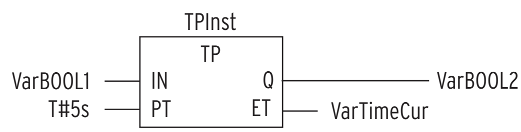


Figure B.92 TP Example in CFC, LD

Timers

TEX

This function block is an expiration timer. When the input changes from False to True, the counter ET begins counting in milliseconds to PT. When ET = PT, output Q is set to True for one processing cycle and ET remains at a constant value until a rising edge on IN is detected.

Inputs:

IN : BOOL;

PT : TIME;

Outputs:

Q : BOOL;

ET : TIME;

Declaration:

TEXinst : TEX;

Example in ST:

```
TEXinst(IN := VarBOOL1, PT := T#5s,
```

Q => VarBOOL3, ET => VarTIME);

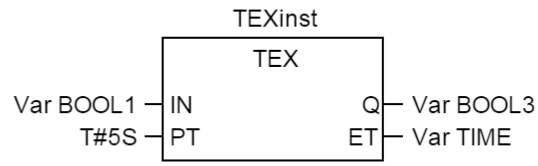


Figure B.93 TEX Example in CFC

TPUDO

This function block is a pickup dropout timer. When the input changes from False to True, the counter ETp begins counting in milliseconds to PUp. When ETp = PUp, output Q is set to True, and ET remains at a constant value. When the input then changes from True to False, the counter ETd begins counting in milliseconds to DOut. When ETd = DOut, output Q is set to False, and ET remains at a constant value.

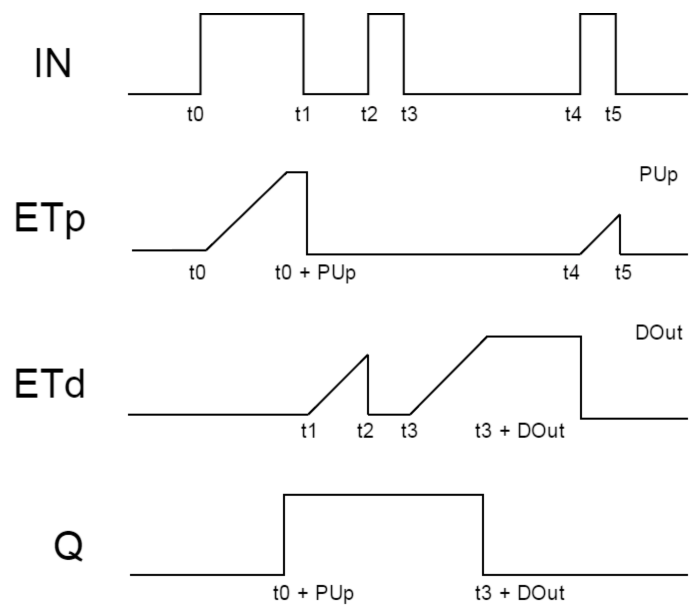


Figure B.94 TPUDO Diagram

Inputs:

IN : BOOL;

PUp : TIME;

DOut : TIME;

Outputs:

Q : BOOL;

ETp : TIME;

ETd : TIME;

Declaration:

TPUDOinst : TPUDO;

Example in ST:

TPUDOinst (IN := VarBOOL1, PUp:= T#5s, DOut := T#15s);

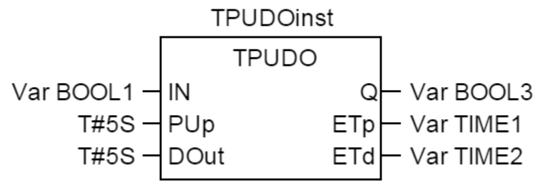


Figure B.95 TPUDO Example in CFC

Instructions

You can use the following instruction types in Structured Text.

Table B.9 Structured Text Instruction List

Instruction Type	Example
Assignment (see also *Assignment Operators on page 681*)	A := B; CV := CV + 1; C := SIN(X);
Calling a function block and use of the FB output	CMD_TMR(IN := °IX5, PT := 300); A := CMD_TMR.Q
RETURN	RETURN;

IF	D := B * B;
	IF D < 0.0 THEN
	C := A;
	ELSIF D = 0.0 THEN
	C := B;
	ELSE
	C := D;
	END_IF;

CASE	CASE INT1 OF 1: BOOL1 := TRUE; 2: BOOL2 := TRUE; ELSE BOOL1 := FALSE; BOOL2 := FALSE; END_CASE;

FOR	J := 101;
	FOR I := 1 TO 100 BY 2 DO
	IF ARR[I] = 70 THEN
	J := I;
	EXIT;
	END_IF;
	END_FOR;

WHILE	J := 1; WHILE J <= 100 AND ARR[J] <> 70 DO J := J + 2; END_WHILE;

Instruction Type	Example
REPEAT	J := 1; REPEAT J := J + 2; UNTIL J = 101 OR ARR[J] = 70 END_REPEAT;
EXIT	EXIT;
CONTINUE	CONTINUE;
Empty instruction	;

RETURN

You can use the RETURN instruction to leave a POU. You might want to do this, for example, in cases that depend on a condition.

Syntax:

RETURN;

Example:

IF b := TRUE THEN

RETURN;

END_IF;

a := a + 1;

If b is TRUE, instruction a := a + 1 will not be executed and the POU will be left immediately.

IF

With the IF instruction, you can check a condition and, depending upon this condition, execute instructions.

Syntax:

IF <Boolean_expression1> THEN

<IF_instructions>

{ELSIF <Boolean_expression2> THEN

<ELSIF_instructions1>

•

•

•

ELSIF <Boolean_expression *n*> THEN

<ELSIF_instructions *n*–1>

ELSE

<ELSE_instructions>}

END_IF;

The part in braces is optional.

If the <Boolean_expression1> returns TRUE, then only the <IF_Instructions> are executed. Otherwise, the software evaluates the Boolean expressions, beginning with <Boolean_expression2>, one after the other until one of the expressions returns TRUE. Then only those instructions after this Boolean expression and before the next ELSE or ELSIF are evaluated.

If none of the Boolean expressions produce TRUE, then only the <ELSE_instructions> are evaluated.

Example:

IF temp<17

THEN heating_on := TRUE;

ELSE heating_on := FALSE;

END_IF;

Here the heating in the example turns on when the temperature drops below 17 degrees. Otherwise, it remains off.

CASE

With the CASE instructions, you can combine several conditioned instructions with the same condition variable in one construct.

Syntax:

CASE <var1> OF

<Value1> : <Instruction 1>

<Value2> : <Instruction 2>

<Value3, Value4, Value5> : <Instruction 3>

<Value6 ..., Value10> : <Instruction 4>

•

•

•

<Value *n*> : <Instruction *n*>

ELSE <ELSE instruction>

END_CASE;

A CASE instruction is processed according to the following model:

➤ If the variable in <var1> has the value <Value *i*>, then the instruction <Instruction *i*> is executed.

➤ If <Var1> has none of the indicated values, then the <ELSE Instruction> is executed.

- To execute the same instruction for several values of the variables, write these values one after the other separated by commas.
- To execute the same instruction for a value range of a variable, write the initial value and the end value separated by two periods.

Example:

```
CASE INT1 OF
1, 5 : BOOL1 := TRUE;
BOOL3 := FALSE;
2 : BOOL2 := FALSE;
BOOL3 := TRUE;
10..20 : BOOL1 := TRUE;
BOOL3 := TRUE;
ELSE
BOOL1 := NOT BOOL1;
BOOL2 := BOOL1 OR BOOL2;
END_CASE;
```

FOR Loop

With the FOR loop, you can program repeated processes.

Syntax:

```
INT_Var : INT;
FOR <INT_Var> := <INIT_VALUE> TO <END_VALUE> {BY <Step size>} DO
<Instructions>
END_FOR;
```

The part in braces {} is optional.

The <Instructions> are executed as long as the counter <INT_Var> is not greater than the <END_VALUE>. This evaluation occurs first so that the <instructions> are never executed if <INIT_VALUE> is greater than <END_VALUE>.

When <Instructions> are executed, <INT_Var> is incremented by <Step size>. The step size can have any integer value. If it is missing, then it is set to 1.

NOTE
If <END_VALUE> is equal to the limit value of counter <INT_VAR>, e.g., if Counter—used in the previous example—is of type SINT and if <END_VALUE> is 127, then you will get an endless loop. Therefore, <END_VALUE> must not be equal to the limit value of the counter.

Example:

```
FOR Counter := 1 TO 5 BY 1 DO
```

```
var1 := var1 * 2;
END_FOR;
Erg := var1;
```

Let us assume that the default setting for var1 is "1". Then, it will have the value 32 after the FOR loop.

You can use the CONTINUE instruction within a FOR loop.

WHILE Loop

You can use the WHILE loop in a manner similar to the FOR loop, but the break-off condition can be any Boolean expression. This means that you must indicate a condition that will cause the loop to execute.

Syntax:

```
WHILE <Boolean expression>
<Instructions>
END_WHILE;
```

The <Instructions> are executed repeatedly as long as the <Boolean_expression> returns TRUE. If the <Boolean_expression> is already FALSE at the first evaluation, then the <Instructions> are never executed. If <Boolean_expression> never is evaluated to the value FALSE, then the <Instructions> repeat endlessly. Endless tight loops will starve the RTAC of CPU resources and cause a watchdog timer violation.

NOTE
You must ensure that no endless loop develops. You can do this by changing the condition in the instruction part of the loop. One method, for example, might be to count up or down one counter.

Example:

```
WHILE counter<>0 DO
    var1 := var1 * 2;
    Counter := Counter – 1;
END_WHILE
```

The WHILE and REPEAT loops are, in a certain sense, more powerful than the FOR loop because you do not need to know the number of cycles before executing the loop. In some cases, therefore, you will only be able to work with these two loop types. If, however, you know the number of the loop cycles, then a FOR loop is preferable because it prevents inadvertent endless loops.

You can use the CONTINUE instruction within a WHILE loop.

REPEAT Loop

The REPEAT loop differs from the WHILE loop because the break-off condition is checked only after execution of the loop. This means that the loop will run through at least once, regardless of the wording of the break-off condition.

Syntax:

```
REPEAT
<Instructions>
UNTIL <Boolean expression>
END_REPEAT;
```

The <Instructions> are executed until the <Boolean expression> returns TRUE.

If <Boolean expression> is evaluated as TRUE in the first pass, then <Instructions> are executed only once. If <Boolean_expression> is never evaluated as TRUE, then the <Instructions> repeat endlessly.

Example:

```
REPEAT
    var1 := var1 * 2;
    Counter := Counter – 1;
UNTIL
Counter = 0
END_REPEAT;
```

You can use the CONTINUE instruction within a REPEAT loop.

EXIT

If the EXIT instruction appears in a FOR, WHILE, or REPEAT loop, then the innermost loop ends, regardless of the break-off condition.

CONTINUE

As an extension to the IEC 61131-3 standard, the CONTINUE instruction is supported within FOR, WHILE and REPEAT loops. CONTINUE effectively ends the present iteration of the loop as though the loop completed normally and the next iteration began. The difference is that all instructions following the CONTINUE statement in the presently executing loop are skipped.

Example:

```
FOR Counter := 1 TO 5 BY 1 DO
IF var1 = 0 THEN
CONTINUE; (* to avoid division by zero, skip to END_FOR *)
END_IF
var1 := var1*2; (* only executed, if var1 is not "0" *)
END_FOR;
Erg := var1;
```

Declarations

You must declare all tags (variables) properly within an RTAC project either in a POU, GVL, or the DTE. In most cases, the ACSELERATOR RTAC software will declare the tags for you automatically. For example, when you add a tag to a protocol, ACSELERATOR RTAC automatically generates and declares the data type all the time, quality, and other variables associated with that tag.

You can also enter tag declarations directly into the declaration portion of a POU, a GVL window, or the DTE.

Note the difference between a data type and a tag type. As you will see in following examples, the tag type declarations are in blocks prefaced by the <Type> command and suffixed with END_<Type>. The common tag types are as follows:

- VAR
- VAR_INPUT
- VAR_OUTPUT
- VAR_IN_OUT
- VAR_GLOBAL
- VAR_TEMP
- VAR_STAT

Attributes

You can also supplement the tag type keywords with attributes, which are also keywords. For instance, RETAIN (VAR_INPUT RETAIN) is an attribute that will enable the tag to keep its value even through a power cycle. The declaration of a tag must match the following rules.

Syntax:

<Identifier> : <Type> {:= <initialization>};

Items in braces {} are optional. See *Data Types on page 747* and *Tag Initialization on page 747*.

Identifiers

An identifier is the name of a tag. The following list includes basic identifier naming rules:

- Must not contain spaces or special characters.
- Not case-sensitive; VAR1, Var1, and var1 are all the same tag.
- Underscore character is recognized; A_BCD and AB_CD are two different tags. An identifier must not have more than one underscore character sequentially.
- Length is not limited.

Namespaces—Scope of Tag Identifiers

You can use the same tag identifier more than once in a project, but ACSELERATOR RTAC requires that a complete tag name (including namespace) be unique. This means that two tags can have the same tag name if you declare these tags and use them in two different programs. This is because the tags are local to the individual programs in which you declare them, so their namespace is different. Two tag names cannot have the same name if they reside in the same namespace or program. The software will verify the following rules:

- An identifier must not be used locally more than once.
- An identifier must not be identical to any IEC 61131-3 keyword.
- A local identifier can be the same as a global identifier; in this case, the local instance will take priority within the context of the POU in which you create the local identifier. The global scope operator "." is an exception to this rule. If you reference a global tag ivar that shares a name with a local tag by using the global scope operator .ivar, the code will reference the global instance.
- Tags that you define in two different GVLs can have the same identifier. Use the following to distinguish these tags.
- Use the name of the GVL as a namespace for the included tags. In this way, you can access tags you have declared with the same name in different GVLs by entering the list name, separated by a period, before a tag name. For example, gvl1.myVar is distinguished from gvl2.myVar because the namespace of the GVL is specified.
- You can access tags defined in GVLs of included libraries by using the syntax <library namespace>.<name of GVL>.<tag name>.
- A library is also a namespace within your project, so you can access a library module or tag by using the syntax <library namespace>.<modulename|tagname>.

Local Tags—VAR

Declare all local tags for a POU between the keywords VAR and END_VAR. These tags cannot be read or written to from outside the POU in which they are declared.

You can add *Attributes* to a VAR. Example:

```
VAR
    iLoc1 : INT; (* 1. Local Tag *)
END_VAR
```

Input Tags—VAR_INPUT

Declare all local tags that serve as inputs for a POU between the keywords VAR_INPUT and END_VAR. Assign the value of these tags within the POU calling statement.

You can add *Attributes* to a VAR_INPUT.

Example:

```
VAR_INPUT
    iIn1 : INT; (* 1. Input Tag *)
END_VAR
```

Output Tags—VAR_OUTPUT

Declare all tags that serve as outputs of a POU between the keywords VAR_OUTPUT and END_VAR. These values are referenced to the calling POU.

You can add *Attributes* to a VAR_OUTPUT.

Example:

```
VAR_OUTPUT
    iOut1 : INT; (* 1. Output Tag *)
END_VAR
```

Output tags in functions and methods

According to IEC 61131-3 draft 2, functions (and methods) can have additional outputs. You must assign these in the call of the function in a manner similar to the following:

```
fun(iIn1 := 1, iIn2 := 2, iOut1 => iLoc1, iOut2 => iLoc2);
```

ACSELERATOR RTAC will write the return value of the function fun to the locally declared tags loc1 and loc2.

Input and Output Tags—VAR_IN_OUT

Declare all tags that serve as inputs and outputs for a POU between the keywords VAR_IN_OUT and END_VAR.

NOTE
With tags of this type, the value of the transferred tag changes (i.e., transferred as a pointer, Call-by-Reference). That means that the input value for such tags cannot be a constant. For this reason, you cannot read or write the VAR_IN_OUT tags of a function block directly from outside via <functionblockinstance><in/outputtag>.

Example:

```
VAR_IN_OUT
    iInOut1 : INT; (* 1. InputOutput Tag *)
END_VAR
```

Global Tags—VAR_GLOBAL

Use the GVL to declare tags or constants global if you need them to be recognized throughout the project.

NOTE
A tag that you define locally in a POU with the same name as a global tag will have priority within the POU.

Use GVLs to structure and handle global tags within a project. Insert a GVL from the IEC 61131-3 ribbon tab and declare the global tags between the keywords VAR_GLOBAL and END_VAR. You can add Attributes to a VAR_GLOBAL.

ACSELERATOR RTAC recognizes a tag as global by a preceding dot, e.g., .iGlobVar1, which identifies a global scope operator.

NOTE
If you define a GVL but do not use any of the variables in the project, the compile will optimize the GVL out of the project build. As a result, when you are online you cannot force or watch those variables.

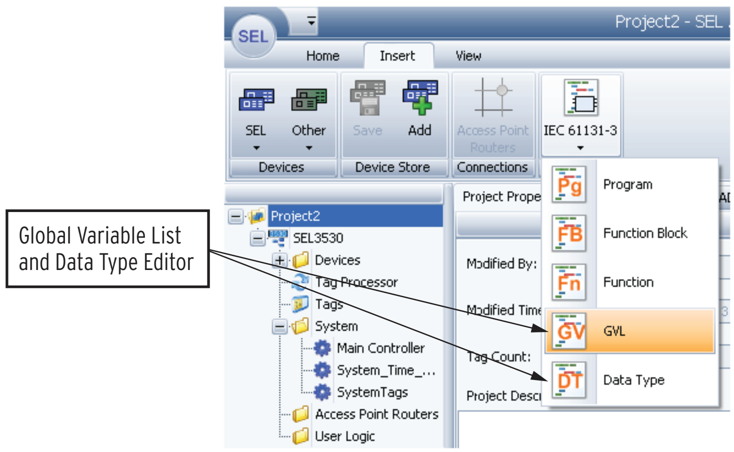


Figure B.96 GVL and DTE

Temporary Tags—VAR_TEMP

This feature is an extension to the IEC 61131-3 standard.

VAR_TEMP declarations are possible only within programs and function blocks. You can access temporary tags only within the body of the program or function block in which they are declared. VAR_TEMP declared tags are (re)initialized every time the POU is called.

Declare the temporary tag declarations locally between the keywords VAR_TEMP and END_VAR.

Static Tags—VAR_STAT

This feature is an extension to the IEC 61131-3 standard.

Use static tags in function blocks, methods, and functions. Declare these tags locally between the keywords VAR_STAT and END_VAR; they will be initialized at the first call of the respective POU.

While you can access static tags only within the scope in which you declare these tags (such as a static variable in C), these tags retain a value even after the system switches context. For example, a function might use such a tag as a counter for the number of function calls.

You can add *Attributes* to a VAR_STAT.

Tag Initialization

The default initialization value is 0 for all declarations, but you can add user-defined initialization values in the declaration of each tag and data type.

Use the assignment operator ":=" to create user-defined initialization, which can be any valid Structured Text expression. Use constant values as well as other tags or functions to define the initial value. Verify that a tag you use for the initialization of another tag is already initialized itself.

Examples for valid tag initializations:

```
VAR

var1 : INT := 12; (* Integer tag with initial value of 12 *)

x : INT := 13 + 8; (* initialization value defined by an expression with
constants *)

y : INT := x + fun(4); (* initialization value defined by an expression
containing a function call; be aware of the order in this case! *)

END_VAR
```

You can initialize a tag with any valid Structured Text expression by accessing tags outside of the scope of the tag you are initializing, even by calling functions. However, ensure first that a tag you use for initialization of another tag is already initialized.

Examples for valid tag initializations:

```
VAR

x : INT := 13 + 8;

y : INT := x + fun(4);

END_VAR
```

Data Types

Standard types are defined explicitly as part of the IEC 61131-3 standard; extended types are not. The data type assigned to each tag dictates the amount of reserved memory space will be reserved for the tag and the type of values it stores.

Standard Data Types

ACSELERATOR RTAC supports all data types the IEC 61131-3 standard describes. Standard data types include the following: BOOL, Integer Data Types, REAL/LREAL, STRING, Time Data Types.

BOOL

BOOL type variables can take either of the two values TRUE (1) and FALSE (0). Eight bits of memory space are reserved for each BOOL type variable.

See also *BOOL Constants on page 676*.

Integer Data Types

See *Table B.10* for a list of all available integer data types. Each of the different number types covers a different range of values. The following range limitations apply to the integer data types:

Table B.10 Integer Data Types

Type	Lower Limit	Upper Limit	Memory Space
BYTE	0	255	8 Bit
WORD	0	65535	16 Bit
DWORD	0	4294967295	32 Bit
LWORD	0	(2^{63}) - 1	64 Bit
SINT	–128	127	8 Bit
USINT	0	255	8 Bit
INT	–32768	32767	16 Bit
UINT	0	65535	16 Bit
DINT	–2147483648	2147483647	32 Bit
UDINT	0	4294967295	32 Bit
LINT	(-2^{64}) / 2	(2^{64}) / 2 - 1	64 Bit
ULINT	0	2_{64} - 1	64 Bit

Note on date type names: D = double, L = long, S = short, U = unsigned

See also *Number Constants on page 678*.

NOTE
A type conversion from a larger to smaller type can cause loss of information.

REAL/LREAL

REAL and LREAL are floating-point types. They must represent rational numbers. The RTAC reserves 32 bits of memory space for REAL and 64 bits for LREAL types.

Value range for REAL: 1.175494351e-38F to 3.402823466e+38F

Value range for LREAL: 2.2250738585072014e-308 to 1.7976931348623158e +308

See also *REAL Constants on page 679*.

STRING

A STRING type variable can contain any string of characters. The size entry in the declaration determines how much memory space is reserved for the variable. You can place this entry, which refers to the number of characters in the string, in parentheses or square brackets. If you provide no size specification, the default size will be 80 characters. The maximum string size is 255 characters in one string variable.

Example of a string declaration with 35 characters:

```
str : STRING(35) := 'This is a String';
```

See also *WSTRING on page 750* and *STRING Constants on page 679*.

Time Data Types

The data types TIME, TIME_OF_DAY (abb. TOD), DATE, and DATE_AND_TIME (abb. DT) are handled internally, similarly to DWORD.

Time is in milliseconds in TIME and TOD. Time in TOD begins at 12:00 a.m.

Time is in seconds in DATE and DT and begins with January 1, 1970, at 12:00 a.m.

See also *LTIME on page 750* (which in extension to the IEC 61131-3 standard is available as a 32-bit time data type), *TIME Constants on page 676*, *DATE Constants on page 677*, *TIME_OF_DAY Constants on page 678*, and *DATE_AND_TIME Constants on page 678*.

Extended Data Types and Data Classes

In addition to the data types listed in the IEC 61131-3 standard, there are some extended data types available implicitly in ACSELERATOR RTAC:

- *UNION on page 749*
- *LTIME on page 750*
- *WSTRING on page 750*

UNION

An extension to the IEC 61131-3 standard makes it possible to declare unions in user-defined types.

In a union, all components have the same offset; they all occupy the same storage location. Thus, assuming a union definition as in the following example, an assignment to name.a would also manipulate name.b. The storage location for a union will be the size of the largest member of the union.

Example:

TYPE name : UNION

a : LREAL;

b : LINT;

END_UNION

END_TYPE

LTIME

ACSELERATOR RTAC supports LTIME as an extension to the IEC 61131-3 to provide a time base for high-resolution timers. LTIME is 64 bits in size and has resolution in nanoseconds.

Syntax:

LTIME#<time declaration>

Include in the time declaration all of the time units necessary for the TIME constant and the following:

us : microseconds

ns : nanoseconds

Example:

LTIME1 := LTIME#1000d15h23m12s34ms2us44ns

Compare to TIME : 32-bit size and resolution in milliseconds.

WSTRING

This string data type is an extension to the IEC 61131-3 standard.

It differs from the standard STRING type (ASCII), in that it is interpreted in Unicode format.

Example:

wstr : WSTRING := "This is a WString";

See also *STRING on page 749* and *STRING Constants on page 679*.

ACSELERATOR RTAC Data Types and Data Classes

ACSELERATOR RTAC has some special extended data types and data classes that follow the IEC 61850 specification closely. Each data class has attributes such as time, quality, and status that define the data further. You can define all protocol data and system tags by one of these data types.

ACSELERATOR RTAC supports the following data classes:

➤ *APC (Controllable Analog Set Point) on page 751*
➤ *BCR (Binary Counter Reading) on page 752*
➤ *BSC (Binary Controlled Step Position Information) on page 753*

APC (Controllable Analog Set Point)

This data class is an extension to the IEC 61131-3 standard. APC is a data structure with the following elements. When the RTAC detects a rising-edge change on trigger, it sends the setMag value.

Table B.11 APC Attributes

Attributes		Type	Default Value Enumeration
oper		operAPC	
	setMag	REAL	0
	trigger	BOOL	
	q	quality_t	See *q (quality_t) on page 766*
	t	timeStamp_t	See *timeAccuracy_t on page 767*
origin		originator_t	See *orCat_t on page 762*
status		MV	See *MV (Measured Value) on page 761*
	instMag	REAL	0
	mag	REAL	0
	range	range_t	normal, high, low, high_high, low_low

Attributes			Type	Default Value Enumeration
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *timeAccuracy_t on page 767*
	db		REAL	100
	zeroDb		REAL	2
	rangeC		rangeConfigReal_t	
		hhLim	REAL	1E+36
		hLIM	REAL	1E+35
		lLim	REAL	–1E+35
		llLim	REAL	–1E+36
		minVal	REAL	–1E+37
		maxVal	REAL	1E+37
origin			originator_t	See *orCat_t on page 762*

BCR (Binary Counter Reading)

This data class is an extension to the IEC 61131-3 standard. BCR is a data structure with the following elements.

Table B.12 BCR Attributes

Attributes			Type	Default Value Enumeration
actVal			UDINT	0
frVal			UDINT	0
frTm			timeStamp_t	
	value		dateTime_t	
		dateTime	DATE_AND_TIME	DT#2000-8-3-20:2:3
		uSec	UDINT	369704
	quality		timeQuality_t	
		leapSecondsKnown	BOOL	TRUE, FALSE
		clockFailure	BOOL	TRUE, FALSE
		clockNotSynchronized	BOOL	TRUE, FALSE
		accuracy	timeAccuracy_t	unspecified See *timeAccuracy_t on page 767*
	daylight_savings_time		DST_t	
		enabled	BOOL	TRUE, FALSE
		activated	BOOL	TRUE, FALSE
		offset	INT	0
	UTC_Offset		INT	0
	source		timeSource	
		value	timeSource_t	Null, IRIG_B, NTP, DNP, Free_Running, IED_Protocol, i870, PTP

Attributes	Type	Default Value Enumeration
priority	USINT	0
q	quality_t	See *q (quality_t) on page 766*
t	timeStamp_t	See *timeAccuracy_t on page 767*
frRs	BOOL	TRUE, FALSE

BSC (Binary Controlled Step Position Information)

This data class is intended to facilitate a mapping between objects within the IEC 61131-3 logic engine and IEC 61850 control objects of the BSC controllable data class defined in the IEC 61850 7-3 standard.

Table B.13 BSC Attributes

Attributes	Type	Default Value Enumeration
oper	operBSC	stop, lower, higher, tcmd_reserved
status	INS	
origin	originator_t	See *orCat_t on page 762*

CMV (Complex Measured Value)

This data class is an extension to the IEC 61131-3 standard. CMV is a data structure with the element shown in *Table B.14*. The angle in CMV is provided in degrees. The built-in IEC 61131 trigonometry functions in the RTAC use radians. Use the following functions to convert from degrees to radians:

deg_to_rad() returns degrees as a REAL

rad_to_deg()

returns radians as a REAL

Table B.14 CMV Attributes

Attributes		Type	Default Value Enumerations
instCVal		vector_t	
	mag	REAL	
	ang	REAL	0
cVal		vector_t	
	mag	REAL	
	ang	REAL	0
range		range_t	normal, high, low, high_low, low_low
q		quality_t	See *q (quality_t) on page 766*
t		timeStamp_t	See *t (timeStamp_t) on page 766*
db		REAL	0
zeroDb		REAL	0
rangeC		RangeConfig_t	

Attributes	Type	Default Value Enumerations
hhLim	REAL	0
hLim	REAL	0
lLim	REAL	0
llLim	REAL	0
minVal	REAL	0
maxVal	REAL	0

CRC (Cyclic Redundancy Check)

This data class is an extension to the IEC 61131-3 standard. CRC is a data structure with the following elements.

Table B.15 CRC Attributes

Attributes	Type	Default Value Enumerations
width	crcWidth_t	8, 16, 32
polynomial	STRING	
initial	STRING	
revDataByteOrder	BOOL	TRUE, FALSE
revResultByteOrder	BOOL	TRUE, FALSE
final XOR	STRING	

CSM (Checksum)

This data class is an extension to the IEC 61131-3 standard. CSM is a data structure with the following elements.

Table B.16 CSM Attributes

Attributes	Type	Default Value Enumeration
Enumeration CSM_type	checksum_t	checksum, checksum_16_bit_byte, checksum_xor
initVal	UDINT	0

DNPC (DNP Controllable Single Point)

This data class is an extension to the IEC 61131-3 standard. DNPC is a data structure with the following elements.

Table B.17 DNPC Attributes

Attributes	Type	Default Value Enumeration
operPulse	operSPC	
ctlVal	BOOL	TRUE, FALSE
q	quality_t	See *q (quality_t) on page 766*
t	timeStamp_t	See *t (timeStamp_t) on page 766*

Attributes			Type	Default Value Enumeration
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persist
		onDur	UDINT	2000 (ms)
		offDur	UDINT	2000 (ms)
		numPls	UDINT	1
	origin		originator_t	See *orCat_t on page 762*
operLatchOn			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persist
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
	origin		originator_t	See *orCat_t on page 762*
operLatchOff			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persist
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
	origin		originator_t	See *orCat_t on page 762*
operTrip			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	2000 (ms)
		offDur	UDINT	2000 (ms)
		numPls	UDINT	1
	origin		originator_5	See *orCat_t on page 762*

Attributes			Type	Default Value Enumeration
operClose			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	2000 (ms)
		offDur	UDINT	2000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
status			SPS	See *SPS (Single Point Status) on page 764*

DPC (Controllable Double Point)

This data class is for a double point controllable output.

Table B.18 DPC Attributes

Attributes	Type	Default Value Enumerations
operSet	operSPC	persist
operClear	operSPC	persist
status	DPS	
origin	originator_t	See *orCat_t on page 762*

DPS (Double Point Status)

This data class is intended to model the DPS common data class as defined in the IEC 61850-7-3 standard.

Table B.19 DPS Attributes

Attributes	Type	Default Value Enumerations
stVal	dbpos_t	dbpos_intermediate, dbpos_off, dbpos_on, dbpos_bad
q	quality_t	See *q (quality_t) on page 766*
t	timeStamp_t	See *timeAccuracy_t on page 767*

I870SC (Single Command Control)

This data class is based on the IEC 60870-5 standard and is used for single command control.

Table B.20 I870SC Attributes

Attributes	Type	Default Value Enumerations
operSet	operSPC	persist
operClear	operSPC	persist
operPulse	operSPC	pulse
status	SPS	
origin	originator_t	See *orCat_t on page 762*

I870DC (Double Command Control)

This data class is based on the IEC 60870-5 standard and is used for double command control.

Table B.21 I870DC Attributes

Attributes	Type	Default Value Enumerations
operSet	operSPC	persist
operClear	operSPC	persist
operPulse	operSPC	pulse
status	DPS	
origin	originator_t	See *orCat_t on page 762*

INC (Controllable Integer)

This data class is an extension to the IEC 61131-3 standard. INC is a data structure with the following elements. When the RTAC detects a rising-edge change on trigger, it sends the ctlVal value.

Table B.22 INC Attributes

Attributes		Type	Default Value Enumerations
operINC		operINC	
	ctlVal	DINT	0
	trigger	BOOL	
	q	quality_t	See *q (quality_t) on page 766*
	t	timeStamp_t	See *t (timeStamp_t) on page 766*
status		INS	See *INS (Integer Status) on page 757*
origin		originator_t	See *orCat_t on page 762*

INS (Integer Status)

This data class is an extension to the IEC 61131-3 standard. INS is a data structure with the following elements.

Table B.23 INS Attributes

Attributes		Type	Default Value Enumerations
stVal		DINT	0
range		range_t	normal, high, low, high_high, low_low
q		quality_t	See *q (quality_t) on page 766*
t		timeStamp_t	See *t (timeStamp_t) on page 766*
rangeC		rangeConfigDint_t	
	hhLim	DINT	1932735282
	hLim	DINT	1717986918
	lLim	DINT	–1717986918
	llLim	DINT	–1932735282
	minVal	DINT	–2147483648
	maxVal	DINT	2147483647

IOC (Input/Output Control)

This data class is an extension to the IEC 61131-3 standard. IOC is a data structure with the following elements.

Table B.24 IOC Attributes

Attributes			Type	Default Value Enumerations
operSet			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	CMDQUAL_T	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
operClear			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	CMDQUAL_T	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1

Attributes			Type	Default Value Enumerations
operPulse			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persist
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
status			SPS	See *SPS (Single Point Status) on page 764*

LBCR (Long Binary Counter Reading)

This data class is intended to model the BCR binary counter reading data class as defined in the IEC 61850-7-3 Ed. 2 standard.

Table B.25 LBCR Attributes

Attributes	Type	Default Value Enumerations
actVal	LINT	
frVal	LINT	
frTm	timeStamp_t	See *timeAccuracy_t on page 767*
q	quality_t	See *q (quality_t) on page 766*
t	timeStamp_t	See *timeAccuracy_t on page 767*
frRs	BOOL	False

LEDC (LED Indication Lamp Control)

This data class is an extension to the IEC 61131-3 standard. LEDC is a data structure with the following elements.

Table B.26 LEDC Attributes

Attributes		Type	Default Value Enumerations
onColor		colorRG_t	off, red, green
offColor		colorRG_t	off, red, green
operSet		operSPC	
	ctlVal	BOOL	TRUE, FALSE
	q	quality_t	See *q (quality_t) on page 766*
	t	timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig	pulseConfig_t	

Attributes			Type	Default Value Enumerations
		cmdQual	CMDQUAL_T	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
operClear			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
status			SPS	See *SPS (Single Point Status) on page 764*

MDBC (Modbus Coil Control)

This data class is an extension to the IEC 61131-3 standard. MDBC is a data structure with the following elements.

Table B.27 MDBC Attributes

Attributes			Type	Default Value Enumerations
operSet			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	CMDQUAL_T	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
operClear			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)

Attributes			Type	Default Value Enumerations
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
status			SPS	See *SPS (Single Point Status) on page 764*

MRBC (MIRRORED BITS Control)

This data class is an extension to the IEC 61131-3 standard. MRBC is a data structure with the following elements.

Table B.28 MRBC Attributes

Attributes			Type	Default Value Enumerations
operSet			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	CMDQUAL_T	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
operClear			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
status			SPS	See *SPS (Single Point Status) on page 764*

MV (Measured Value)

This data class is an extension to the IEC 61131-3 standard. MV is a data structure with the following elements.

Table B.29 MV Attributes

Attributes		Type	Different Value Enumerations
instMag		REAL	
mag		REAL	
range		range_t	normal, high, low, high_high, low_low
q		quality_t	See *q (quality_t) on page 766*
t		timeStamp_t	See *t (timeStamp_t) on page 766*
db		REAL	0
zeroDb		REAL	0
rangeC		rangeConfigReal_t	
	hhLim	REAL	1E+36
	hLim	REAL	1E+35
	lLim	REAL	–1E+35
	llLim	REAL	–1E+36
	minVal	REAL	–1E+37
	maxVal	REAL	1E+37

orCat_t

The attribute orCAT_t is an enumerated attribute of originator_t. The undefined_origin value is not defined by the standard and is therefore interpreted as an initialized variable or variable with an unknown state. The enumerated values of this attribute are as follows:

- orcat_undefined_origin := –1,
- orcat_not_supported := 0,
- orcat_bay_control := 1,
- orcat_station_control := 2,
- orcat_remote_control := 3,
- orcat_automatic_bay := 4,
- orcat_automatic_station := 5,
- orcat_automatic_remote := 6,
- orcat_maintenance := 7,
- orcat_process := 8

SPC (Controllable Single Point)

This data class is an extension to the IEC 61131-3 standard. SPC is a data structure with the following elements.

Table B.30 SPC Attributes

Attributes		Type	Default Value Enumerations
operSet		operSPC	
	ctlVal	BOOL	TRUE, FALSE

Attributes			Type	Default Value Enumerations
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
operClear			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
status			SPS	See *SPS (Single Point Status) on page 764*

SBRC (SEL Breaker Bit Control)

This data class is an extension to the IEC 61131-3 standard. SBRC is a data structure with the following elements.

Table B.31 SBRC Attributes

Attributes			Type	Default Value Enumerations
operTrip			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
operClose			operSPC	
	ctlVal		BOOL	TRUE, FALSE

Attributes			Type	Default Value Enumerations
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*

SPS (Single Point Status)

This data class is an extension to the IEC 61131-3 standard. SPS is a data structure with the following elements.

Table B.32 SPS Attributes

Attributes	Type	Default Value Enumerations
stVal	BOOL	TRUE, FALSE
q	quality_t	See *q (quality_t) on page 766*
t	timeStamp_t	See *t (timeStamp_t) on page 766*

SRBC (SEL Remote Bit Control)

This data class is an extension to the IEC 61131-3 standard. SRBC is a data structure with the following elements.

Table B.33 SRBC Attributes

Attributes			Type	Default Value Enumerations
operSet			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
operClear			operSPC	
	ctlVal		BOOL	TRUE, FALSE

Attributes			Type	Default Value Enumerations
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*
operPulse			operSPC	
	ctlVal		BOOL	TRUE, FALSE
	q		quality_t	See *q (quality_t) on page 766*
	t		timeStamp_t	See *t (timeStamp_t) on page 766*
	pulseConfig		pulseConfig_t	
		cmdQual	cmdQual_t	pulse, persistent
		onDur	UDINT	1000 (ms)
		offDur	UDINT	1000 (ms)
		numPls	UDINT	1
origin			originator_t	See *orCat_t on page 762*

STR (String)

This data class is an extension to the IEC 61131-3 standard. STR is a data structure with the following elements.

Table B.34 STR Attributes

Attributes	Type	Default Value Enumerations
strVal	STRING(255)	
q	quality_t	See *q (quality_t) on page 766*
t	timeStamp_t	See *timeAccuracy_t on page 767*

TIM (Time)

This data class is an extension to the IEC 61131-3 standard. TIM is a data structure with the following elements.

Table B.35 TIM Attributes

Attributes	Type	Default Value Enumerations
timeVal	TIME	
q	quality_t	See *q (quality_t) on page 766*
t	timeStamp_t	See *timeAccuracy_t on page 767*

Data Attributes

Some data attributes such as quality (q) and time (t) have their own data type, specified with an underscore and lowercase "t" (_t). The attributes listed in the following tables provide greater specification about each tag value.

q (quality_t)

Table B.36 quality_t Attributes

Attributes			Type	Default Value Enumerations
q			quality_t	
	validity		validity_t	good, invalid, reserved, questionable
	detailQual		detailQual_t	
		overflow	BOOL	TRUE, FALSE
		outOfRange	BOOL	TRUE, FALSE
		badReference	BOOL	TRUE, FALSE
		oscillatory	BOOL	TRUE, FALSE
		failure	BOOL	TRUE, FALSE
		oldData	BOOL	TRUE, FALSE
		inconsistent	BOOL	TRUE, FALSE
		inaccurate	BOOL	TRUE, FALSE
	source		source_t	process, substituted
	test		BOOL	TRUE, FALSE
	operatorBlocked		BOOL	TRUE, FALSE

t (timeStamp_t)

Table B.37 timeStamp_t Attributes

Attributes			Type	Default Value Enumerations
t			timeStamp_t	
	value		dateTime_t	
		dateTime	DT	1970-1-1-0:0:0
		uSec	UDINT	0

Attributes		Type	Default Value Enumerations
quality		timeQuality_t	
	leapSecondsKnown	BOOL	TRUE, FALSE
	clockFailure	BOOL	TRUE, FALSE
	clockNotSynchronized	BOOL	TRUE, FALSE
	accuracy	timeAccuracy_t	unspecified (See *timeAccuracy_t on page 767*)
daylight_savings_time		DST_t	
	enabled	BOOL	TRUE, FALSE
	activated	BOOL	TRUE, FALSE
	offset	DINT	0
UTC_Offset		INT	0
source		timeSource	
	value	timeSource_t	Null, IRIG_B, NTP, DNP, Free_Running, IED_Protocol, i870, PTP
	priority	USINT	0

timeAccuracy_t

This data class is an extension to the IEC 61131-3 standard. The attribute timeAccuracy_t is an enumerated attribute of many data custom data classes. The enumerated values of this attribute are as follows:

- T0 := 7 (10 ms)
- T1 := 10 (1 ms)
- T2 := 14 (0.1 ms)
- T3 := 16 (0.025 ms)
- T4 := 18 (0.004 ms)
- T5 := 20 (0.001 ms)
- unspecified := 31

Create Custom Data Type Structures and Enumerations

Select **Insert > User Logic > Data Type** to create your own data type structure or enumeration.

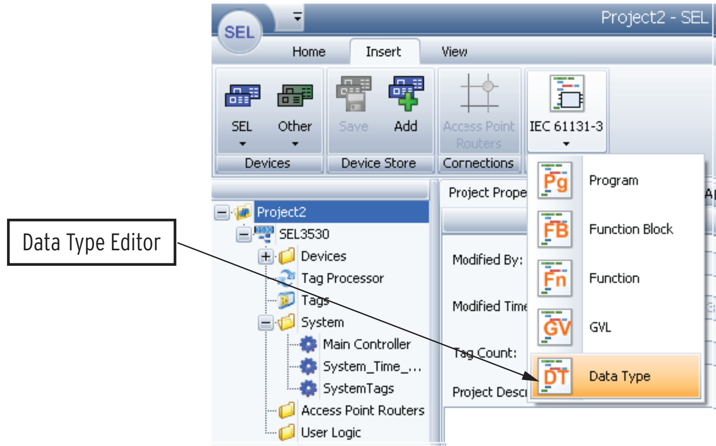


Figure B.97 Data Type Editor

Structures

From the **Insert** ribbon, select the IEC 61131-1 icon to create structures as Data Type Unit (DUT) objects.

Structures begin with the keywords TYPE and STRUCT and end with END_STRUCT and END_TYPE.

The syntax for structure declarations is as follows:

```
TYPE <structurename>:
STRUCT
<declaration of variables 1>
•
•
•
<declaration of variables n>
END_STRUCT
END_TYPE
```

<structurename> is a type that ACSELERATOR RTAC recognizes throughout the project. You can use this type similarly to a standard data type.

ACSELERATOR RTAC allows interlocking structures, but variables cannot be assigned to addresses (ACSELERATOR RTAC does not allow the AT declaration).

Example for a structure definition named polygonline:

```
TYPE Polygonline:
STRUCT
    Start : ARRAY [1..2] OF INT;
    Point1 : ARRAY [1..2] OF INT;
    Point2 : ARRAY [1..2] OF INT;
```

Point3 : ARRAY [1..2] OF INT;

Point4 : ARRAY [1..2] OF INT;

End : ARRAY [1..2] OF INT;

END_STRUCT

END_TYPE

Initialization of Structures

Example for the initialization of a structure:

Poly_1 : polygonline := (Start := 3,3, Point1 := 5,2, Point2 := 7,3, Point3 := 8,5, Point4 := 5,7, End := 3,5);

Initializations with variables are not possible. See an example of the initialization of an array of a structure under Arrays.

Access on Structure Components

Use the following syntax to gain access to structure components:

<Structure_Name>.<Componentname>

For the above mentioned example of the structure polygonline, you can access the component start by Poly_1.Start.

Enumerations

An enumeration is a user-defined data type that consists of a number of string constants. We refer to these constants as enumeration values.

Enumeration values are recognized globally in all areas of the project, even if the values were declared within a POU.

From the **Insert** ribbon, select the IEC 61131-1 icon to create an enumeration as a DUT object.

Syntax:

TYPE <Identifier> : (<Enum_0>, <Enum_1>, ..., <Enum_n>) |<base data type>;

END_TYPE

A variable of type <Identifier> can take on one of the enumeration values <Enum_...> and will be initialized with the first enumeration value. These values are compatible with whole numbers, so you can perform operations with these values just as you would do with integer variables. You can assign a number $_x$to the variable. If the enumeration values are not initialized with specific values within the declaration, counting will begin with 0. When initializing, ensure that the initial values increase within the row of components. The validity of the number will be checked at the time it runs.

Example:

TYPE TRAFFIC_SIGNAL : (red, yellow, green := 10); (* The initial value for each of the colors is red 0, yellow 1, green 10 *)

```
END_TYPE
TRAFFIC_SIGNAL1 : TRAFFIC_SIGNAL;
TRAFFIC_SIGNAL1 := 0; (* The value of the traffic signal is red *)
FOR i := red TO green DO
   i := i + 1;
END_FOR;
```

Extensions to the IEC 61131-3 standard:

1. You can use the type name of enumerations to disambiguate access to an enumeration constant.

   It becomes possible, then, to use the same constant in different enumerations.

   Example:

   Definition of two enumerations:

   ```
   TYPE COLORS_1 : (red, blue);
   END_TYPE
   TYPE COLORS_2 : (green, blue, yellow);
   END_TYPE
   ```

   Use of enumeration value blue in a POU:

   Declaration:

   ```
   colorvar1 : COLORS_1;
   colorvar2 : COLORS_2;
   ```

   Implementation:

   (* possible: *)

   ```
   colorvar1 := colors_1.blue;
   colorvar2 := colors_2.blue;
   ```

   (* not possible: *)

   ```
   colorvar1 := blue;
   colorvar2 := blue;
   ```

2. The base data type of the enumeration, which per default is INT, can be specified explicitly.

   Example:

   The base data type for enumeration BigEnum should be DINT:

   ```
   TYPE BigEnum : (yellow, blue, green := 16#8000) DINT;
   END_TYPE
   ```

Arrays

ACSELERATOR RTAC supports one-, two-, and three-dimensional fields (arrays) as elementary data types. You can define arrays both in the declaration part of a POU and in the GVLs.

Syntax:

```
<Field_Name> : ARRAY [<ll1>...<ul1>,<ll2>...<ul2>] OF <elem. Type>
```

ll1, ll2, ll3 identify the lower limit of the field range; ul1, ul2, and ul3 identify the upper limit. The range values must be integers.

Example:

Card_game : ARRAY [1..13, 1..4] OF INT;

Initializing Arrays

Example for complete initialization of an array:

arr1 : ARRAY [1..5] OF INT := [1, 2, 3, 4, 5];

arr2 : ARRAY [1..2, 3..4] OF INT := [1, 3(7)]; (* short for 1, 7, 7, 7 *)

arr3 : ARRAY [1..2, 2..3, 3..4] OF INT := [2(0), 4(4), 2, 3];

(* short for 0, 0, 4, 4, 4, 4, 2, 3 *)

Example of the initialization of an array of a structure:

Structure definition:

```
TYPE STRUCT1
STRUCT
    p1 : int;
    p2 : int;
    p3 : dword;
END_STRUCT
END_TYPE
```

Array Initialization

ARRAY[1..3] OF STRUCT1 := [(p1 := 1, p2 := 10, p3 := 4723), (p1 := 2, p2 := 0, p3 := 299), (p1 := 14, p2 := 5, p3 := 112)];

Example of the partial initialization of an Array:

arr1 : ARRAY [1..10] OF INT := [1,2];

If you do not initialize the array elements explicitly, they will be initialized with the default value of the basic data type. In the previous example above, the software therefore initializes the elements arr1[6] through arr1[10] to a value of 0.

Accessing Array Components

Use the following syntax to access array components in a two-dimensional array:

<Field_Name>[Index1,Index2]

Example:

Card_game [9,2]

APPENDIX C

Firmware Upgrade Instructions

Overview

These instructions guide you through the process of upgrading firmware in the device. The firmware upgrade will be either a standard release or a point release. A standard release adds new functionality to the firmware beyond the specifications of the existing version. A point release is reserved for modifying firmware functionality to conform to the specifications of the existing version.

A standard release is identified by a change in the R-number of the device firmware identification (FID) string.

Existing firmware: SEL-2240-**R100**-V0

Standard release firmware: SEL-2240-**R101**-V0

A point release is identified by a change in the V-number of the device FID string.

Existing firmware: SEL-2240-R100-**V0**

Point release firmware: SEL-2240-R100-**V1**

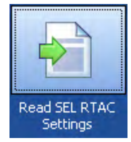
The instructions that follow explain how to read settings, upgrade the firmware, and restore the settings. These instructions also apply to upgrading the part number of the RTAC. We refer to the part number also as the model option table (MOT) number. The MOT defines the RTAC hardware configuration and also enables optional firmware applications (for example, the web-based HMI). You must use a local user account on the RTAC and not log in using the Lightweight Directory Access Protocol (LDAP) when you perform the firmware upgrade.

Settings Read

A firmware upgrade overwrites all existing SEL-2241 RTAC settings. This includes the present project, Ethernet settings, LDAP configuration, local users/passwords, certificates, and logs. The only user account retained during a firmware upgrade is the local account of the user performing the upgrade. For this reason, you must log in as a local user (not through LDAP) to perform a firmware upgrade. Use the following steps to read the settings before the upgrade so that you can restore them after the upgrade. An MOT upgrade does not overwrite existing RTAC settings, so you do not have to perform this step if you are only updating the MOT.

Step 1. Ensure that your PC is connected to the RTAC through Ethernet or, where applicable, a USB connection.

Step 2. Click the read icon to read the existing settings from the RTAC.



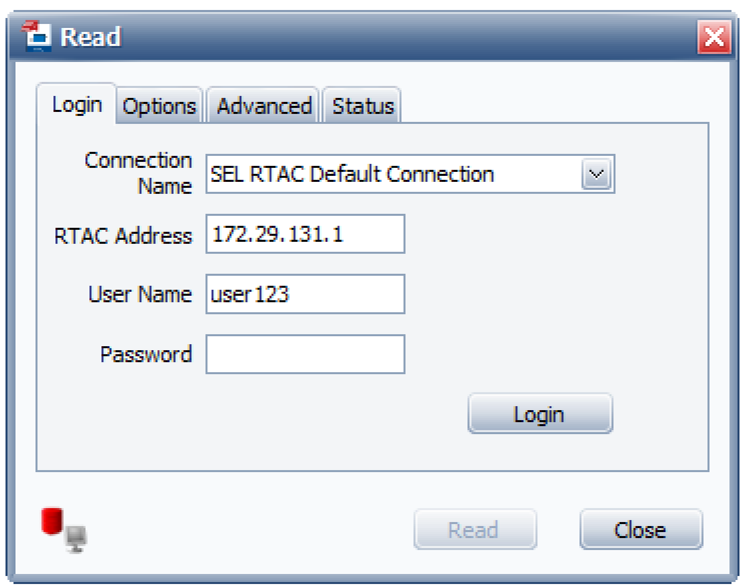
Step 3. Set **RTAC Address** to either of the following:

- the fixed IP address of 172.29.131.1 if using the front USB port

  or

- the IP address of the RTAC Ethernet port to which you are attached.

Step 4. Enter the local account login information for the RTAC and click **Login**.

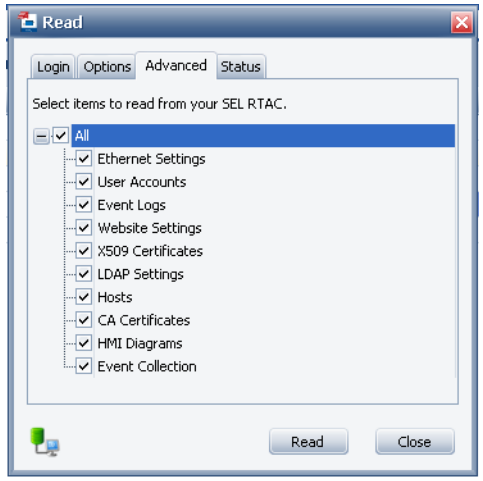


Step 5. After the status indicates connected, do the following:

a. Select the **Advanced** tab.

b. Click the check boxes for items you want to restore later. If you read the SOE and Event logs, the read and restore times will be longer.

c. Click **Read**.

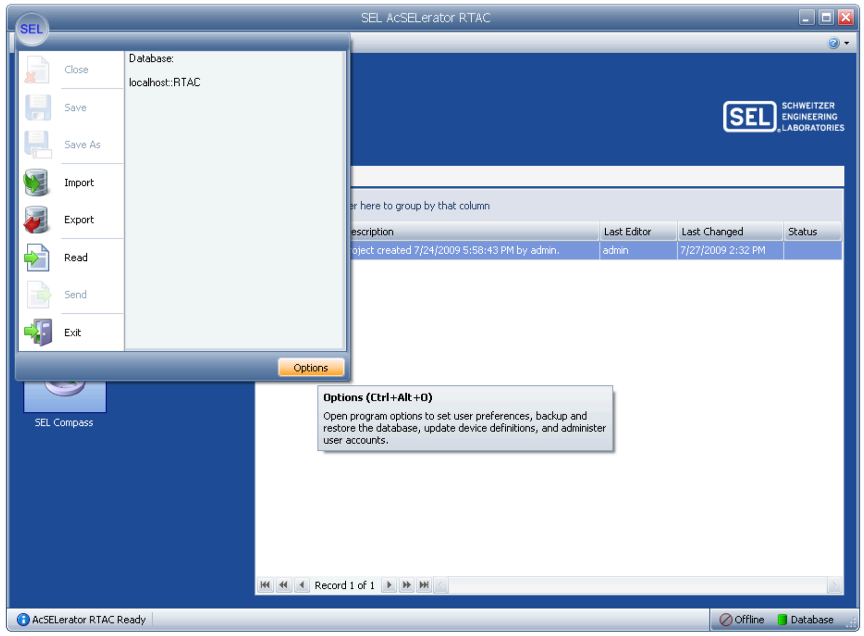


This reads the existing project from the RTAC along with all Ethernet, certificate, usernames and other RTAC settings. The downloaded settings are stored in a new project. If the project name already exists in the ACSELERATOR RTAC database, it will be stored under a new project name with a 0, 1, etc. appended to indicate the instance of that project name. You can now upgrade the firmware.

ACSELERATOR RTAC Upgrade Procedure

Step 1. With ACSELERATOR RTAC SEL-5033 Software closed, run the new setup.exe to install the updated RTAC software. You can download the latest setup.exe file from the SEL website, or use Compass to update your software.

Step 2. Open the RTAC software and click the **SEL Application** button, then **Options** to access the update function.



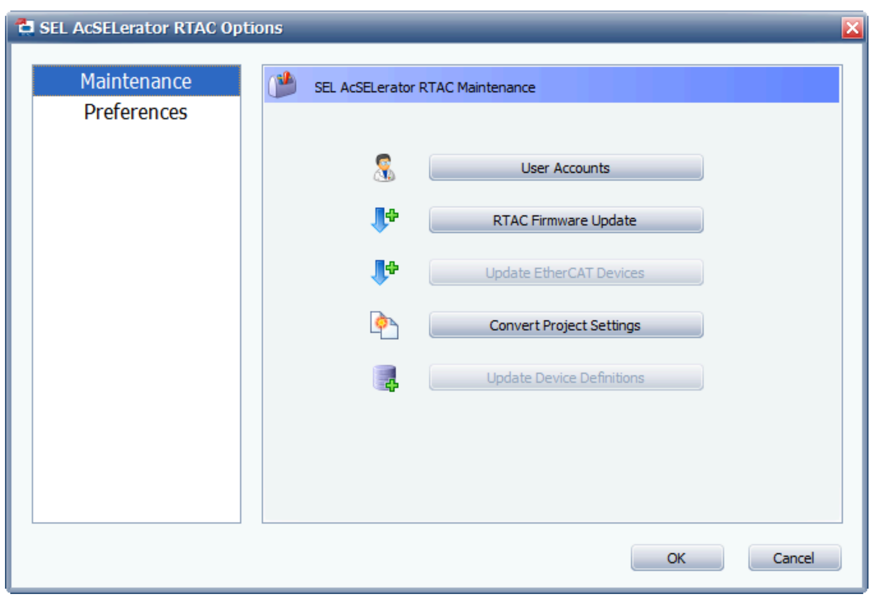
NOTE

The update function will not start if the RTAC is not connected to the computer with a communications cable or the RTAC is not powered.

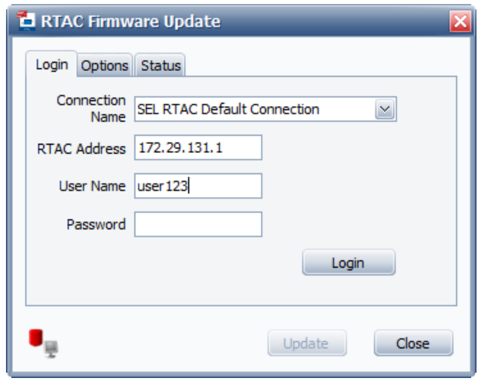
NOTE

Do not perform a firmware upgrade while the factory reset jumper is set.

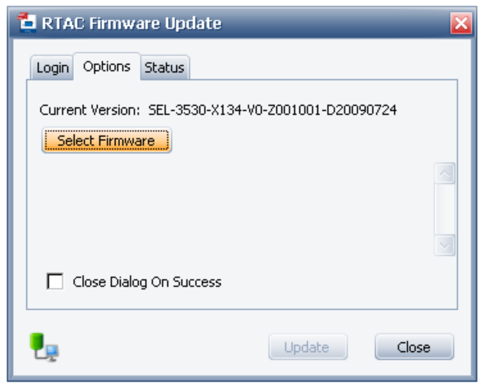
Step 3. Begin the update process by clicking **RTAC Firmware Update**.



Step 4. Enter local account **User Name** and **Password** and click **Login**.

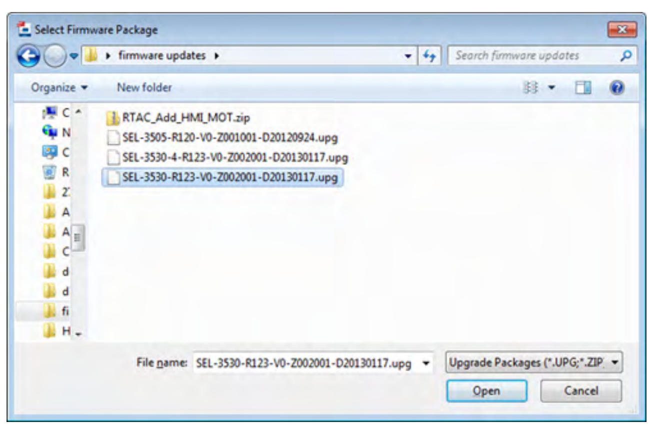


Step 5. Begin browsing for the package file by clicking **Select Firmware**.



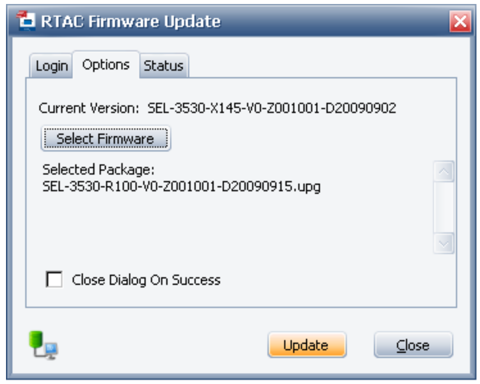
Step 6. Browse to the directory you want, click on the package (.upg or .zip) file to be loaded, and then click **Open**. Use the .upg file for a firmware upgrade and the .zip file for an MOT upgrade.

NOTE
Although SEL-3530 firmware is shown here as an example, ensure you select the correct firmware package for the product you are upgrading.

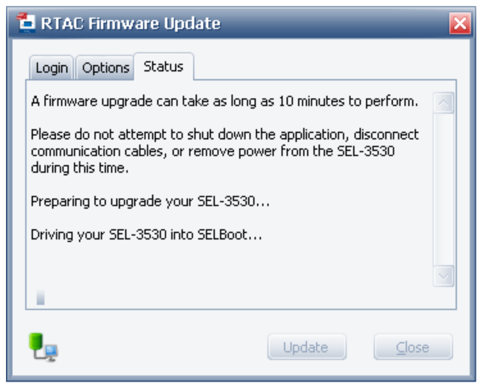


Step 7. Click **Update** to begin the loading process.

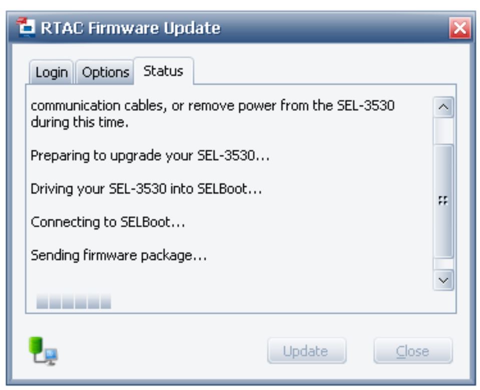
⚠CAUTION
Do not disconnect or remove and restore RTAC power during the firmware upgrade process.



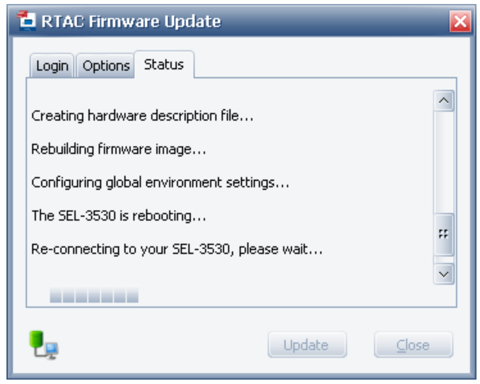
The software goes through several stages. The first stage drives the RTAC into SELBOOT mode, the **ENABLED** LED extinguishes, and the **ALARM** LED illuminates.



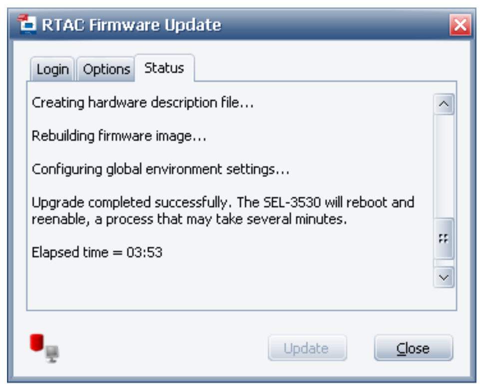
The second stage transfers the firmware package file to the RTAC.



Once the package file has been transferred, several more stages prepare the RTAC to use the new firmware or MOT (or part number). The RTAC then reboots, the **ENABLED** LED illuminates, and the **ALARM** LED extinguishes.



Step 8. Finish the update process by clicking **Close**.

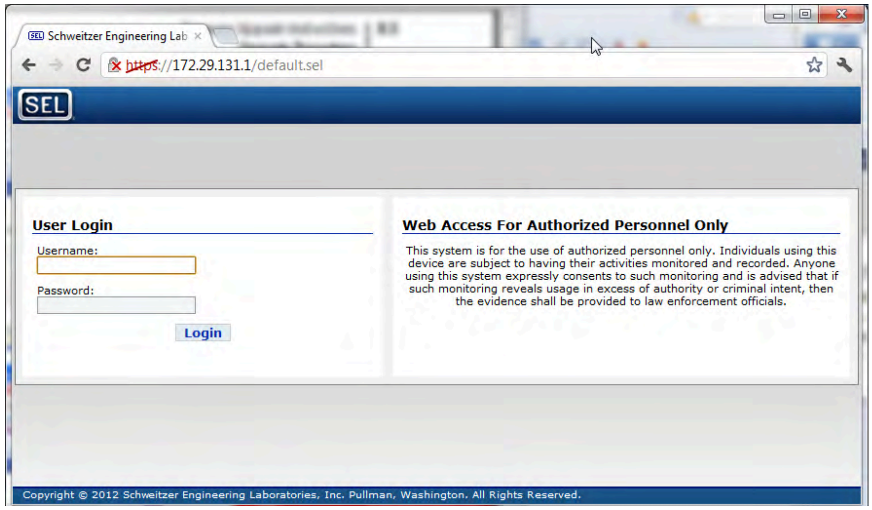


Step 9. Verify RTAC operation by first using **https** to access the RTAC homepage. In your web browser, enter the address **https://** followed by:

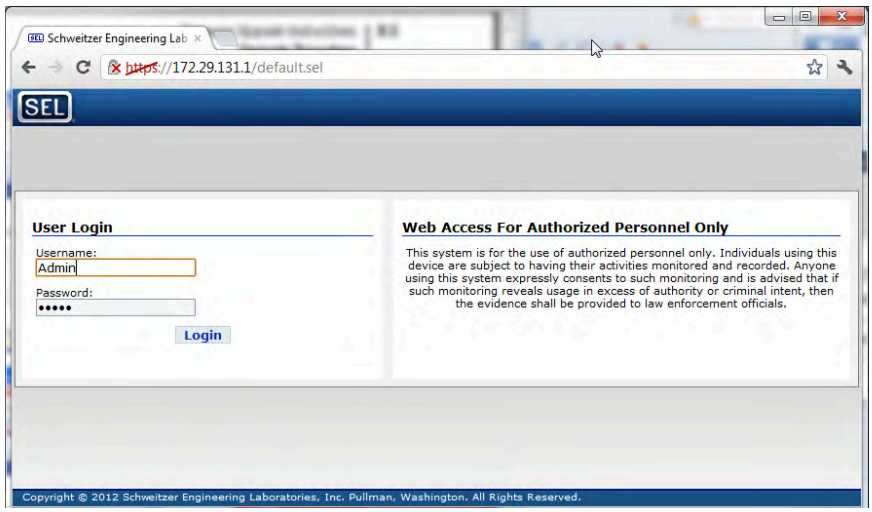
- **172.29.131.1** if using the front USB port

  or

- the IP address of the RTAC Ethernet port to which you are joined.



Step 10. Enter your local account username and password.

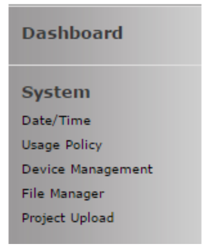


Step 11. To ensure proper web page formatting, refresh the web page by pressing **<F5>** in Internet Explorer or **<Ctrl+R>** in Chrome or Firefox.

Step 12. Verify the **Firmware Version** and **Part Number** on the RTAC **Dashboard** page.

RTAC Web Upgrade Procedure

Step 1. Navigate to the RTAC web interface and log in to the RTAC with administrator-level credentials.

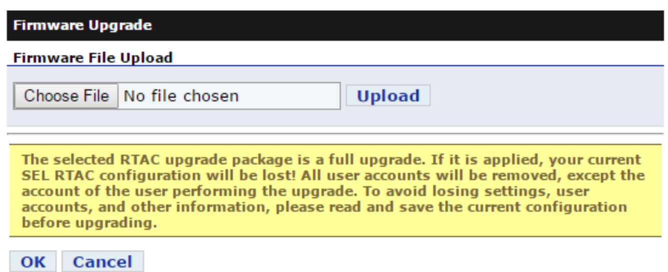


Step 2. Navigate to the **Device Management** tab.

Step 3. In the **Firmware Upgrade** section, click **Choose File** and browse to the directory you want, click on the package (.upg or .zip) file to be uploaded, and click **Open**.

Step 4. After selecting the correct file, click **Upload** and wait for confirmation that the file has been successfully uploaded.

Step 5. When the file is uploaded, a firmware confirmation appears in yellow. Click **OK**.



Step 6. Firmware upgrade commences. After completion, you will see the login screen for the web browser. The firmware upgrade is complete at this stage.

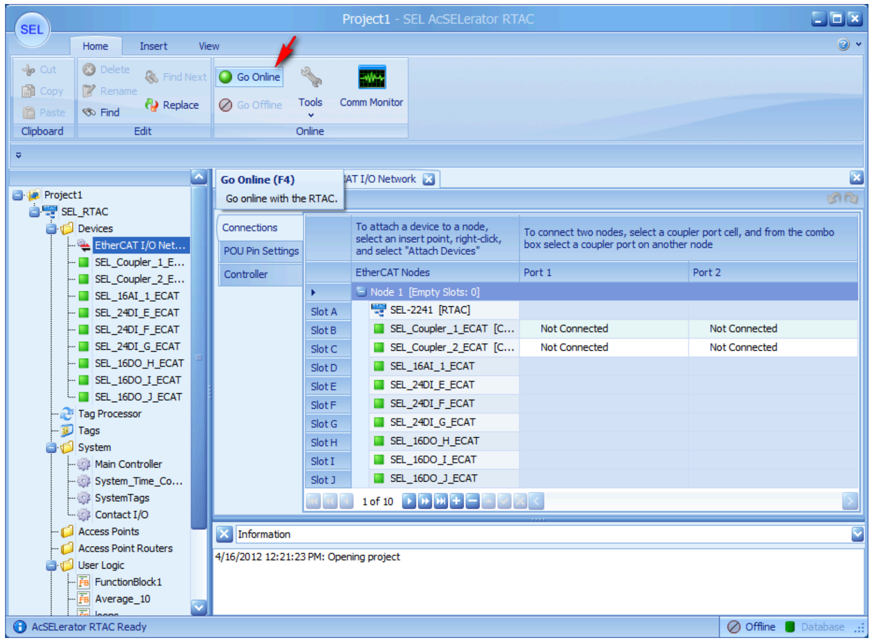
Analog Module Upgrade Procedure

From time to time, SEL issues firmware upgrades for the SEL-2245 analog modules. The instructions that follow explain how to upgrade the firmware. The RTAC project contains the device settings. No additional steps are necessary to back up or restore settings. You must use a local user account on the RTAC and not log using LDAP when you perform the firmware upgrade.

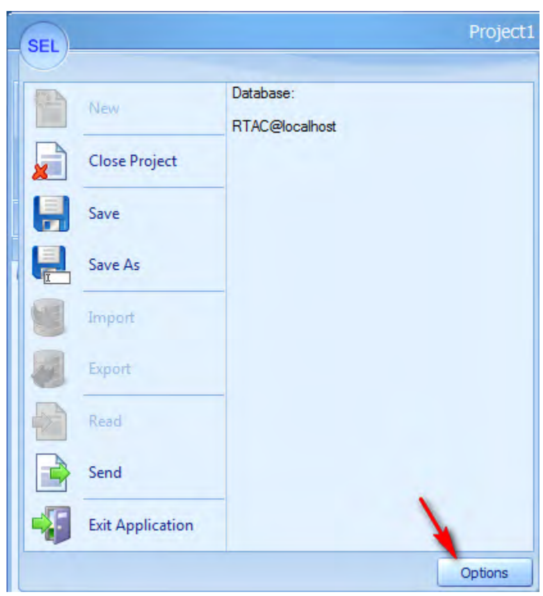
NOTE
Downgrading analog module firmware is supported in RTAC firmware R125 and higher.

Step 1. Open an existing ACSELERATOR RTAC project or create a new one. If you are creating a new project, you will have to configure the EtherCAT Network properly first by adding all modules.

Step 2. Select **Go Online** and wait for the project to deploy.

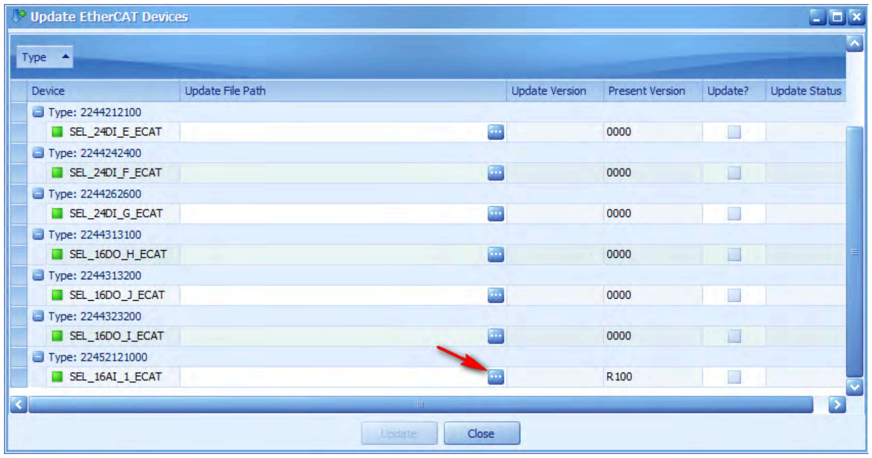


Step 3. Select the **SEL Application** button and then **Options** to access the update function.

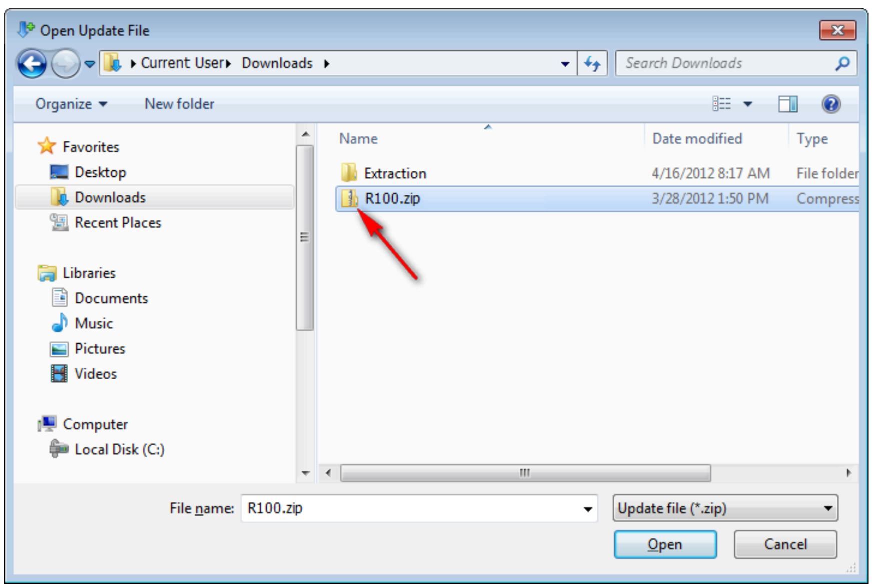


Step 4. Begin the update process by selecting **Update EtherCAT Devices**.

Step 5. Click the **Open File** button to the right of the module being upgraded.



Step 6. Browse to the location of the upgrade file. Select it and click **Open**.



Step 7. Select **Update**.

NOTE

Disabling the system is necessary to update module firmware. Ensure the system is in a safe state prior to upgrading.

Step 8. Wait while the upgrade completes. The green **ENABLED** LED will blink slowly and the **ALARM** LED will illuminate while the update is in progress.

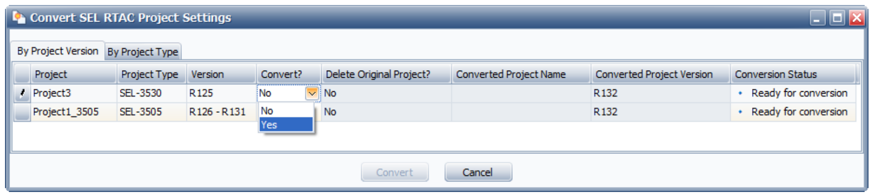
Step 9. After the update finishes, a green check mark will appear in the Update Status column and the **ENABLED** LED will illuminate. Close the **Update EtherCAT Devices** window and click **OK** on the Options window to return to your project.

Settings Restore

After completing the firmware upgrade, you can restore your RTAC to its previous configured settings. *Do not perform this step if you only performed an MOT update*.

Step 1. Click **SEL Application > Options > Convert Project Settings** to convert the project settings for compatibility with the newly updated firmware.

Step 2. The software lists all projects that were created with previous software versions. For each project you want to convert, change the **Convert?** field from **No** to **Yes**.

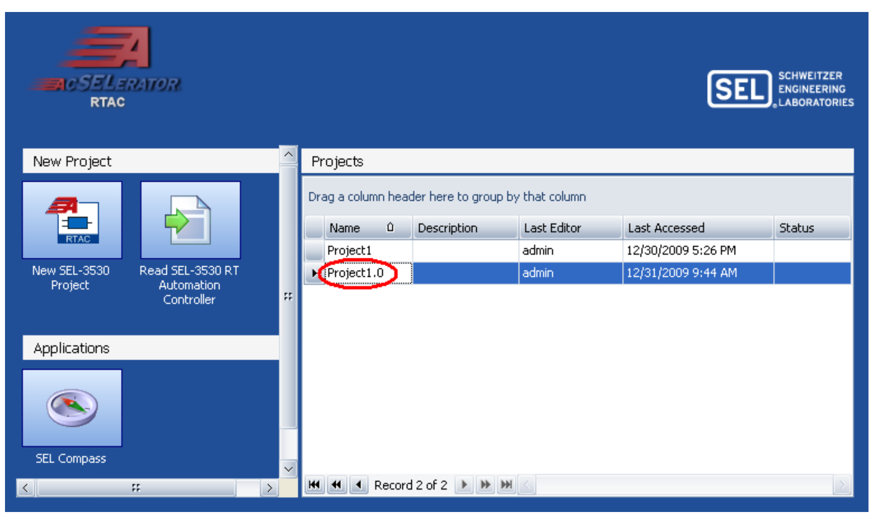


Step 3. Click the **Convert** button. This takes several minutes because the software makes a copy of each selected project, converts the settings for compatibility with the new firmware, and saves the new project settings.

NOTE

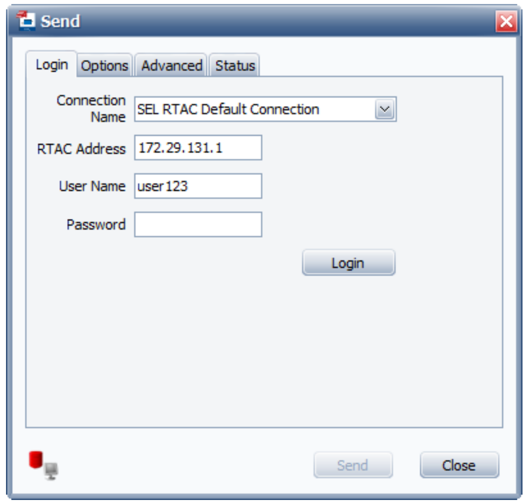
You can also convert a project to work on a different RTAC model by clicking the **By Project Type** tab.

Step 4. Open the project you read from the RTAC in *Settings Read on page 773*. You can do this from the RTAC start page by double-clicking on the project name.



Step 5. Click the send icon to send the project to the RTAC.

Step 6. Enter the local account login information for the RTAC and click **Login**.

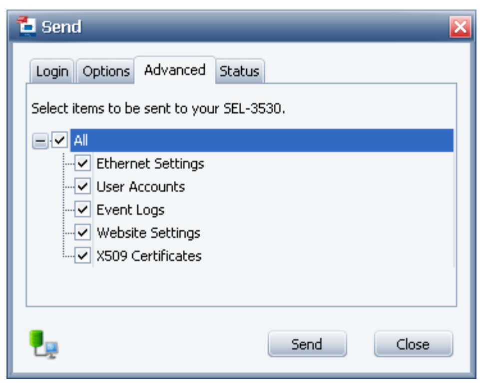


Step 7. After the status indicates connected, do the following:

a. Select the **Advanced** tab.

b. Click the **All** check box.

c. Click **Send**.



The previous project and settings are restored to the RTAC.

Technical Support

We appreciate your interest in SEL products and services. If you have questions or comments, please contact us at:

Schweitzer Engineering Laboratories, Inc.
2350 NE Hopkins Court
Pullman, WA 99163-5603 U.S.A.
Tel: +1.509.338.3838
Fax: +1.509.332.7990
Internet: selinc.com/support
Email: info@selinc.com

This page intentionally left blank

APPENDIX D

Configure Dynamic Disturbance and Fault Recording Systems

Overview

The SEL-3555 RTAC with SEL-2240 Axion modules is designed to implement powerful recording solutions that exceed all of the requirements of PRC-002. The current draft standard of PRC-002 requires a minimum of 10 days storage of 16 samples per cycle for event reports, 30 samples per second of dynamic disturbance records (DDR), and time-stamp accuracy of digital values within 4 ms. The SEL-3555 RTAC with larger solid-state drives (SSDs) has more than enough storage to accommodate the higher-resolution measurements from SEL Axion modules for this 10-day period. SEL Axion modules are capable of recording at 24 kHz for event reports, DDR recording at 60 samples per second, and SOE accuracy of 1 ms. This section provides a high-level outline of the configuration steps required for creating Dynamic Disturbance and Fault Recording Systems with the SEL-3555 RTAC and SEL Axion input/output (I/O) modules.

Dynamic Disturbance and Fault Recording Systems are intended for applications using an SEL-3555 RTAC. Recording Group configuration is supported on other RTACs that support EtherCAT, however, storage is limited and the processing power is capable of combining fewer modules. *Table D.1* shows expected performance for RTACs that support EtherCAT.

NOTE
Recording Groups configured at 24 kHz for 24 seconds on the SEL-2241, SEL-3530, or SEL-3530-4 with more than two SEL-2245-42 Modules will time out before they are complete. An SEL-3555 RTAC should be used for any systems larger than two Modules at 24 kHz for 24 seconds.

NOTE
The configured RTAC task interval will significantly affect how long recording groups take to combine events. Table D.1 has a configured RTAC task interval of 30 ms.

Table D.1 Dynamic Disturbance and Recording System Performance

Number of SEL-2245-42 Modules	Sample Rate (kHz)	Length of Event (seconds)	Length of Time to Combine (minutes)
SEL-3555 RTAC			
16	24	24	28
16	1	576	36
16	24	4	5
16	8	4	3

Number of SEL-2245-42 Modules	Sample Rate (kHz)	Length of Event (seconds)	Length of Time to Combine (minutes)
SEL-2241, SEL-3530, or SEL-3530-4 RTAC			
3	24	5	19
3	1	120	23
4	24	5	21
4	1	120	24
7	8	4	13
7	24	4	30

System Components

- SEL-3555 RTAC (with the MOT option for the DDR Library)
- SEL RTAC Protection Library
- SEL-2242 Backplane(s)
- SEL-2243 Power Coupler(s)
- SEL-2244-2 Digital Input Module(s)
- SEL-2244-3 Digital Output Module(s) (Optional)
- SEL-2244-5 Fast High-Current Digital Output Module(s) (Optional)
- SEL-2245-42 AC Protection Module(s)

Figure D.1 Dynamic Disturbance and Fault Recording System Components

SEL-3555 RTAC and Axion Configuration

Starting with a new SEL-3555 RTAC project, add the SEL Axion components you have selected for your PRC-002 recording solution. Refer to the SEL-3555 RTAC Manual, SEL-2240 Manual, and *EtherCAT on page 151* for more details on the individual configuration options of the RTAC and Axion components.

Configure a Recording Group

Refer to *EtherCAT on page 151* for details on configuring single or multiple recording groups. Recording Group configuration is essential to consolidate multiple event reports if you have more than one SEL-2245-42 AC Protection module in your Axion and include inputs from the SEL-2244-2 Digital Input modules in the same COMTRADE file. This simplifies fault analysis so that all information pertaining to a fault is in the same file. Additionally, custom digital tags can be mapped from the RTAC to include in the resulting COMTRADE file if desired. Recording group settings override the individual SEL-2245-42

AC Protection module oscillography settings so that all oscillography data in the recording group is recorded at the same rate. Be sure to enable SOE tags on the SEL-2244 Digital Input modules so that state changes that occur faster than the RTAC processing interval will be recorded. After you configure a recording group, assert Recording-GroupX.Event_Trigger to control the individual event report triggers of all SEL-2245-42 AC Protection modules included in the recording group.

Configure Event Triggers for Recording Groups

Event_Trigger for each recording group is a binary trigger that begins the recording of a new COMTRADE event report. As this is simply a binary trigger, you must configure the logic to determine when an event report should be recorded. Common inputs that determine when recording is needed are breaker operation (digital input), overcurrent, over- or undervoltage, or other abnormal conditions. Download the Recording Triggers Extension from the SEL website for basic pre-built protection elements to detect these abnormal conditions. The Recording Triggers Extension includes instructions to configure each protection element for proper use. Add as many Recording Triggers as necessary in the Extension to configure triggers for each Recording Group.

Configure Dynamic Disturbance Records

The optional DDR library allows you to record time-aligned synchrophasor data from SEL-2245-42 AC Protection modules on the SEL-3555 SSD drive for DDR archiving. Any of the analog tags of the SEL-2245-42 AC Protection modules can be recorded using the DDR library, but the synchrophasor tags are best suited because they are precisely time-aligned values at up to 60 messages per second. Refer to the instructions included in the DDR library download for more details on how to configure DDR recording.

Configure SOE Recording

Configure SOE records for the SEL-2244-2 Digital Input modules by enabling the SOE tags on each SEL-2244-2 module and enabling logging for those SOE tags in the tag processor. Enabling these SOE tags is necessary for 1 ms accurate time-stamping of the digital inputs. Without the SOE tags, only the status tags, which update at the RTAC processing interval, will be available. Refer to *EtherCAT on page 151* for more details on configuring SOE recording on SEL-2244-2 Digital Input modules.

Data Collection

Multiple options are available for collecting the SOE, DDR, and Fault Record data outlined in this section. ACSELERATOR TEAM SEL-5045 Software is one option to set up automatic collection of these data. The FTP server in the RTAC is another option for remote collection of these data. Additionally these data can be manually collected from the RTAC web interface. Refer to the corresponding sections documentation of each method for more details on configuration. After collection, use SEL-5601-2 SYNCHROWAVE Event Software to analyze the COMTRADE data.

APPENDIX E

EtherCAT Overview

Evolution of Fieldbus I/O Networks

NOTE
This section is a brief introduction to EtherCAT®. For more detailed information, visit ethercat.org.

More than 30 years ago, process control systems began transitioning from hard-wired relay logic and single-function loop controllers to programmable logic controller (PLC) based systems. The limited computing capability of early PLCs supported a relatively small number of field inputs and outputs. Because of that limitation, I/O wiring could be economically terminated within the same cabinet as the PLC. However, as automation vendors developed more advanced PLCs that accommodated many more field points—and a larger variety of signal types—users needed an alternative I/O networking method to avoid terminating hundreds or thousands of points directly in the controller cabinet.

In response, equipment vendors developed I/O devices that could communicate with a PLC via a specialized network, or fieldbus. Fieldbus networks were based on EIA-232 or EIA-485 communications and provided a means for a controller to read and write large quantities of I/O points via a single communications cable. Early fieldbus implementations were based on proprietary network protocols. In the 1990s, however, a number of standard protocols, such as FOUNDATION™ fieldbus, became popular in the industry.

Development of EtherCAT

The serial networking that process control systems used for fieldbuses in the 1990s became bandwidth limited as controllers with larger and larger processing capabilities entered the market. Use of Ethernet technology as a second generation fieldbus was a natural evolution, but the inherently nondeterministic nature of Ethernet created difficult new problems. Additionally, most Ethernet equipment was not designed for industrial use, which made the adoption of Ethernet-based fieldbuses even slower.

There exist a number of Ethernet protocol standards that operate under constraints intended to make them more deterministic. Some standards use messaging rules to eliminate collisions and reduce signal jitter. Others take advantage of Ethernet switches that use virtual local area networks (VLANs) to reduce network burden within a subnet to only messages sent from or meant for devices within that subnet.

While these methods improve network determinism, they still do not use bandwidth efficiently. Most of these protocols adhere to an Ethernet paradigm that demands that each device send an entire Ethernet frame for each message and that every message delivered to a device consist of an entire Ethernet frame.

Each frame can be as long as 1,500 bytes, even if the usable process information (either inputs or control outputs) only requires a few bytes. The result, even when using multicast messages, is that administrative information consumes a large amount of the network traffic.

EtherCAT Technology

The developers of EtherCAT® created solutions for both the time and efficiency challenges of Ethernet. The fundamental difference between EtherCAT and other Ethernet fieldbus protocols is that a single EtherCAT frame contains I/O point updates from many devices in a network, not just a single device. Existing transport protocols could not accomplish this, so developers defined a new EtherType explicitly for the EtherCAT protocol. As we will see, this approach provides complete compatibility with Ethernet standards.

Development Objectives and Requirements

Although it is an open standard protocol today, EtherCAT began as the invention of Beckhoff Automation. The main development objectives were as follows:

- Deterministic network operation
- Fast network update time
- Efficient use of network bandwidth
- Compatibility with existing controller Ethernet hardware
- Full conformity with the Ethernet standard
- Economical implementation for small and large I/O devices

Definition of EtherType for EtherCAT Messages

Developers designed the EtherCAT frame (shown in *Figure E.1*) specifically to incorporate process data from many Ethernet nodes into a single message. The telegram can extend to multiple frames and accommodate a maximum size of 4 gigabytes. The developers configured individual devices in the network to read and write data from specific regions of the telegram, which means that the telegram mapping sequence has no direct relation to the physical network configuration.

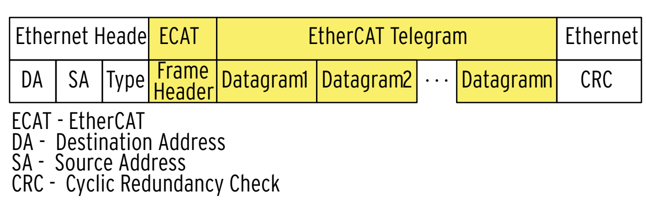


Figure E.1 Standard Ethernet Frame for EtherCAT Messages

As shown in *Figure E.2*, the independent data mapping allows designers to create telegrams based on specific process sequences or mapping preferences in the PLC or other Level 1 controllers.

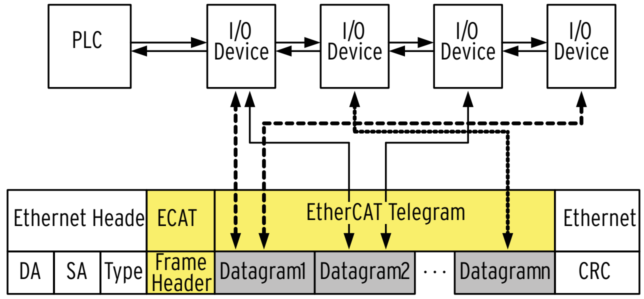


Figure E.2 Standard Ethernet Frame for EtherCAT Messages

How On-the-Fly Processing Works

To achieve the network speed necessary for critical applications, EtherCAT devices use a low-level, on-the-fly processing method in which all devices within a network segment receive the entire EtherCAT message. The PLC, or EtherCAT master, begins the sequence by sending a message with updated output control data and input status from the previous data cycle.

As the first I/O device in the network starts receiving the frame, it automatically reads the proper control data from the telegram and writes updated process data into the telegram. The first device also automatically forwards the updated telegram to the next I/O device with a delay of only 1 to 2 ns. Each subsequent device similarly reads and writes portions of the telegram; the last device returns the completed message to the PLC. Even in very large systems, completion of the entire round trip can occur in less than 100 µs.

The hardware interface for each device consists of standard Ethernet ports. However, each I/O device has a field-programmable gate array (FPGA) or low-cost application-specific integrated circuit (ASIC) that reads and writes the EtherCAT telegram to maintain the incredibly short signal delay. Automatic configuration of a fieldbus memory management unit (FMMU) process in the FPGA or ASIC occurs upon network initialization to establish relevant input and output locations in the EtherCAT telegram. Once the network enters normal operating mode and begins transmitting telegrams, it wastes no time in evaluating an entire telegram. The FPGA or ASIC can quickly read and write just the necessary memory locations and forward the telegram. After sending a telegram, the device acts internally on new control commands it receives and updates inputs before the next telegram arrives.

The PLC, or in this case the RTAC module, maintains overall control system determinism by starting each telegram transmission on a fixed schedule. Low processor jitter in modern control hardware enables a repeatable schedule.

This page intentionally left blank

SCHWEITZER ENGINEERING LABORATORIES, INC.
2350 NE Hopkins Court • Pullman, WA 99163-5603 U.S.A.
Phone: +1.509.332.1890 • Fax: +1.509.332.7990
selinc.com • info@selinc.com