PowerLogic™ SCADA

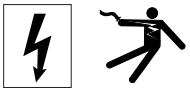
System integrator's manual
Version 7.10, r01 Service release 3

63220-100-200D1
05/2011

Schneider Electric

HAZARD CATEGORIES AND SPECIAL SYMBOLS

Read these instructions carefully and look at the equipment to become familiar with the device before trying to install, operate, service or maintain it. The following special messages may appear throughout this bulletin or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of either symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result** in death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result** in death or serious injury.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result** in minor or moderate injury.

CAUTION

CAUTION, used without the safety alert symbol, indicates a potentially hazardous situation which, if not avoided, **can result** in property damage.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

Contents

Contents 1
Introduction 7
PowerLogic SCADA—A Complete Solution 8
Assumptions 8
What is SCADA? 8
What is PowerLogic SCADA? 8
Preparing for PowerLogic SCADA 8
Components of a Project 9
Configuration Tools 9
Runtime Environment 9
Supported Device Type Drivers 9
System Requirements and Installation Instructions 11
Uninstalling and Reinstalling PowerLogic SCADA 11
Quick Start: Typical Workflow 12
Getting More Information 15
Manuals 15
Help files 15
Customer support 15
The Profile Editor 16
Overview of the Profile Editor 17
Typical Workflow Illustration 18
Workflow Overview 18
Create/Edit Device Type 19
Create/Edit Device Profile 20
Create/Edit Unit Templates 21
Launching the Profile Editor 22
Locked and Custom Icons 22
Setting the Screen Resolution 22
Adding Engineering Unit Templates, Units, and Conversions 23
Setting Up Engineering Templates and Selecting Conversions 23
Creating or Editing an Engineering Unit Template 24
Selecting Conversions 24
Deleting a Template 25
Adding or Editing a Base Engineering Unit or Conversion 25
Adding a Base Engineering Unit and Unit Conversion 26
Editing a Base Engineering Unit or Conversion 26
Deleting a Base Engineering Unit or Conversion 26
Defining Device Types and Tags 27
Device Type Screens and Workflow 27
Using the Define Device Type Tags Tab 27
Adding, Editing, or Deleting a Device Type 30
Adding a Device Type 31
Editing a Device Type 32
Deleting a Device Type 33
Assigning Tags to Generic I/O Points 33
Printing the .CSV File 34
Setting Up Device Type Categories 34
Adding a Category or Subcategory 35
Editing a Category/Subcategory Name 35
Deleting a Category or Subcategory 35
Adding Custom Tags and Tag Addresses 36
Setting Up Custom Tags 36
Adding a Custom Tag 38
Editing a Custom Tag 39
Deleting a Custom Tag 39
Editing Tag Addresses 40
Real-Time Tag Addresses 40
Onboard Alarm Tag Addresses 41

Reset Tag Addresses 41
Control Tag Addresses 42
Editing Address Information 43
Adding a New Tag Address 43
Creating Device Profiles 44
Enabling Waveforms 44
Viewing Device Profiles 44
Adding, Editing, or Deleting a Device Profile 45
Adding a Device Profile 46
Editing a Device Profile 47
Deleting a Device Profile 47
Setting Up Projects in the Profile Editor 49
Project Screens and Workflow 49
Typical Workflow 49
Viewing Project Data and Exporting the Project 50
Adding, Editing, or Deleting a Project 50
Adding a Project 50
Editing a Project 51
Deleting a Project 51
Exporting a Project 51
Before you export 52
Exporting 52
Moving Files if the Profile Editor is not with the Server (Reusing a Project) 53
Editing and Deleting Information in a PowerLogic SCADA Project 53
Customizing Tag Names 53
Understanding Tags 54
IEC61850 Tag Construction 54
Defining an Enumeration 55
Using Special Tags to Control Circuit Breaker Status 55
Format Code Definitions 56
Real-Time Format Code Definitions 57
Alarm Format Code Definitions 58
Control Format Code Definitions Rules of Operation 58
Predefined Control Format Codes 58
Predefined Reset Format Codes 58
Custom Control and Reset Format Codes 59
Logic Codes 59
Using Block Writes 65

The Profile Wizard 67
Before You Use the Profile Wizard 68
Adding the Cluster, Network Address, and Servers 68
When Setting Up More than two I/O Servers per Cluster 70
Entering Copy for Translation 71
Using the Wizard to Add and Remove Devices 72
Adding a Device to a PowerLogic SCADA Project 72
Removing a Device From the PowerLogic SCADA Project 73
Editing Devices in a PowerLogic SCADA Project 74
Editing a Profile and Adding it Back to the Project 74
Editing a Device in PowerLogic SCADA Only 74
Adding Device Data in PowerLogic SCADA Only 74
Adding Multiple Devices through the Automation Interface 74
Using the OLE Automation Interface to Add and Remove Devices 77
Examples 77
C# 77
Excel VBA 78
Command Line Interface 78
Interface Workflow 78
Prerequisite information 79
Interface 80

Graphics: Using the Design Time Environment ... **82**
Typical Workflow ... 83
The PowerLogic SCADA Pages ... 84
Creating a PowerLogic SCADA Project ... 85
Reusing Projects Created in the Profile Editor ... 85
Before You Begin ... 85
Adding a New Graphics Page ... 86
Setting a new page as the project's startup page ... 86
Creating a One-Line on a Graphics Page ... 86
Using Menu.dbf to Change the Graphics Page Appearance ... 89
Creating and Editing Genies ... 90
To create a custom genie, follow these steps: ... 90
To create a custom symbol for a custom genie, follow these steps: ... 91
Deleting Information from PowerLogic SCADA ... 91
Working with Alarms ... 92
Alarms Overview ... 92
PC-Based Alarms ... 92
Onboard Alarms ... 92
Adding Setpoints and Delays ... 92
Making a PC-Based Alarm Digital ... 92
Setting Up an Alarm Based on an Enumeration ... 93
Changing an Alarm Severity ... 93
Enabling Waveforms for Onboard Alarms ... 93
Setting Parameters for Event Log Length and Historical Logging of Events ... 94
Event Storage: [Alarm]SummaryLength Parameter ... 94
Event Logging: [PLSEventLog] Parameter ... 94
Adding an Onboard Alarm Tag ... 94
Setting Up Audible Alarms ... 95
Restoring and Backing Up Projects ... 96
Restoring a Project ... 96
Backing Up a Project ... 96
The Project Color Change Tool ... 97
Adding a Custom Genie for Use in the Color Change Tool ... 97
Changing the Colours of Busbars and Circuit Breakers ... 97
Using Cicode to Customize a Project ... 99
Customizing PowerLogic SCADA Default Behaviors ... 100
Time Zone Settings ... 100
Time Synchronization ... 100
Graphics Library Parameters ... 101
Maximum number of entries that can be held in Event Log ... 101
Parameters for Alarm and Event States ... 101
Trend Tag Scan Intervals ... 102
Calculating Disk Storage for Trends ... 103
Waveform Management ... **104**
Waveform Storage ... 105
Waveform to Alarm Linkage and Special Waveform Tags ... 106
Graphics: Using the Runtime Environment ... **108**
Launching and Closing the Runtime Environment ... 109
Viewing the Graphics Page in the Runtime Environment ... 110
Logging On to the Runtime Environment ... 110
Viewing the Interface ... 110
Viewing the Alarms/Events Page ... 110
Viewing, Printing, and Export Options ... 111
Event/Alarm Log Columns Table ... 111
Changing the Display Name in an Alarm or Event Log ... 112
The Event Log ... 113
The Alarm Log ... 113
Unacknowledged Alarms and Disabled Alarms ... 113
Alarm and Events Logging ... 114
Filtering and Freezing Log Views ... 114

Filtering Information 114
Freezing the View 115
Changing the Display Name in an Alarm or Event Log 115
The Event Log 115
The Alarm Log 115
Unacknowledged Alarms and Disabled Alarms 116
Alarm and Events Logging 116
Using the Analysis Page 116
Using the Genie Status Page 117
Viewing Waveforms 119
Entering Setpoints for Alarms 120
Viewing Real-Time Trends 121
Viewing Lists of Real-Time Information for the Genie 121
Multi-Monitor Support 122
Using the Graphics Test Mode 123
Using the Demo Mode 124
Assigning and Controlling User Privileges 125
Default User Access Settings 126
Adding and Modifying User Accounts 127
Adding Windows Integrated Users 127
Changing Access Rights 128
Appendix 1: Default Genie Library 129
PLS_ALARM 129
PLS_ANSI_BASE_1 / PLS_ANSI_BASE_2 129
PLS_ANSI_CB_1 / PLS_ANSI_CB_2 130
PLS_ANSI_SW_1 / PLS_ANSI_SW_2 132
PLS_DISPLAY 132
PLS_GEN_BASE_1 / PLS_GEN_BASE_2 132
PLS_GEN_CMD_1 / PLS_GEN_CMD_2 133
PLS_GEN_EQ_1 / PLS_GEN_EQ_2 133
PLS_IEC_BASE_1 / PLS_IEC_BASE_2 134
PLS_IEC_CB_1 / PLS_IEC_CB_2 135
PLS_IEC_SW_1 / PLS_IEC_SW_2 135
PLS_METER 135
ITEM1 135
Appendix 2: Using Citect INI Parameters 137
Parameters Database 137
General PowerLogic Driver Parameters 139
watchtime 139
kernelStatisticUpdateRate 139
UseWriteMultiRegistersOnly 139
timeout 140
retry 140
RetryTimeout 140
RetryException 141
standbyRefreshRate 141
standbyCheckTime 141
statusUnitCheckTime 142
initUnitCheckTime 142
initCacheTimeout 142
cacheRefreshTime 143
StatusRegister 143
StatusRegisterType 143
ModbusBase 144
RegMode 144
timeZone 145
Performance Tuning Parameters 146
Bandwidth Allocation Parameters 146
Packet Blocking Optimisation Parameters 147
Advanced Tag Block Capabilities (Invalid Memory Access Blocks defined) 149

Invalid Block Tag Definition Syntax 150
Configuration Notes 150
Write-only Memory 151
Tag Blocking Notes 152
Write-only Tags 152
Waveform Parameters 154
WaveformsDisable 154
WaveformMatchMargin 154
WaveformCheckTime (PM, CM, and Sepam) 154
WaveformZone (Sepam) 155
Sepam Event Reading Parameters 156
EventTable 156
EventIdle 156
MicroLogic Modules Configuration Parameters 157
CCM 157
Module-Specific Packet Blocking Optimisation Settings 157
Data Replication Parameters 158
FileReplicationDst 158
FileReplicationSrc 158
Graphics Library Parameters 159
Maximum number of entries that can be held in Event Log 159
Parameters for Alarm and Event States 159
Appendix 3: Glossary 161
alarm categorization 161
alarm text (On/Off) 161
alarm filters 161
alarm groups 161
alarm levels 161
alarm types 161
Automation Interface 161
bandwidth 161
baud rate 161
bitmask 161
Cicode 162
cluster 162
Color Change Tool 162
CommsMethod (communications method) 162
ComPort 162
control 162
custom device type 162
custom tag 162
DataBits 163
demo mode 163
design time environment 163
device category 163
device type drivers 163
device profile 163
device type 163
engineering unit templates 163
enumeration (used for the circuit breaker status) 163
format code 164
genie 164
graphics test mode 164
IEC tag name 164
logic code 164
metadata 164
multi-monitor support 164
onboard alarm 164
parity 164
PC-based alarms 165

PMCU ... 165
point (see SCADA tag) ... 165
polling priority ... 165
PowerLogic SCADA tag name library ... 165
Profile Editor ... 165
Profile Wizard ... 165
Project Color Change Tool ... 165
register scaling ... 165
resets ... 165
runtime environment ... 166
SCADA tag (SCADA point) ... 166
StopBits ... 166
tag address ... 166
tag group ... 166
user privileges (user access, user rights) ... 166
vector math ... 166
voltage levels ... 166

Appendix 4: Frequently Asked Questions (FAQs) ... 167
How do multiple engineers create device types and profiles at the same time? ... 167
If I don't use PowerLogic drivers, how do I create device profiles? ... 167
Can I install ION E or SMS on the same computer as PowerLogic SCADA? ... 167
When creating custom tags, how do we ensure consistency in their names and attributes? ... 168
How should we manage categories and subcategories? ... 168
How should we manage units and unit templates? ... 168
When should I create a device type rather than device profile? ... 168
How do we share or transfer device types and device profiles between computers? ... 169
How can we re-use previous Profile Editor projects? ... 169
How do we synchronize a new PC with the master Profile Editor PC? ... 170
What do I do before I add or remove devices in the Profile Wizard? ... 170
What are the requirements for device names? ... 170
How do I use MODBUS communications methods? ... 170
How can I add more than one device at a time? ... 171
In PowerLogic SCADA, what are the naming conventions for servers and clusters? ... 171
How and when do I create users for the Runtime environment? ... 171
How do I manage projects in the Citect Explorer environment of PowerLogic SCADA? ... 172
On the Graphics page, what do I need to know about creating genies? ... 173
How do we customize existing templates? ... 173
How do I change the default pickup/dropout text for alarms? ... 174
What can I modify during runtime? ... 175
Why do the browser navigation buttons not work? ... 175
What can I set up in logging and archiving? ... 175
On the Graphics page, how are colour changes displayed for voltage levels? ... 175
How do I create and configure busbars? ... 176
What INI parameters should I use for debugging? ... 177
How do I tune my system for best performance? ... 178
If a tag is configured in PowerLogic SCADA, how is it polled in the device? ... 179
Device popup from a one-line: Why do the fields overlap? ... 179
Can I change the %CLUSTER% name in the Profile Wizard? ... 180
When viewing PowerLogic SCADA from a web client, I see a licensing error. How do I fix it? ... 180
A device can prevent writes to its registers: how do I ensure that writes are successful? ... 180
How do I prevent PowerLogic SCADA from accidentally making invalid areas in memory available to reads and writes? ... 181
How do I create an audit in the Event Log for user logins and logouts? ... 181
Why do I get an "out of memory" message when I try to run the Automation Interface? ... 182
Why am I seeing #COM for circuit breaker status in the genie status page? ... 182

Index of Terms ... 183

Introduction

Welcome to the help file for PowerLogic™ SCADA version 7.10. To navigate through the file, you can use the search options on the left. Additionally, you can access the PowerLogic SCADA help files for PowerLogic SCADA-specific information.

In this section, you will find these topics:

PowerLogic SCADA—A Complete Solution
- *Assumptions*
- *What is SCADA?*
- *What is PowerLogic SCADA?*

Preparing for PowerLogic SCADA

Components of a Project
- *Configuration Tools*
- *Runtime Environment*

Supported Device Type Drivers

System Requirements and Installation Instructions

Uninstalling and Reinstalling PowerLogic SCADA

Quick Start: Typical Workflow

Getting More Information
- *Manuals*
- *Help files*
- *Customer support*

PowerLogic SCADA—A Complete Solution

The PowerLogic SCADA system provides a complete solution for the monitoring and control of any electrical distribution network. Using PowerLogic SCADA as the engine, PowerLogic SCADA includes additional tools that users will use to acquire data from various devices, and then to apply the data for HMI viewing, alarming, trending, and reporting.

In this section, you will find these topics:

Assumptions

What is SCADA?

What is PowerLogic SCADA?

Assumptions

The person who installs PowerLogic SCADA will have received training in the entire PowerLogic SCADA product and will understand the application's basic functions.

What is SCADA?

SCADA (Supervisory Control and Data Acquisition) is a system that collects data from various points, both local and remote, and then stores the data at a central location. The system also can control equipment at an installation.

What is PowerLogic SCADA?

The PowerLogic SCADA product includes several tools that use the PowerLogic SCADA engine to collect information from all PowerLogic devices:

- In the Profile Editor, you create, configure, and modify profiles for device types that use the same tags (for example, to set up alarms and trends).
- In the Profile Wizard, you add and modify devices in the PowerLogic SCADA system.
- In the Project Color Change Tool, you set colours for the different states of circuit breakers and busbars. These colours display on the final runtime environment page.
- In the CitectSCADA Graphics Builder (design time mode), you create one-line drawings, into which you can incorporate objects (genies). Genies are versatile objects that include variables or other expressions that are specified when you view the drawing.
- In the CitectSCADA Graphics Builder (runtime mode), users can view the final one-line drawings, including alarms, events, and history data. With the appropriate degree of password-controlled authority, users can also perform advanced tasks, such as changing alarm setpoints and racking devices in and out.

Preparing for PowerLogic SCADA

For a full discussion about preparation for a PowerLogic SCADA project, see Planning a Project in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help). This section of the help discusses physical layout; requirements such as architecture and security; and project design.

Components of a Project

For detailed information about the components that make up a PowerLogic SCADA project, see Components of a project in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help). The project components include graphic components, tags, alarms, system components, communications components, I/O Server components and Cicode/CitectVBA.

Configuration Tools

Configuration tools consist of:

Profile Editor: Use this tool to select tags to be used by device types (tags must be consistent with IEC 61850 naming conventions), create device profiles for individual devices, and create projects that include the device profiles to be used in a single installation. You can specify real-time tags, PC-based alarm tags, onboard alarm tags, trend tags, and reset tags to be generated for this device.

Profile Wizard: Using this wizard, you will import device profile information from the Profile Editor into a PowerLogic SCADA project. This tool is simply a means of moving device profile information into the PowerLogic SCADA project and converting it into formats that PowerLogic SCADA can use.

Citect Explorer: Use Citect Explorer for basic navigation. From here, you also choose the active project.

Citect Project Editor: Use the Project Editor for entering database-type information, such as adding clusters and servers, creating new users, and editing tags within PowerLogic SCADA projects.

Graphics Builder, Design-Time environment: Use the Citect Graphics Builder to create one-line drawings that users can view in the runtime environment. These drawings are populated with interactive objects that are generated by genies. You can also use the graphics tool to set up system alarms and trends.

Project Color Change Tool: This tool is part of the Project Editor. Use it to change colours for the states of busbar and circuit breaker genies used in graphics pages.

Runtime Environment

The runtime environment is where the end user views system information. This environment includes (from the information added in the design-time page) one-line pages with interactive objects, alarm and event pages, and analysis pages (trends and waveforms).

Supported Device Type Drivers

When you install PowerLogic SCADA, you are prompted to choose the drivers that you will use. A certain number of generic drivers are installed by default (including PowerLogic device types), and you are not prompted for them. Device type drivers supported in PowerLogic SCADA are:

- Generic MODBUS (includes BCPM and any device, such as a PLC or UPS, that communicates via MODBUS). When adding a controllable device in the Profile Editor, such as a circuit breaker, use the “Controllable Device” driver; otherwise, use the “Generic Power Device” driver.
- Sepam 20, 40, and 80 Range
- MicroLogic 5P and 6P, A
- CM4000 series

- PM800 series
- PM700 series
- ION 7650 (MODBUS)

System Requirements and Installation Instructions

See the *PowerLogic SCADA Installation and Configuration Guide* for a complete discussion of system requirements and installation instructions. A PDF copy is on the installation disk.

Uninstalling and Reinstalling PowerLogic SCADA

Use Add/Remove Programs in the Control Panel to uninstall these PowerLogic SCADA programs:

- PowerLogic SCADA 7.10 (if you uninstall this, you also uninstall the Profile Editor and the Project Color Change Tool)
- PowerLogic SCADA Profile Editor 1.0
- PowerLogic SCADA Project Color Change Tool
- Any additional PowerLogic SCADA programs, such as the WebServer, that you installed

If you uninstall PowerLogic SCADA after you have already created projects, the project data will not be deleted. It is in [Project Drive]\Documents and Settings\ All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10 (for Vista operating systems, it is [Project Drive]\ProgramData\Schneider Electric\PowerLogic\PowerLogic SCADA 7.10). The first time you launch the application after you re-install it, it will locate the project data and re-link it.

Quick Start: Typical Workflow

The following is a brief description of the steps you will take when you create a PowerLogic SCADA system. Also, see *Typical Workflow Illustration* on page 18 for a flow chart that illustrates the Profile Editor workflow.

1. **Identify the equipment and devices to be used in the system**: List all of the device types that you will need.
2. **Create a system architecture drawing**: Indicate how all devices will interrelate.
3. **Prepare the network addressing scheme**: Define the Ethernet IP address for each gateway/PLC/CM4000 and the MODBUS slave address devices.
4. **Install the hardware**: Install the devices and equipment for the system. Create all of the communications connections that are necessary. Make sure that devices are set up for the onboard alarms and waveform captures that you want to view from graphics pages.
5. **Set up device type categories and units/conversions in the Profile Editor**: Use the Setup Device Type Categories screen to add categories (an optional feature that will make it easier to sort device types in the Profile Wizard). Also use the Profile Editor to add unit conversions (e.g., watts/megawatts) that are used in individual installations. Most typical conversions have already been added to PowerLogic SCADA.
6. **Add/edit device types in the Profile Editor**: The device type includes all of the details that define a device: real-time tags, alarm tags, trend tags, resets, and controls. Use the Define Device Type Tags tab to add and edit device types, create custom tags, and to assign tags to device types. Although the device type includes all of the information we know about a device, the device profile (see step 8) includes only the specific tags that are needed for that device in a customer's installation.
7. **(optional) Edit tag addresses in the Profile Editor**: As needed, add or edit the address attributes of individual tags for a single device type. This includes such items as the number of registers used by the tag, bitmasking, and logic code.
8. **Create device profiles in the Profile Editor**: The profile is a subset of the device type's functionality. Use the Create Device Profile tab to create a device profile; select only the tags required for a customer. On this tab and the related Add/Edit Device Profile screen, you also specify tags to be used for onboard waveform captures, PC-based alarms, and trends.
9. **Create and configure a project in Citect Explorer**: From Citect Explorer, add a new project. From the Project Editor page, add at least one cluster and network address, and the servers that will be used (alarm, report, I/O, and trend). Set 'Publish Alarm properties' to TRUE when setting up the alarm server (expand the screen by pressing F2). This must be set; otherwise you will not be able to edit while in the setpoints page.
10. **Create a corresponding project in the Profile Editor**: In the Set Up Projects tab, select the device profiles for the devices that are included in the project created in the previous step. Add the project to the Profile Editor, using the same name you created in the previous step. To view the list of projects created in PowerLogic SCADA, click the Display Projects button (on the Add/Edit Project screen) :

    Each profile will include all of the device attributes that you have set up in earlier steps.
11. **Export devices from the Profile Editor**: From the Set Up Projects tab, choose a single project and click Export Project to convert device tags into a format usable

within PowerLogic SCADA, and to move them to the location where they can be imported into PowerLogic SCADA.

12. **Use the Profile Wizard to set up clusters and add devices to PowerLogic SCADA**: Use the Profile Wizard to set up clusters (created in step 9). See *Adding a Device to a PowerLogic SCADA Project* on page 72 for additional information.

    Use the Profile Wizard to add the exported devices, one at a time; or use the Automation Wizard to add all of the devices from one page (example found in: Program Files\Schneider Electric\PowerLogic SCADA 7.10\Bin*Example Automation.xls*). You will need device networking information for this step.

13. **Compile and correct errors in PowerLogic SCADA**: Note: It is always a good idea to pack before you compile (Project Editor > File > Pack). To compile the project, click Compile, located at the top of the Project Editor:

    Then correct any errors and note any warnings.

14. **Create graphics in the Citect Graphics Builder**: Use the Graphics Builder (the design time environment) to create one-line diagrams that incorporate icons and objects for the user to read information from the system. You will use this feature to add genies that, when launched from the runtime environment, provide real-time system information.

15. **Use menu.dbf to set up the appearance of the runtime environment**. Menu.dbf is the template that controls tabs and menus for a specific project in the runtime environment. It is located in Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\User\[project name]\menu.dbf (for Vista operating systems, it is ProgramData\Schneider Electric\PowerLogic SCADA 7.10\User\[project name]\menu.dbf). You can copy and paste from the standard template in the PLS_Example project; or you can create your own template.

16. **Add at least one user in the Citect Project Editor**. From the Users window (System > Users), add at least one user to the project.

17. **Set colours for busbars and circuit breakers in the Project Color Change Tool (Citect Project Editor)**: Click Tools > Project Color Change Tool to launch. Use the Color Change Tool to add unique colours for different states of busbars and circuit breakers. You can also add a second colour if you want the object to flash between the two colours when it is in a particular state. (Each time you add a genie, after the system is set up, you need to run the Project Color Change Tool again to ensure that the colours are assigned to the new genie.)

18. **Run the Computer Setup Wizard**. This wizard, located in Project Editor > Tools > Computer Setup Wizard, customizes your computer for use with PowerLogic SCADA. To run the wizard, click:

19. **Compile (File > Compile), then run the project**: Compile the project:

    Then correct any errors and note any warnings. Run the project:

    Each time you add genies to a project, you need to associate them with devices, and then compile and run again.

20. **Test the system**: Verify that the runtime environment objects all operate correctly, that the system is recording data and that all alarms and waveforms function.

Getting More Information

For additional information, use one of these options:

Manuals

Help files

Customer support

Manuals

In addition to this *System Integrator's Guide*, there are these additional documents that may provide helpful information:

- **PowerLogic Installation and Configuration Guide**: a PDF copy is on the installation disk
- **Release Notes**: located on the installation disk, this file includes information specific to this release of the product
- **Readme file**: located on the installation disk, this file also includes late-breaking information about this release

Help files

In addition to the help file released with PowerLogic SCADA, there are several PowerLogic SCADA-specific help files. They are located in the PowerLogic SCADA Bin folder: \Schneider Electric\PowerLogic SCADA 7.10\Bin).

Customer support

Americas: +1 250 544 3010 (24 hours, Monday-Friday)

Asia/Pacific: +852 2579 9568

Europe/Africa: +33 4 76 60 62 35

e-mail: Global-PMC-Tech-Support@Schneider-Electric.com

web support: www.powerlogic.com

The Profile Editor

This section describes how to use the Profile Editor to create device type tags, and then to use these tags as building blocks for device types. Secondly, there are instructions for creating device profiles for unique devices. Finally, there are instructions for creating projects for each installation or customer.

In this section, you will find these topics:

Overview of the Profile Editor
- *Typical Workflow Illustration*

Launching the Profile Editor

Adding Engineering Unit Templates, Units, and Conversions

Defining Device Types and Tags
- *Device Type Screens and Workflow*
- *Printing the .CSV File*
- *Adding Custom Tags and Tag Addresses*
- *Adding, Editing, or Deleting a Device Type*
- *Editing Tag Addresses*

Creating Device Profiles
- *Enabling Waveforms*
- *Viewing Device Profiles*
- *Adding, Editing, or Deleting a Device Profile*
  - *Adding a Device Profile*
  - *Editing a Device Profile*
  - *Deleting a Device Profile*

Setting Up Projects in the Profile Editor
- *Project Screens and Workflow*
- *Viewing Project Data and Exporting the Project*
- *Adding, Editing, or Deleting a Project*
- *Exporting a Project*
- *Editing and Deleting Information in a PowerLogic SCADA Project*
- *Customizing Tag Names*

Understanding Tags
- *Customizing Tag Names*
- *IEC61850 Tag Construction*
- *Defining an Enumeration*
- *Format Code Definitions*
- *Logic Codes*

Overview of the Profile Editor

The Profile Editor is a multiple-screen application that allows you to perform four basic tasks: create device types, create device profiles, and set up projects.

Use the **Define Device Type Tags** tab and its screens to add and edit information for real-time, onboard alarm, control and reset tags and to create and edit device types. See *Defining Device Types and Tags* on page 27 for complete instructions.

Use the **Create Device Profiles** tab and its screens to add and edit individual profiles for specific devices. A device profile is a subset of the possible variable tags, alarm tags, and trend tags for a particular device type. See *Creating Device Profiles* on page 44 for complete instructions.

Use the Set Up **Project** tab and its screens to bring together all of the system attributes for a single customer or installation. For example, the customer installation will include a certain combination of device profiles (depending on the devices installed at the site). The project allows a specific unit template to be applied, converting units (such as watts) into units used by the customer (such as megawatts). This causes tags to display in the converted format. Projects also allow you to rename tags to suit a customer's needs (for example, Current A could be renamed to Current Phase A). See *Setting Up Projects in the Profile Editor* on page 49 for complete instructions.

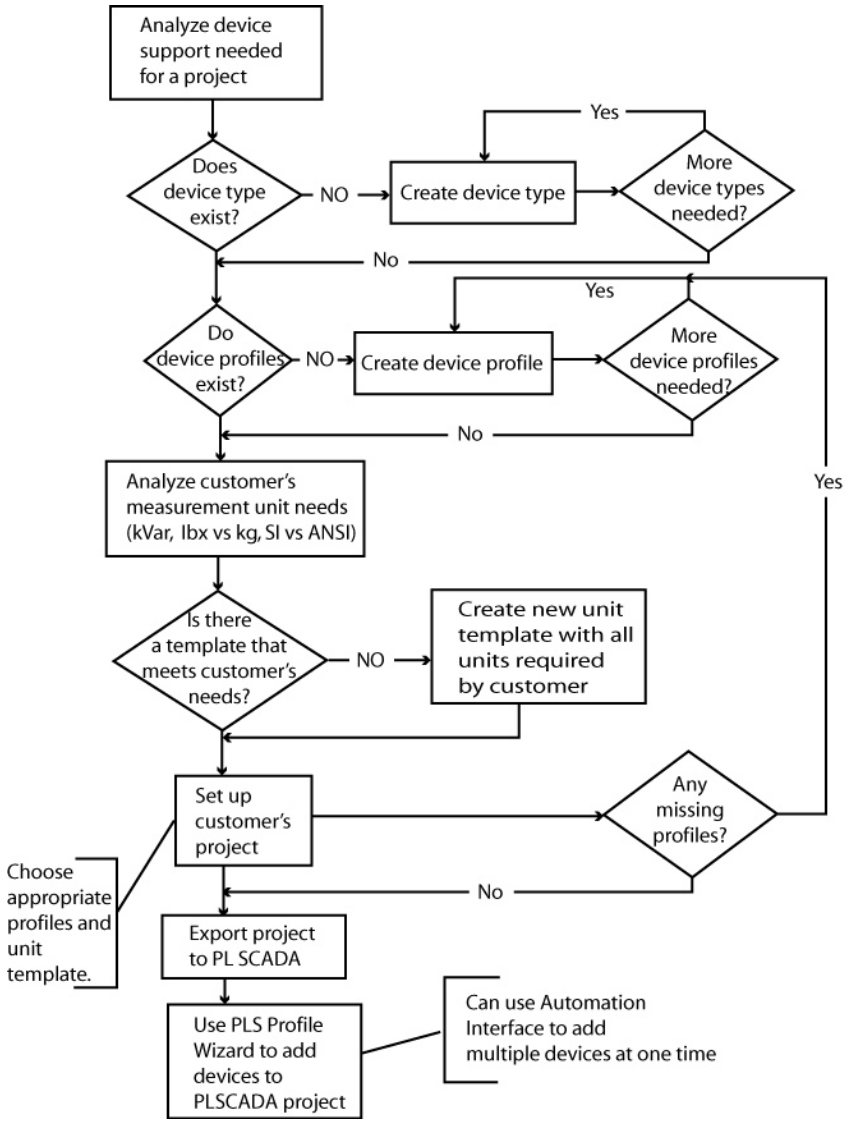
PowerLogic SCADA uses the IEC61850 tag-naming convention to create tags that measure device quantities. Although most of the tags you will use are already entered into the system, you can add custom tags. For more information, see *Understanding Tags* on page 54.

Typical Workflow Illustration

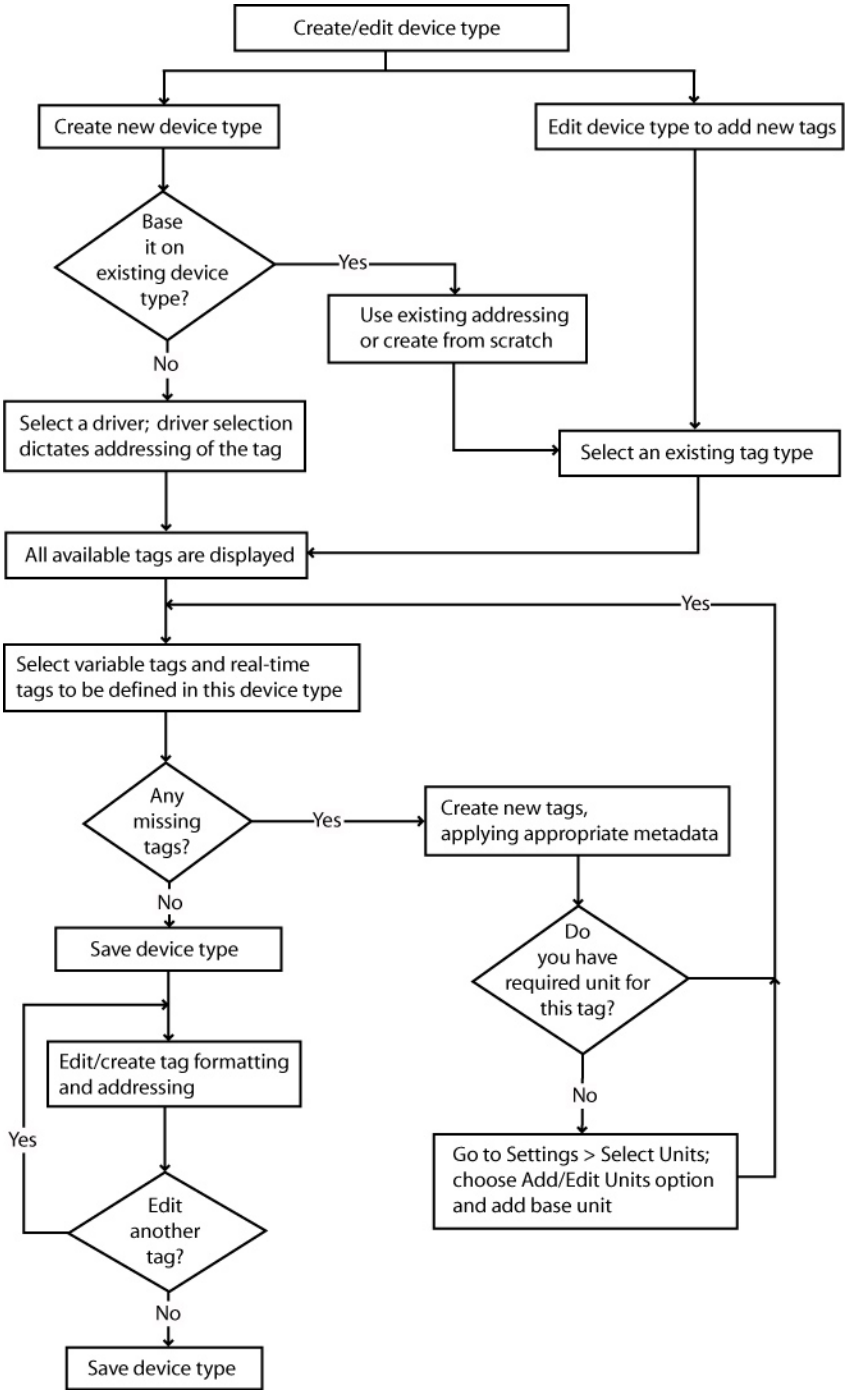
The following four flow charts illustrate the use of the Profile Editor. The first illustration provides an overview, and the following illustrations show:

- creating/editing a device type
- creating/editing a device profile
- creating/editing unit templates

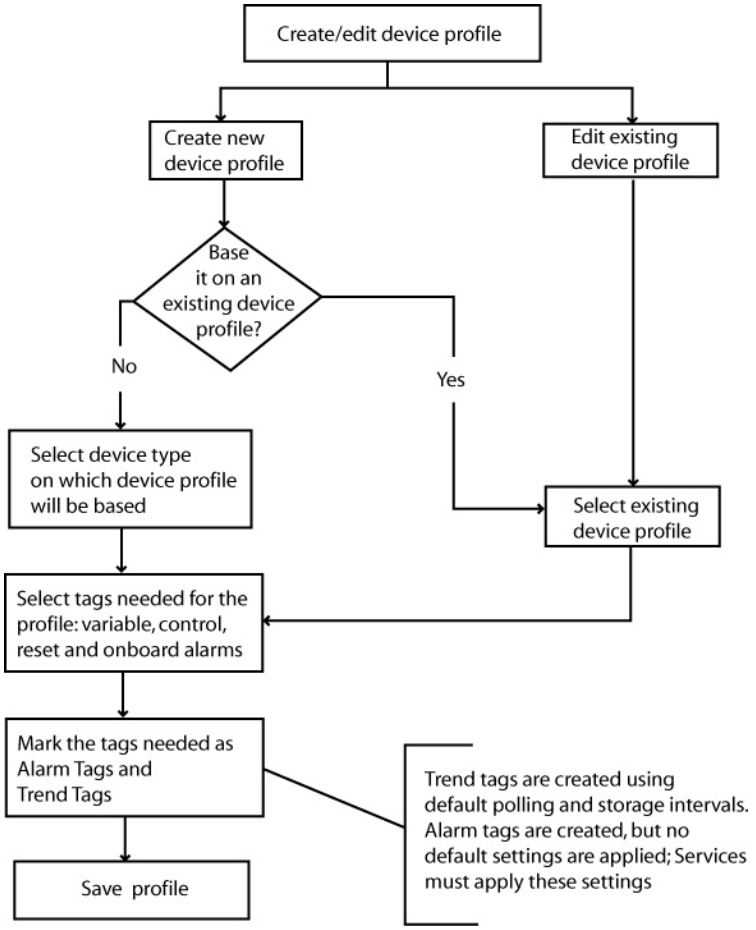
Workflow Overview



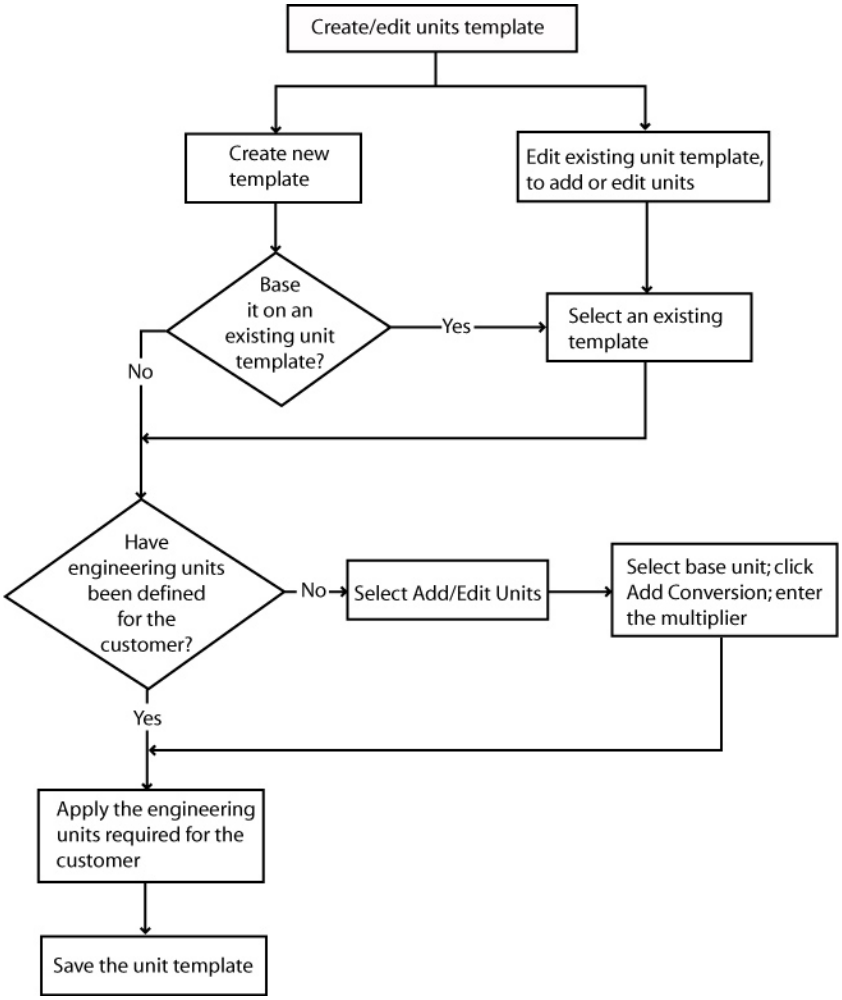
Create/Edit Device Type



Create/Edit Device Profile



Create/Edit Unit Templates



Launching the Profile Editor

To launch the Profile Editor, click Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > Config Tools > Profile Editor. The Profile Editor screen displays with the Define Device Tags tab selected. There are two other tabs, used to create device type profiles and projects.

Locked and Custom Icons

Two icons may appear to the right of the Add/Edit button on some screens: the locked icon and the custom icon.

The Locked Icon 

:

This icon indicates that the selected file (e.g., device type, profile, or project) cannot be edited. All standard device types (for example, Circuit Monitor 4000, MicroLogic Type P, Power Meter 800) are automatically locked; they cannot be unlocked.

To lock a device type that you create, follow these steps:

1. From the Define Device Type Tags screen, click Add/Edit.
2. On the Add/Edit Device Type screen, you can choose Create New, Create From, or Edit Existing.
3. Enter the new device information, or select the device to be edited.
4. To lock the device, check the box named Lock This Device Type.
5. Click Save & Exit to save the “lock” and exit the screen.

After you lock a device type, you cannot unlock it. However, you can restore it in this manner: Copy the locked device type (Create From on the Add/Edit Device Type screen), then save the copy with a new name.

The Custom Icon 

:

This icon indicates that a device type or profile is user-created. It may have been created new, created from an existing device type or profile, or created by editing an unlocked custom device type or profile.

Setting the Screen Resolution

Depending on the screen resolution you use, some of the Profile Editor screens may take up the entire viewing area. We recommend that you use at least 1024 x 768 resolution. You can also auto-hide the taskbar to provide more room.

Adding Engineering Unit Templates, Units, and Conversions

An engineering unit is a part of a tag. Use engineering unit templates to simplify the conversion between base units and their conversions (such as inches to centimetres) and to provide consistency in recording data in reports and on-screen viewing. For example, in one project you might want to see amperes reported as kiloamps. In another, you might want to see amperes as milliamps. You will use the Units screens to determine the conversion for standard units and custom units (tied to custom tags) that you create.

You can also create templates to organize user-created unit/conversion pairs. Each template will include all of the predefined engineering units and conversions, as well as the ones you assign to it. These templates can then be used in system projects (see the Set Up Project tab for creating projects).

In this section, you can learn about:

Setting Up Engineering Templates and Selecting Conversions

Adding or Editing a Base Engineering Unit or Conversion

Setting Up Engineering Templates and Selecting Conversions

Use the Set Up Engineering Unit Templates screen to add, edit, or delete an engineering unit template, and to begin adding, editing, or deleting templates, engineering units, and conversions to the system. To view the Set Up Engineering Units screen, click Settings > Set Up Engineering Unit Templates. The following table describes the parts of the Set Up Engineering Unit Templates screen (it assumes that Display 'Advanced' Fields is checked). Detailed instructions are after the table.

Field Name	Valid Entries	Comments
Options box	Create New	Click to begin creating a new engineering unit template.
	Create From	Click to create an engineering units template that is based on an existing template.
	Edit Existing	Click to edit an engineering unit or its conversion.
	Delete	Click to begin deleting an engineering unit and its conversion. You cannot delete a locked template.
Unit Template to Create From	From the drop-down menu, select the template you wish to copy, in order to create a new template.	This field is live only when *Create From* is chosen as the option. The new template will initially include all of the units/conversions of the original; but you can add units and change the conversion settings.
Unit Template Name	This field is blank if you selected *Create New* or *Create From*; type the name of the new template. A name displays if you have selected a template to edit; you can change the name. A name displays, but it is greyed out if you selected a template to delete. Click *Save* to save the changes you make.	When creating a new template or creating from an existing template, type the name of the new template. To change the name of an existing template, choose it from the *Unit Template to Delete* menu, then change the name here.

Field Name	Valid Entries	Comments
Lock this Template	Click to prevent the template from being edited in the future.	The only way to "edit" a locked template is to delete it, and add back a new one with the edits entered.
Display Associated Projects	Live only when in "Edit" mode. Displays all projects that use this template.	You only need this if you want to delete a template that is associated with a project. Note the projects that display in the list, then go to the Set Up Project tab. For each project that you noted, change the unit template.
Display 'Advanced' Fields	Check this box to display additional columns of information about the template.	Unchecked: displays the unit and its abbreviation only. Checked: displays also the conversion, and its abbreviation and multiplier.
Base Unit	n/a	Many standard units are pre-defined; they cannot be edited or deleted. To add a unit or edit a user-created unit, see *Adding or Editing a Base Engineering Unit or Conversion* on page 25.
Abbreviation	n/a	Added for the unit when the base unit was created. To edit a user-created unit, see *Adding or Editing a Base Engineering Unit or Conversion* on page 25.
Selected Unit	Click the down arrow to display and select the preferred conversion for the unit.	Many conversions are pre-defined. To add or edit a conversion unit, see *Adding or Editing a Base Engineering Unit or Conversion* on page 25. Fahrenheit to Celsius temperature conversions must be handled by editing Cicode (Citect.ini).
Abbreviation	n/a	This is abbreviation for the selected unit. When the Selected Unit is changed, this field changes accordingly.
Multiplier	n/a	Added for the unit and for the conversion when the base unit was created. Pre-defined units/conversions cannot be changed. To edit a user-created unit, see *Adding or Editing a Base Engineering Unit or Conversion* on page 25.
Add/Edit Units button	Click to display the *Add/Edit Units* screen.	Use that screen to add units/conversions, or to edit user-created units/conversions.

Creating or Editing an Engineering Unit Template

To create or edit a template:

1. From the Define Device Type Tags tab, click Settings > Set Up Engineering Unit Templates.
2. Click Create New or Create From to begin adding a template. Click Edit Existing to edit a template.
   a. If you are creating a new template, type the name in the Units Template Name field.
   b. If you are creating from an existing template or editing an existing template, choose that template from the drop-down list. Type the new of the new template in the Units Template Name field.
   c. If you are editing an existing template, choose that template from the drop-down list.
3. Click Save to save a new or edited template, or click Save & Exit to save the template and close the screen. See the next section, Choosing Conversions, to change the unit conversions. To add new unit conversions, see *Adding or Editing a Base Engineering Unit or Conversion* on page 25.

Selecting Conversions

Use this screen to select unit conversions. To add a conversion, see *Adding or Editing a Base Engineering Unit or Conversion* on page 25.

To select a conversion:

1. From the main window of the Profile Editor, click Settings > Set Up Engineering Unit Templates.
2. Click Edit Existing, then select the template for which you want to select unit conversions.
3. Locate the first base unit you want to change. In the Selected Unit column, click the down arrow to select the conversion you want to use. Fahrenheit to Celsius temperature conversions must be handled by editing Cicode (Citect.ini file).
4. Repeat step 3 for all units that you want to change.
5. Click Save to save the change, or click Save & Exit to save changes and close the screen.

Deleting a Template

You cannot delete a locked template .

To delete a template:

1. From the Define Device Type Tags tab, click Settings > Set Up Engineering Unit Templates.
2. Click the Delete radio button, on the left, to delete a template.
3. Choose the template from the drop-down list.
4. Click Delete, on the right, to delete the selected template. At the Confirm Delete prompt. click Yes.

Adding or Editing a Base Engineering Unit or Conversion

Use the Add/Edit Units screen to add, edit, or delete base units and conversion units for custom tags. You cannot make any changes to predefined units (those that are greyed out). At the Set Up Engineering Units screen, click Add/Edit Units. The following table describes the fields of the Add/Edit Units screen. Detailed instructions are after the table.

Field Name	Valid Entries	Comments
Unit Conversions	n/a for pre-defined units/conversions (greyed out) Select user-created units to begin edits.	All base engineering units and their conversions display. Greyed-out items are predefined; they cannot be edited or deleted. Note that predefined units can have custom conversions, which are editable.
Add Base	Click to begin adding a new base unit.	The Base Unit and Base Unit Abbreviation fields become live.
Add Conversion	Click to begin adding a conversion to a base unit.	The Base Unit field displays the unit you highlighted; the Conversion Unit, Conversion Abbreviation, and Multiplier fields become live.
Edit Unit/ Delete Unit	Click to either edit a custom unit/conversion, or to delete it.	These buttons are only live when you select a custom, unlocked, unit.
Base Unit	When editing a unit/conversion, select the unit from this drop-down menu. When adding a new base unit, type the name.	Used in the Profile Editor only; not passed to PowerLogic SCADA projects for graphics viewing.

Field Name	Valid Entries	Comments
Base Unit Abbreviation	Type the abbreviation for the selected base unit.	If there is no conversion, this is passed to PowerLogic SCADA for viewing graphics.
Conversion Unit	Type the name of the conversion unit, e.g.,milliamps, when amps is the base unit.	Used in the Profile Editor only; not passed to the PowerLogic SCADA project for graphics viewing.
Conversion Abbreviation	Type the abbreviation for the conversion unit.	This is passed to the PowerLogic SCADA project for viewing graphics.
Multiplier	Use this field to determine the number of base units that are in the conversion unit. Type the multiplier "M," where Conversion Unit x M = Base Unit.	Example: There are 1,000 bytes in a kilobyte; so, the conversion unit multiplier is 1000, If you have 17.3 kB, 17.3 x 1,000 = 17300 bytes

Adding a Base Engineering Unit and Unit Conversion

To add an engineering unit and/or conversion:

1. Click Add Base.
2. In the Base Unit field, type a unique name for the new unit.
3. In the Base Unit Abbreviation field, type an abbreviated name.
4. Click Save.
5. To add a conversion, highlight the name of the Unit, then click Add Conversion.
6. Type the name of the Conversion Unit.
7. Type the name of the Conversion Abbreviation.

Editing a Base Engineering Unit or Conversion

Changes are global, for all templates. You cannot change any predefined engineering units/conversions (greyed out).

To edit a unit or conversion:

1. To edit the unit: With the base unit highlighted, click Edit Unit. You can edit the base unit and base unit abbreviation. Click OK to save the changes.
2. To edit a conversion: With the conversion highlighted, click Edit Unit. You can edit the conversion unit, abbreviation, and multiplier. Click Save to save the changes or click Save & Exit to save the changes and close the screen.

Deleting a Base Engineering Unit or Conversion

Deletions are global, for all templates. You cannot delete units any predefined units/conversions (greyed out).

To delete a unit or conversion:

1. To delete a unit: With the base unit highlighted, click Delete Unit. At the Confirm Delete prompt, click Yes. Click OK to close the screen.
2. To delete a conversion: With the conversion highlighted, click Delete Unit. At the Confirm Delete prompt, click Yes. Click Save & Exit to save the changes and close the screen.

Defining Device Types and Tags

The Define Device Type Tags tab and its related screens are used to define device-related data: custom tags, device types, and base units/conversions. You will use several screens to add and manage this data.

To access these screens, click the Define Device Type Tags tab on the main Profile Editor screen.

In this section, you can learn about:

Device Type Screens and Workflow

Using the Define Device Type Tags Tab

Adding, Editing, or Deleting a Device Type

Printing the .CSV File

Setting Up Engineering Templates and Selecting Conversions

Adding or Editing a Base Engineering Unit or Conversion

Device Type Screens and Workflow

On the Define Device Type Tags tab, follow these general steps to add tags and devices to your system:

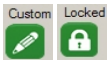
1. To manage the units and unit conversions that you want to use (such as amperes into kilo amperes or milliamperes), see *Adding or Editing a Base Engineering Unit or Conversion* on page 25.
2. To manage the tags you want to use, see *Defining Device Types and Tags* on page 27.
3. To add and edit custom tags, see *Adding Custom Tags and Tag Addresses* on page 36.
4. To add or edit device types, see *Adding, Editing, or Deleting a Device Type* on page 30.
5. To establish device type categories and subcategories, used in reporting, see *Printing the .CSV File* on page 34.
6. To edit tag addresses, see *Editing Tag Addresses* on page 40.

The PowerLogic SCADA system uses the IEC61850 tag naming convention. For information about the types of IEC61850 tags and their configuration, see *Understanding Tags* on page 54.

Using the Define Device Type Tags Tab

The Define Device Type Tags tab displays device types and all of the tags that may be associated each device type. This includes real-time, onboard alarm, control, and reset tags. Most of the fields on this tab are read only (they can be changed on other screens). The following table describes the parts of the this tab. The tags listed assume that Advanced Properties has been checked. Not all elements appear on every sub-tab.

Field Name	Valid Entries	Comments
File and Settings menus You can also use the icons beneath the menu to save changes, create a .CSV file or create the spreadsheet for printing.	File > Save	Saves any current changes.
	File > Create CSV file	Creates .CSV file of basic tag data. Store the file in a folder you designate. Can be viewed in Excel.
	File > Print Tag Selections	Displays a print preview of all of the tags for this device type. You can then print the spreadsheet.
	Settings > Display Advanced Properties	Causes additional “advanced information” columns to display.
	Settings > Set Up Custom Tags	Displays the Add/Edit Custom Tags screen. See *Setting Up Custom Tags* on page 36 for a description of this screen.
	Settings > Set Up Device Categories	Displays the Set Up Device Type Categories. See *Setting Up Device Type Categories* on page 34 for a description of this screen.
	Settings > Set Up Engineering Unit Templates	Displays the Set Up Engineering Unit Templates screen. Click *Setting Up Engineering Templates and Selecting Conversions* on page 23 for a description of this screen.
Tag groups (left-hand pane)	Select a tag group; the tags included in that group display on the right.	Each tag belongs to a group. The group is determined when the device is added to the system. For custom tags, this is on the Add/Edit Custom Tags screen. Tags for standard device types are pre-determined and cannot be changed.) **Note**: If a tag group displays in red copy, there is at least one address that is not valid for the tag to which it is assigned. To correct this issue, click the tag group, ensure that Display Advanced Properties is selected, then scroll down through the tags in the right-hand column. The tags that have invalid addresses will have the “Edit…” displayed in red. Click this field to open the Edit Address page; correct the errors in the address.

Field Name	Valid Entries	Comments
Tag tabs: Real Time, OnBoard Alarm, Control, and Reset	Click a tab to view the tags of that type that are included for the selected device type.	For device types that are not locked: You can use the Add/Edit Device Type screen to edit the list of tags.
Device Type Name	Select the device type.	Each device type includes a different number of tag categories, which also changes the list of tags that display. The device list includes the default device types, as well as any that have been created for this system.
Add/Edit button	Click to open the Add/Edit Device Type screen.	Provides a means of adding new device types and editing custom device types (user-created device types that are not locked). Also provides a means of adding new custom tags and editing existing tags.
Locked/Custom icons: 	n/a	Locked icon indicates that the list of selected tags cannot be edited. Custom icon indicates that the device type was created by a user. See *Locked and Custom Icons* on page 22 for complete information.
Category/ Subcategory/ (Description)	n/a	These optional fields, which display under the Device Type Name, are for information only. They are added when the device type is defined.
Tag Description	n/a	For custom tag: This name comes from the Tag Name field in the Add/Edit Custom Tags screen.
Units	n/a	Displays the units that were added for this tag when it was created.
IEC Tag Name	n/a	Tag name that conforms to IEC61850 standard. See *Understanding Tags* on page 54 for more information.
Type	n/a	Displays the data type chosen when the tag was created.
Address	n/a	Displays the address information for this tag, including elements such as type of register, number of registers, and scaling and bitmasking data. See *Understanding Tags* on page 54 for a detailed description of address construction.
Edit Addr	Click to display Edit Address screen.	Provides the means of changing the elements of an unlocked real-time tag address (for example, the number of registers, their numbers, and whether they are consecutive). See *Editing Tag Addresses* on page 40. for detailed information.
Register 1	View or enter the register number.	This field contains first register used to store this tag. If there are additional registers, they are indicated in the address. The total number of registers is listed in the Num Registers column. This field allows you to verify and/or change the value of Register 1 without having to open the Edit Address screen. Note: If you enter a number that is not compatible with other address settings, you are prompted to go to the Edit Address screen.
Num Registers	n/a	Displays the number of registers used by this tag.
Formatting	From the drop-down menu, select the format type.	After you change formatting for a tag and move to another field, you are asked whether you want to open the Address Editor. If you click No, the format is unchanged; if you click Yes, the Edit Address screen opens for you to enter the appropriate changes for this tag. See *Editing Tag Addresses* on page 40.
Scaling Register	View or enter the register number.	This is entered in the Edit Address screen, but it can be edited here. It is the register used to read the value for scaling. Note: If you enter a number that is not compatible with other address settings, you are prompted to go to the Edit Address screen.

Field Name	Valid Entries	Comments
Category Type Utility Type Statistical Type Quantity	n/a	Each of these types is a real-time filter, added when the tag was created. See *Setting Up Custom Tags* on page 36 for more information.
Categorization Subcategorization Alarm Type (onboard alarm) Alarm Group (onboard alarm) Alarm Level (onboard alarm)	n/a	Each of these is an onboard alarm folder, added when the tag was created. See *Setting Up Custom Tags* on page 36 for more information.
On Command (control)	n/a	Determined when the tag was added.
Off Command (control)	n/a	Determined when the tag was added.
Normally Closed (control)	Check this box to invert the functionality of the control. See description.	For a control with one command, writing a 1 to the tag will cause the command to occur. (This option is greyed out.) For a control with two commands that is either static or normally open, writing a 1 to the tag will cause the first command to occur; writing a 0 will cause the second to occur. (Checkbox not checked.) For a control with two commands that is normally closed, writing a 1 to the tag will cause the second command to occur; writing a 0 will cause the first command to occur. (Checkbox checked.)
Command (reset)	n/a	Lists the quantity that is to be reset for this tag.

Adding, Editing, or Deleting a Device Type

Use the Add/Edit Device Type screen to begin adding, editing, or deleting a device type from the system. To view this screen, click Add/Edit, to the right of the Device Type Name field. The following table describes the parts of the Add/Edit Device Type screen. Detailed instructions are after the table.

Field Name	Valid Entries	Comments
Device Type Options Box		
Create New		Click to begin adding a new device type that is not based on an existing type.
Create From		Click to make a copy of any existing device type.
Edit Existing		Click to edit an unlocked device type.
Delete Existing		Click to delete an unlocked device type that is not associated with a profile.
Device Type	select type	Select the device type that you want to create from, edit, or delete.
Copy Addressing		Active only when you choose Create From. Check this box if you want to copy the addressing of the “from” device. This gives each tag in the new device type the same address string as the matching tag in the “from” device.

Field Name	Valid Entries	Comments
Device Type Name	Type or select the name: maximum 32 characters, do not use \ / : * ? < > |	If creating a device type, type the name. If editing an existing device type, the device type that was selected for editing displays here. You can overwrite the name here to change it.
Lock this Device Type	Check to lock device, so that it cannot be edited.	Once locked, you cannot edit a device type. If it is a standard, you cannot edit or delete it. If you added the device type, you can delete it, but not edit it.
Device Category	Choose the category for this device.	To create categories, see the *Printing the .CSV File* on page 34 screen. Categories are used in the Device Creation Wizard
Set Up Categories	Click to open the Set Up Device Type Categories screen.	Use this option if you need to add a new category.
Subcategory	Choose the subcategory for this device, if needed.	As with categories, subcategories are created in the *Printing the .CSV File* screen.
Driver	Select the driver for the device type.	Predefined drivers have been created for all PowerLogic compatible devices, though you may need to use these drivers again. For example, you would use the CM4000 driver for a CM3000. Use the Generic Power Device driver for third-party devices. The Controllable Device driver is currently not used.
Display Associated Profiles	(Active only in Edit mode) Click to display a list of profiles that are associated with the selected device type.	This list is ant when you want to delete a device type that is associated with a profile. In Edit mode, select the device type you want to delete; then note the profile(s) associated with the device type. Before you can delete the device type, go to the Add/Edit Device Profile screen, locate each profile in the list. You cannot save an empty profile. so you must either select another device type for it or delete the profile. Then you can delete the device type.
Description	(optional) Type a description for this device type.	This entry will also display on the Define Device Type Tags tab. It will be in parentheses, following the category and subcategory names.
Tags	n/a	This list includes all tags that have been added to the system, standard tags as well as custom tags that you have added. Tags are listed in their groups (100ms, Onboard Alarm, Power Factors, etc.).
Selected Tags	Select tags from Tags; click the right arrow to move them to this box.	You can move single tags or entire tag groups. They must be moved one at a time (cannot shift+click to select). Note: You cannot deselect tags for a device type if that device is associated with a device profile.
Custom Tags	Click to begin adding a custom tag.	Live when creating or editing a tag. Opens the Add/Edit Custom Tags screen. See *Adding Custom Tags and Tag Addresses* on page 36 for instructions. If you add a custom tag here, you are prompted to save the device type. After adding the tag, you have the option of adding that tag to the device type.

Adding a Device Type

To add a device type:

1. From the Define Device Type Tags tab, click Add/Edit.
2. In the Device Type Options box, click Create New or Create From.

   If you are creating from another device type, choose it from the Device Type to Create From drop-down list.
3. Type a Device Type Name for the new device type; maximum 32 characters. Do not use \ / : * ? < > |
4. If you want to lock the device, preventing anyone from editing it, check the Lock this Device Type box. This action cannot be undone. If you wish to edit a locked device, you must use the Create From feature to add a new one, then delete the locked one.
5. From the Device Category drop-down list, choose the category to which this device belongs. In addition to predefined categories, you can add custom categories. See *Setting Up Device Type Categories* on page 34 for instructions. Categories are used in the Device Creation wizard, and are a means of shortening the list of devices you must view.
6. From the Subcategory drop-down list, choose a subcategory. As with categories, subcategories are added in the Set Up Device Type Categories screen.
7. From the Driver drop-down list, choose the correct driver for this device type. Drivers have been added for all of the standard, PowerLogic-compatible, devices.
8. The Display Associated Profiles link displays a list of profiles to which a device type has been added. You will use this link only when you want to delete a device type that has been added to a profile.

   A device type that is associated with a profile will not display in the “Delete Existing” mode. So, to view associated devices, you need to switch to “Edit Existing” mode and display the device type you want to delete; then click this link. You will see a list of all profiles to which this device type has been added. Note all of the profiles. Then, do the following:

   Open the Create Device Profiles tab, and click Add/Edit.

Editing a Device Type

You cannot edit any locked device type. When a device type is locked, the locked icon displays on the Define Device Type Tags tab:

If you want an edited version of a locked device type, you must create a new device type from it and then delete the original device type. Certain “standard” device types can be used to create new types, but they cannot be deleted. Examples: Circuit Monitor 4000, Power Meter 800, and Sepam S42.

To edit a device type:

1. From the Define Device Type Tags tab, click Add/Edit.
2. In the Device Type Options box, click Edit Existing.
3. From the drop-down list, select the Device Type to Edit.
4. You can change the device type name, lock the device, and choose a new protocol.
5. Select tags and tag groups and move them into or out of the Selected Tags list.

   If a device type is associated with a device profile, you cannot deselect tags.

6. After all of the appropriate changes are made, click Save to save your current settings or click Save & Exit to save the settings and close the Add/Edit Device Type screen.
7. To create additional custom tags, click Add/Edit Custom Tags; otherwise, click Save & Exit to close the window.

   If you add a tag to a group that is already included in a device type, you must then individually add the tag to that device type.

Deleting a Device Type

You cannot delete any standard device type or any custom device type that is associated with a device profile; those device types do not display in this option.

To view a list of profiles associated with a device type, switch to the Edit Existing view, then select the device type you want to delete. Click List Profiles associated with this Device Type to display all associated profiles. Before you can delete the device type, return to the Add/Edit Device Profile screen for each profile in the list; either select a new device type for the profile or delete the profile entirely.

To delete a device type:

1. From the Define Device Type Tags tab, click Add/Edit.
2. In the Device Type Options box, click Delete.
3. From the drop-down list, select the device type you want to delete (the list includes only unlocked or custom device types; you cannot delete any of the “standard” device types or custom device types that are associated with a profile.
4. Click Delete. At the Confirm prompt, click Yes.

Assigning Tags to Generic I/O Points

Device types have default tags that have the appropriate formatting and addressing assigned for all the generic I/O points. It may be necessary to redefine a generic I/O point by assigning it to a tag that has a specific meaning.

Example 1: The Branch Circuit Monitor 42 has been configured to read 42 current channels. To assign channel 1 to Current A, follow these steps:

1. From the Branch Circuit Monitor 42 device type, choose the “Ch.01 Current tag.”
2. Note the addressing and formatting for the tag.
3. Locate and add the standard tag that you want to assign to this channel. In the example above, you would add “Current A.”
4. Edit the address of the Current A tag to match the address of Channel 1.

Example 2: If the Sepam I11 / I12 have been configured to represent circuit breaker position, you may choose to redefine the tag name. Follow these steps:

1. From the Sepam 40 Series device type, choose tags “Input Status I11” / “Input Status I12.”
2. Note the addressing and formatting for each tag.
3. Locate and add the standard tag that you want to assign to these I/’O points. In the example above, you would add “Device Closed.”
4. Edit the address of the Device Closed tag. In order to create the “device closed” functionality, you must combine inputs 11 and 12 into an enumerated status

(choose the Enumerated Status logic code for the indicated address for I11 and I12),

Printing the .CSV File

For each device type, device profile, or project, you can create a .CSV file that includes the following data:

Type of File	Data Included
Device Type	tag descriptions, IEC tag names, type, and address
Device Profile	tag descriptions, IEC tag names
Project	data profiles and custom tag names included in the project

Follow these steps to create the CSV file:

1. Display the device type, profile, or project for which you want the file. For example, to create a CSV file for the Sepam 42 Full device profile, select the Create Device Profiles tab and choose Sepam S42 Full from the drop-down list.
2. Click File > Create CSV File.
3. From the Save As window, choose a location for the file and rename it, if you wish. Click Save.
4. The file is created in the location you specified.
5. For viewing and printing, open the file in Microsoft Excel.

Setting Up Device Type Categories

Use the Set Up Device Type Categories screen to add, edit, and delete categories. These categories are used in the Profile Wizard to logically group the list of profiles that display, and to make them easier to locate.

When you add device types in the Add/Edit Device Type screen, you associate a category and subcategory with each device.

To view the Set Up Device Type Categories screen, click Settings > Set Up Device Type Categories.

The following table describes the parts of this screen. Detailed instructions are after the table.

Field Name	Valid Entries	Comments
Categories Options box	Create New	Click to begin adding a new device type that is not based on an existing type.
	Edit Existing	Click to begin editing the category or subcategory name.
	Delete Existing	Click to begin deleting a category. You cannot delete a category that is associated with a device type.
	Category Name	If new: Type the name. If editing or deleting, select the name from the drop-down menu. Predefined categories do not display. Currently, there is one predefined category: Schneider Electric.
Subcategories Options box.	As with categories, you can create new, edit existing, or delete.	If new: Type the name. If editing or deleting, select the name from the drop-down menu. You cannot delete a subcategory that is associated with a device type. Predefined subcategories do not display. Currently, the predefined subcategories are: Protection Device, Monitoring Device, and PLC.

Adding a Category or Subcategory

To add a category/subcategory:

1. Click Create New in the appropriate box (Categories or Subcategories).
2. In the Name field, type the name of the new category or subcategory.
3. Click OK to save the new entry and close the screen.

Editing a Category/Subcategory Name

To change the name of a category or subcategory:

1. From the appropriate box, click Edit Existing.
2. From the dropdown menu, select the category or subcategory that you want to edit.
3. Type the new name for this category or subcategory.
4. Click Save to make the change, or click Save & Exit to save changes and close the screen.

Deleting a Category or Subcategory

Predefined categories/subcategories, or those associated with a device type, do not display for deletion. To delete a device type associated with a category/subcategory, change to the “Edit” view, select the category or subcategory, then click the List Device Types... button. Note the device types and go to the Add/Edit Device Types screen. Change the category/subcategory on that page. Then you can return to the Set Up Device Type Categories screen to delete the category/subcategory.

To delete a category or subcategory:

1. From the appropriate box, click Delete Existing.
2. From the dropdown menu, select the category or subcategory that you want to delete.
3. Click Delete.
4. At the Confirm prompt, click Yes.
5. Click Save to save the change, or click Save & Exit to save changes and close the screen.

Adding Custom Tags and Tag Addresses

PowerLogic SCADA comes with most of the tags that are needed for each device type. However, you can create custom tags to assign to device types and device profiles. You can also edit address attributes for any tag.

NOTE: To avoid potential communication errors, you should use the Profile Editor to create all custom tags that will communicate with equipment.

In this section, you can learn about:

Setting Up Custom Tags

Editing Tag Addresses

Setting Up Custom Tags

Use the Add/Edit Custom Tags screen to edit the information about a custom tag. To view this screen, click Add/Edit Custom Tags at the bottom of the Add/Edit Device Type screen or, from the upper left of the Profile Editor screen, click Settings > Set Up Custom Tags. Detailed instructions are after the table.

Field Name	Valid Entries	Comments
Custom Tag Options	Create New	Click to begin adding a new tag.
	Create From	Click to begin adding a new tag that is based on an existing custom tag. For example, you might want to change metadata for another custom tag.
	Edit Existing	Click to edit the attributes of an existing tag.
	Delete Existing	Click to delete a tag (tag cannot be associated with a device type).
	Tag to Edit/Tag to Delete	From the drop-down menu, select the tag you want to edit or delete.
	Delete button	Live only when Delete Existing is selected. Click to delete the tag. You can only delete custom tags not associated with a device type.
Display Associated Device Types	Click to display device types that are associated with this tag.	Live only when in Edit mode. Allows you to locate device types so that you can delete the tag from them before you delete the tag.
Display 'Advanced' filter selection	Check to display additional filter options in the Real Time Filter and Alarm Filter tabs	Displays several additional filter options on the two "Filter" tabs. These options will be useful in the future for reporting purposes.
List Device Types associated with this Tag.	In Edit mode, click to display all device types that are associated with the selected tag.	You will use this option when you want to delete a custom tag that is associated with a device type. To be able to delete the tag, you need to deselect it from all device types that are listed in the list (Add/Edit Device Type screen).

Field Name	Valid Entries	Comments
Tag Name	Type the new tag name; or type the changed name for a tag you are editing.	Maximum 32 characters; can include any alpha or numeric character, as well underscore (_) and backslash (\). Must begin with either an alpha character of underscore.
Description	Optional	You might use this field for additional information on the Add/Edit Custom Tags screen. For example, you could describe the data that it logs. It does not display anywhere else in the system.
Group	Select the group.	Includes all of the real-time groups (100ms, controls, currents, etc.) plus onboard alarms, resets, and controls.
Data Type	Select the data type.	These are PowerLogic SCADA tag data types. They affect the logic codes that are available for display in the Edit Address screen. See *Logic Codes* on page 59 for the data type that matches each logic code.
Eng. Units	Select the base unit.	These are the base engineering units for tags, from Engineering Unit Setup.
Ignore Unit Conversion	Check to cause the system to ignore any conversions that were added for this tag.	Causes reporting to be according to the base unit, rather than the conversion that was chosen for this tag in the template that is being used.
Add Engs Unit	Click to open the Add/Edit Units screen, to add a new engineering unit and/or conversion.	Provides a quicker means of adding an engineering unit that had been overlooked.
PowerLogic SCADA Format	Select the numerical format.	This is used in for display purposes in PowerLogic SCADA graphics pages. It determines where the decimal displays. For example, if you select #.##, the number 8.12579 would be displayed as 8.12.
Polling Priority	Low, Normal, or High	Indicates the level of priority PowerLogic SCADA uses when reading data from devices. Note: In the address field, a priority of 1 = High, 2 = Medium, 3 = Low.
Alarm On Text	For onboard alarms only: enter the text for when the alarm is On.	This text displays on the Create Device Profiles tab for the onboard alarm tag, when it is selected for the device type in the profile. It also displays in the Alarm Log.
Alarm Off Text	For onboard alarms only: enter the text for when the alarm is Off.	
Real Time Filter tab (list is expanded when "Display 'Advanced' filter selections" is checked)		
Category Type	Select a category for this tag.	This field provides metadata about the tag. It will be used in future reports.
Utility Type	Select a utility type.	Metadata for future use in reporting.
Statistical Type	Select a statistical type.	Metadata for future use in statistical reporting.
Quantity	Select a quantity.	Metadata for future use in statistical reporting.
Alarm Filter tab (list is expanded when "Display 'Advanced' filter selections" is checked)		
Categorization	Select the alarm category	Used for filtering and sorting alarm data. Also metadata for future use in statistical reporting.
Alarm Type	Select the alarm type.	Used for filtering and sorting alarm data. Also metadata for future use in statistical reporting.
Alarm Group	Select the group.	Used for filtering and sorting alarm data. Also metadata for future use in statistical reporting.
Subcategorization	Select a subcategory.	Used for filtering and sorting alarm data. Also metadata for future use in statistical reporting.
Alarm Level	Select the severity level of the alarm.	Used for filtering and sorting alarm data. Also metadata for future use in statistical reporting.

Adding a Custom Tag

Follow these steps to create a custom tag that is not already used by PowerLogic compatible devices; for example, to create a tag for a third-party device.

To add a tag:

1. Open the Add/Edit Custom Tags screen (from the Add/Edit Device Type screen, click Add/Edit Custom Tags).
2. In the Custom Tag Options box, click Create New *or* Create From.

   If you are creating from custom another tag, choose it from the Tag to Create From drop-down list.
3. List Device Types associated with this Tag: Used when you want to delete a tag that has been associated with a device type on the Add/Edit Device Type screen. See *Deleting a Custom Tag* on page 39 for instructions on using this button.
4. Tag Name: Type a descriptive name for the new tag (see table above for limitations).
5. Description (optional); You might use it to add an additional description about the tag, such as how it is used in logging. It displays on this screen, but not elsewhere in the system.
6. Group drop-down menu: Choose the group to which you are adding this tag. IEC tags are organized into these groups. There are multiple groups for real-time tags, and one group each for onboard alarms, controls, and resets.
7. Data Type drop-down menu: Choose the PowerLogic SCADA data type that matches the logic code for this tag. See *Logic Codes* on page 59 for a list of all logic codes with references to the correct data types.
8. Units drop-down menu: Choose the unit of measurement used for this tag. These are the base units set up in the Units screens.
9. To report on the base unit for this tag, rather than the unit conversion has been added in the template you are using, check Ignore Unit Conversion. All reporting will use the base unit for the tag.
10. PowerLogic SCADA Format: Choose the PowerLogic SCADA display reporting format, from ## to #0.#####.
11. Polling Priority drop-down menu: Choose the priority that PowerLogic SCADA will use when reading data from devices: high, normal, or low. In the address field, a priority of 1 = High, 2 = Normal, 3 = Low.

Real-Time Filters: (list is expanded when Display 'Advanced' Filter Selections is checked)

12. Category Type drop-down menu: Choose the category or N/A. Category type populates Custom 2.
13. Utility Type drop-down menu: Choose the type of utility or N/A. Utility type populates Custom 4.
14. Statistical Type drop-down menu: Choose the statistical type or N/A. Statistical type populates Custom 5.
15. Quantities drop-down menu: Choose a quantity or N/A. Quantities populates Custom 6.

Alarm Filters: (list is expanded when *Display 'Advanced' Filter Selections* is checked)

16. Categorization Type drop-down menu: Choose the type or N/A. This data is currently used for filtering and sorting alarm data in PowerLogic SCADA. Category type populates Custom 3. Metadata will be used in future reporting.

17. Alarm Type drop-down menu: Choose an alarm type or N/A. This data is currently used for filtering and sorting alarm data in PowerLogic SCADA. Alarm type populates Custom 4. Metadata will be used in future reporting.
18. Alarm Group drop-down menu: Choose a group or N/A. This data is currently used for filtering and sorting alarm data in PowerLogic SCADA. Alarm group populates Custom 5. Metadata will be used in future reporting.
19. Subcategorization drop-down menu: Choose a subcategory or N/A. This data is currently used for filtering and sorting alarm data in PowerLogic SCADA. Subcategorization populates Custom 6. Metadata will be used in future reporting.
20. Alarm Level drop-down menu: Choose the severity level of the alarm. This data is currently used for filtering and sorting alarm data in PowerLogic SCADA. Alarm level populates Custom 7. Metadata will be used in future reporting.
21. Click Save to save changes, or click Save & Exit to save changes and close the screen.
22. When the Select Tag screens, click Yes if you want to add this tag to the currently selected device type; click No to not add it.

Editing a Custom Tag

You can edit any custom tag.

To edit a tag:

1. Open the Add/Edit Custom Tags screen (from the Add/Edit Device Type screen, click Add/Edit Custom Tags).
2. In the Custom Tag Options box, click Edit Existing.
3. You can change any of the tag attributes. (This does not change the tag's assignment status; if it is selected for a device type, it does not move back to the IEC Tags list.)
4. Click Save to save changes, or click Save & Exit to save changes and close the screen.

Deleting a Custom Tag

You can delete any custom tag that is not associated with a device type.

If the tag is associated with a device type, you must first deselect the tag:

1. Change the option to Edit Existing and display the tag you want to delete.
2. Click Display Associated Device Types to display all device types that include this tag. Make a note of the device types.
3. Return to the Add/Edit Device Type screen. For each device type listed, deselect the tag that you want to delete.

Continue deleting the tag:

1. Open the Add/Edit Custom Tags screen.
2. In the Custom Tag Options box, click Delete Existing.
3. From the drop-down menu, choose the tag you want to delete.
4. Click Delete. At the Confirm prompt, click Yes.
5. Click Save to save the change, or click Save & Exit to save changes and close the screen.

Editing Tag Addresses

Use the Edit Address screen to edit the attributes of a single tag address. If a device type is locked, you cannot edit any of its tag addresses; they will be greyed out. A thorough discussion of IEC61850 tags and their construction is included in *Understanding Tags* on page 54 and *Logic Codes* on page 59.

NOTE: Case and order are critical in the tag address. Be careful to observe the exact address order. For address order, see *Logic Codes* on page 59. Also, be sure you use the correct case. For example, use **M** for register numbers in hexadecimal, and use **m** for register numbers in decimal.

To view the Edit Address screen, open the Define Device Type Tags tab, then click the Edit... field for the tag that you want to change. The Edit Address screen is different for real-time and alarm tags. The real-time tag table displays first, followed by alarm tag addresses.

Real-Time Tag Addresses

The following table describes the fields of the Edit Address screen for real-time tags. Detailed instructions are after the table.

Box Name	Field Name	Comments
Data Information box	Data Type: for display only	You can edit this field in the Add/Edit Custom Tags screen.
	Priority: High, Normal, or Low Logic Code:	You can edit this field either here or in the Add/Edit Tag screen.
	Logic Code: Select the logic code for this tag.	The logic code list depends on the Data Type for this tag. For more information about logic codes, see *Logic Codes* on page 59.
Device Information box	Display Registers in: hexadecimal/decimal	Click the radio button for the way you want to view register information.
	Module	Choose the type of module in which the tag is used. Used for Micrologic at this time.
	Register Type	Select the type of register that is to be written or read.
Number of Registers	Select the total number of registers for this address (1-10). Is Consecutive, although greyed out, is checked if the registers are consecutive (determined in the logic code).	Enables for editing the appropriate registers in the lines below.
Fixed Scale/Register Scale	Click the radio button for the correct type of scale.	A fixed scale is the actual value. of the scale. A register scale is the register address where the scale is held. The value will be scaled in this manner: Value x 10x where X = the scale. Scales can only be –10 to 10.

Box Name	Field Name	Comments
Conversion Factor	Enter the multiplier to convert the base units to the desired conversion.	Conversion factors are used for straight multiplication with the value. The conversion factor could also be changed in the Add/Edit Units screen (Settings > Select Units > Add/Edit Units. Conversion factors take this form: ####E##. For example, 123E-2 becomes 123x10-2 which becomes 1.23.
Register 1-4	Enter the register number.	Be aware of whether you chose hexadecimal or decimal. Use the same format here.
Bitmask for Register 1-4	For digital input/output tags: Set the bits to 1 or 0 to match the pattern for "True" in the device register.	When all bits match exactly the pattern in the register, the status is True. When any one bit does not match the pattern in the register, the status is False. Note: On PM8s and CM4s, there is a device-specific format, DIgIn and DigOut. In each case, you must first specify the indicator register (which becomes the first register). The second register will have the mask.
Invert Result	Check this box to invert.	Will turn False to True or vice versa; typically used for Normally Open or Normally Closed.

Onboard Alarm Tag Addresses

The following table describes the fields of the *Edit Address* screen for onboard alarm tags. Detailed instructions are after the table

Box Name	Field Name	Comments
Tag Information box	Tag name: for display only	This is the tag name, which cannot be changed.
Alarm Information box	File Number	This is the file number for the alarm file on the device.(Sepam has no file number; enter 0.)
	Module	Choose the type of module in which the tag is used. Used for Micrologic at this time.
	Unique ID	This unique identifier must be used to ensure that alarms will annunciate correctly. For CM4, PM8, and Micrologic, the unique ID must be decimal. For Sepam, the unique ID is the coil bit address that indicates the alarm; it must be in hexadecimal.
	Hexadecimal	Check this box if you want to display the ID in hexadecimal, rather than decimal.
	Has Unique Sub ID	Check if this tag has a unique sub-identifier (Micrologic, CM4000, and PM800 devices).
	Unique Sub ID	Enter the unique sub-identifier.

Reset Tag Addresses

NOTE: Once the tag is set up, writing a 1 to the tag will cause the "write" to occur.

Standard device types include some pre-defined resets. These pre-defined commands cause proprietary functions within the device. Do not edit these commands.

To add a custom reset that will operate by writing to a register, do the following:

1. From the Add/Edit Custom Tags screen, set the Group to Resets and the Data Type as Digital. Save the tag.
2. Add the new tag(s) to the appropriate device type(s).
3. From the Define Device Type Tags tab, locate the tag and click Edit.

The following table describes the fields of the Edit Address screen for reset tags. Detailed instructions are after the table.

Box Name	Field Name	Comments
Tag Information	Command Type	The Command Type and Command to Edit are already selected.
	Command to Edit	
Data Information box	Data Type: for display only	You can edit this field in the Add/Edit Custom Tags screen.
	Priority: High (default)	Cannot be edited.
	Logic Code: Select the logic code for this tag.	Choose the appropriate logic code for this tag. See *Logic Codes on page 59*.
Device Information box	Display Registers in: hexadecimal/decimal	Click the radio button for the way you want to view register information.
	Module	Choose the type of module in which the tag is used. Used for Micrologic at this time.
	Register Type	Select the type of register that is to be written or read.
Number of Registers	There is only one register for this address.	Enables for editing the appropriate registers in the lines below.
Fixed Scale/Register Scale	n/a	Not used for digital logic codes.
Conversion Factor	n/a	Not used for digital logic codes.
Register 1	Enter the register number.	Be aware of whether you chose hexadecimal or decimal. Use the same format here.
Bitmask for Register 1	For digital input/output tags: Set the bits to 1 or 0 to match the pattern for “True” in the device register.	When all bits match exactly the pattern in the register, the status is True. When any one bit does not match the pattern in the register, the status is False. Note: On PM8s and CM4s, there is a device-specific forma, DIgIn and DigOut. In each case, you must first specific the indicator register (which becomes the first register). The second register will have the mask.
Invert Result	n/a	Not used for resets.

Control Tag Addresses

NOTE: For a control with one command, once the tag is set up, writing a 1 to the tag will cause the “write” to occur. For a control with two commands that is either static or normally open, writing a 1 to the tag will cause the first command (ON) to occur; writing a 0 will cause the second (OFF) to occur. For a control with two commands that is normally closed, writing a 1 to the tag will cause the second command (OFF) to occur; writing a 0 will cause the first command (ON) to occur.

Standard device types include some pre-defined controls. For example, Operate (ENERGIZE). These pre-defined commands cause proprietary functions within the device. Do not edit these commands.

To add a custom control that will operate by writing to a register, do the following:

1. From the Add/Edit Custom Tags screen, set the Group to Controls and the Data Type as Digital. Save the tag.
2. Add the new tag(s) to the appropriate device type(s).
3. From the Define Device Type Tags tab, locate the tag and click Edit.

The following table describes the fields of the Edit Address screen for control tags.

Box Name	Field Name	Comments
Tag Information	Command Type	For commands that have an opposite (such as On and Off), choose Normally Open/Normally Closed or Static with Off Command. For commands with only one action, choose Static without Off Command.
	Command to Edit	If you are editing a command with two parts, use the Command to Edit drop-down menu to select the On Command.
Data Information box	Data Type: for display only	You can edit this field in the Add/Edit Custom Tags screen.
	Logic Code: Select the logic code for this tag.	Choose the appropriate logic code for this tag. See *Logic Codes on page 59*.
Device Information box	Display Registers in: hexadecimal/decimal	Click the radio button for the way you want to view register information.
	Module	Choose the type of module in which the tag is used. Used for Micrologic at this time.
	Register Type	Select the type of register that is to be written or read.
Number of Registers (1)	n/a	Enables for editing the appropriate registers in the lines below.
Fixed Scale/Register Scale	Click the radio button for the correct type of scale.	A fixed scale is the actual value. of the scale. A register scale is the register address where the scale is held. The value will be scaled in this manner: Value x 10x where X = the scale. Scales can only be –10 to 10.
Conversion Factor	n/a	Not used for digital controls.
Register 1	Enter the register number.	Be aware of whether you chose hexadecimal or decimal. Use the same format here.
Bitmask for Register 1	For digital input/output tags: Set the bits to 1 or 0 to match the pattern for “True” in the device register.	When all bits match exactly the pattern in the register, the status is True. When any one bit does not match the pattern in the register, the status is False. **Note**: On PM8s and CM4s, there is a device-specific forma, DIgIn and DigOut. In each case, you must first specific the indicator register (which becomes the first register). The second register will have the mask.
Invert Result	n/a	Not used for digital controls.

Editing Address Information

To edit address information for a real-time tag:

1. From the Define Device Type Tags tab, choose a device type (cannot be locked). From the Real Time Tags sub-tab, highlight the tag whose address you want to edit.
2. In the Edit Address column, click “Edit” for the address you want to edit.

   The Edit Address screen displays.
3. You can change any of the tag address attributes. See the preceding table for descriptions of each field.
4. Click *OK* to save changes and close the screen.

Adding a New Tag Address

You can also add a tag address, when none exists. As with editing addresses, click the Edit Address column for a tag; then follow instructions in the table above.

Creating Device Profiles

Use the Create Device Profiles screens to view and edit profiles for individual devices. Profiles are predefined for the standard devices; you will mostly use this feature to add third-party device profiles. Before you create profiles, you need to be sure that all of the tags and device types that you need are created (see *Defining Device Types and Tags* on page 27). Also make sure that you have added any new units/conversions and device type categories/subcategories that are needed.

In this section, you can learn about:

Enabling Waveforms

Viewing Device Profiles

Adding, Editing, or Deleting a Device Profile

Enabling Waveforms

On the Create Device Profiles tab, in the Onboard Alarm Tags sub-tab, there is a Waveform checkbox. Check the box for each alarm tag for which you want to be able to view waveforms. On the device, the alarm must also be set up for the waveform to be captured on event and stored in one of the device's data logs.

To acquire waveforms for Sepam, use the SFT manual. For PowerLogic devices, reference the PMCU help file.

As device information is polled and received by PowerLogic SCADA, the waveform becomes available for viewing. See *Changing the Display Name in an Alarm or Event Log* on page 115 for information on viewing waveforms in the runtime environment.

Viewing Device Profiles

The Create Device Profiles tab displays all of the tags that are included in each device type profile. It is the starting point for creating/editing device profiles for individual devices. Most of the data on this screen displays for information only; however, to enable waveforms, you need to check the Waveform box (see *Enabling Waveforms* on page 44 for more information). To begin making changes, click Add/Edit. The following table describes the fields on this tab. The tags listed assume that Advanced Properties has been checked. Not all elements appear on every sub-tab. Detailed instructions are after the table.

Field Name	Valid Entry	Comments
Tag Groups (left-hand pane)	Click a group to display the groups of tags that have been selected for the chosen device profile.	To associate tags and tag groups with a device type (thus creating a device profile), click Add/Edit.
Device Profile	Choose the device for which you want to view profile details.	Device Profiles are created on the Add/Edit Device Profile screen (click Add/Edit).
Add/Edit button	Click to display the Add/Edit Device Profile screen.	Use that screen to add device profiles and to associate PC-based alarms and trends.
Tag type sub-tabs	Click to display the selected tags for each type of tag: real-time, trend, PC-based alarm, onboard alarm, control, or reset.	Organized according to tag groups.
Tag Description	n/a	This is the tag name used when adding the tag.
IEC Tag Name	n/a	This is the IEC61850-compatible name created when the tag was added.

Field Name	Valid Entry	Comments
Category Type	n/a	These are real-time filters. They provide metadata to be used in future reporting.
Utility Type	n/a	
Statistical Type	n/a	
Quantity	n/a	
Categorization (PC Based and Onboard Alarm)		These are alarm filters. They can be used for filtering and sorting alarm data in the runtime environment. They also provide metadata to be used in future reporting.
Subcategorization (PC Based and Onboard Alarm)		
Alarm Type (PC Based and Onboard Alarm)		
Alarm Group (PC Based and Onboard Alarm)		
Alarm Level (PC Based and Onboard Alarm)		
Waveform (Onboard Alarm)	Check this box as part of the process of enabling waveform viewing in PowerLogic SCADA.	You must also set up the alarm and waveform capture in the onboard files of the device. Waveforms will then be viewable in the runtime environment.
Alarm On Text (Onboard Alarm)	Override default text.	When creating the onboard alarm tag, the default text was added. It displays in the Alarm Log to indicate the status of the alarm. You can edit the text here.
Alarm Off Text (Onboard Alarm)	Override default text.	**Note**: If no text was added when the tag was created to the Profile Editor, the default text is “Appearance” and “Disappearance.”

To view profile information, select the device profile from the drop-down menu. Use the tag tabs (real-time, trend, etc.) to view the tag groups included in this device profile.

To begin adding, editing, or deleting a profile, click Add/Edit.

Adding, Editing, or Deleting a Device Profile

Use the Add/Edit Device Profile screen to add, edit, or delete device profiles from the system. To view this screen, click Add/Edit from the Create Device Profiles tab. The following table describes the parts of the Add/Edit Device Profile screen. Detailed instructions are after the table.

Field Name	Valid Entries	Comments
Profile Options		
Create New		Click to add a new profile.
Create From		Click to add a profile based on an existing profile.

Field Name	Valid Entries	Comments
Edit Existing		Click to edit an existing, unlocked, profile.
Delete		Click to begin deleting an unlocked profile
Device Profile Name	Type or select the profile name: maximum 32 characters do not use \ / : * ? < > |	When creating a profile, type its name here. When editing or deleting a profile, choose its name.
Device Type	Choose the device type.	When creating a new profile, choose a device type. When editing or creating from a device profile, its device type displays.
Description	optional	This adds descriptive information for "tool tip" display. When you hover over the name of the profile on the Create Device Profiles tab, the description displays. It also displays with the profile name on the Selected Device Profiles sub-tab on the Set Up Project tab.
Real Time Tags Onboard, Control, and Reset Tags	Select tags or tag groups and click the right arrow to move them to the Selected box.	Included are all tags, both predefined and custom, that have been added to the system for the selected device type. Until you move a tag or tag group to the Selected box, it is not a part of the device profile.
Select All/Unselect	Click to check all/uncheck all of the tags listed in the box below	With a tag group selected in the Selected drop-down menu, the tags display
Tag Name	Check the boxes to individually select or unselect individual tags for PC-based alarms and for trends.	This is the tag name that was entered when the tag was created.
Selected Real Time Tags Selected Onboard, Control and Reset Tags	Click the right and left arrow to move tags into and out of the Selected box. Tags display by tag group, so you need to choose the correct group from the Selected Real Time Tags drop-down list.	These are the individual tags that are associated with the device profile. If a real-time tag is selected, and a check mark displays under PC Based Alarm Tag (for PC based alarms) or Trend Tag (for historical logging), the corresponding information will be created when the device is added to the PowerLogic SCADA project: PC based alarms are added to the Analog Alarms or Digital Alarms file; trends are added to the Trend Tags file. When you check "Trend Tag" for a tag, logging will automatically begin when the tag is added to the PowerLogic SCADA project. There are two different intervals for scanning trend tags. All selected tags are scanned every 15 minutes with FIFO storage of 12 months. For the following tags, there is an additional "short" scanning interval of 5-seconds, with FIFO storage of two weeks: Current A, Current B, Current C, Voltage A-B, Voltage B-C, Voltage C-A, Power Factor Total, Apparent Power Total, Reactive Power Total, Real Power Total, and Frequency. For instructions on changing the "short" scan interval settings, see *Trend Tag Scan Intervals* on page 102 You add alarm setpoints and delays in the PowerLogic SCADA project. See *Adding Setpoints and Delays* on page 92 for instructions. These checked real-time tags will also display for the profile on the Create Device Profiles tab. NOTE: When you select a large number of trend tags, it can cause the Trend Server to take a long time to start up. Be careful to use only the trend tags that you need. For the recommended tag limits in each system size, see the *Design Reference Guide*.

Adding a Device Profile

To add a profile:

1. Open the Create Device Profiles tab (from the Create Device Profiles tab, click *Add/Edit*).

2. In the Profile Options box, click Create New or Create From.

   If you are creating from another device profile, choose it from the Device Profile to Create From drop-down menu.
3. Type a unique Device Profile Name*:* maximum 32 characters; do not use \ / : * ? < > |
4. If you want to lock this profile, preventing anyone from editing it, check the Lock this Device Profile box. This action cannot be undone. If you wish to edit a locked profile, you must use the Create From option to add a new one, then delete the locked one.
5. Choose a Device Type from the drop-down menu.
6. Type an optional description. This description displays as a tool tip when you hover over the profile in the Create Device Profile tab.
7. From the Real-Time Tags list, highlight the first individual tag or tag group to be included in this profile. Click the right arrow button to move it to the Selected Real-Time Tags box. You must select and move tags or groups one at a time (no shift+click to select multiples).

   Note that, as you add tags, the Selected Real-Time Tags drop-down menu is populated with a list of their tag groups.
8. Select real-time tags for PC-based alarms or trends: Choose a tag group from the drop-down menu, then use one of two methods:

   Click Select All for PC-based alarm tags and/or trend tags. You can then individually de-select individual tags if necessary.

   NOTE: When you select a large number of trend tags, it can cause the Trend Server to take a long time to start up. To avoid this problem, be careful to use only the trend tags that you need. For the recommended tag limits in each system size, see the *Design Reference Guide*.

   See the table above for description of trend tag intervals.

Editing a Device Profile

Only unlocked profiles are available for editing.

To edit a profile:

1. Open the Create Device Profiles tab (from the Create Device Profiles tab, click Add/Edit).
2. In the Profile Options box, click Edit Existing.
3. From the drop-down menu, choose the profile you want to edit.
4. You can change any of the attributes that have been selected for this profile.
5. Click Save to save the change, or click Save & Exit to save changes and close the screen.

Deleting a Device Profile

You cannot delete standard profiles or custom profiles that have been associated with projects. To delete a custom profile that is associated with a project, you need to go to the Set Up Project tab.

To delete a profile:

1. Open the Create Device Profiles tab (from the Create Device Profiles tab, click Add/Edit).
2. In the Profile Options box, click the Delete Existing radio button.

3. From the drop-down menu, highlight the profile you want to delete.
4. Click Delete.
5. At the Confirm prompt, click Yes.
6. Click Save to delete the profile, or click Save & Exit to delete the profile and close the screen.

Setting Up Projects in the Profile Editor

The Set Up Projects tab and screen are used to create separate projects for each customer or installation. This tab makes it easy to select only the devices that are used at that site. Project data is exported to PowerLogic SCADA for use in the Device Creation Wizard.

To access these screens, click the Set Up Projects tab on the main Profile Editor screen.

In this section, you can learn about:

Project Screens and Workflow

Viewing Project Data and Exporting the Project

Adding, Editing, or Deleting a Project

Exporting a Project

Editing and Deleting Information in a PowerLogic SCADA Project

Customizing Tag Names

Project Screens and Workflow

This section contains two screens:

The **Set Up Project tab** has two sub-tabs:

- The Selected Device Profiles tab displays all of profiles that are included in the Project that is displayed in the drop-down menu.
- The Customize Tag Names sub-tab provides a means of customizing individual tag names.

On either tab, you can click Export Project to create an .XML file that contains all of the project data necessary for use in the Profile Wizard. If PowerLogic SCADA is installed and the corresponding PowerLogic SCADA project has been created, this also copies the file that is used by the Device Creation Wizard to the PowerLogic SCADA project.

On the **Add/Edit Project screen**, you can add, edit, or delete projects.

Typical Workflow

To create a project file, you must first have established tags, device types, and device profiles. Additionally, you need to set up at least one base unit/conversion template. After these files are created, follow these steps:

1. Open the Add/Edit Project screen.
2. Add a new project, or copy and edit an existing project.
3. Select the device profiles that you want to use for this installation.
4. Save the project and close the Add/Edit Project screen.
5. To customize tag names: From the Set Up Project tab, click the Customize Tag Names sub-tab. Change the name of any tag. For example, the customer might need “Current A” to read “Current Phase A.” The customized tag name will be used in all device profiles in the project for which you have created the customized tag. For this change to be in the PowerLogic SCADA project: you need to delete the device profile from the PowerLogic SCADA project and then re-export it.
6. To refresh tags: From the Set Up Project tab, click the Selected Device Profiles sub-tab. Click the Refresh Tags button for any profile. You are prompted to confirm that you want to update changes you have made to this tag for the selected profile. For this change to be in the PowerLogic SCADA project: you need to remove the device profile from the PowerLogic SCADA project and then re-add it.

7. Click Export Project to create an Equipment.Profiles file of all of the profiles included in the project. To view Equipment.Profiles, click the folder button, to the right of the Export button:

Viewing Project Data and Exporting the Project

Use the Set Up Project tab to begin adding, editing, or deleting projects. A project includes all of the tags that belong to the device profiles that you have created and added to the project. From this screen you also export individual projects to the .XML file format that can be added to PowerLogic SCADA (using the Profile Wizard).

This screen includes two tabs:

- Selected Device Profiles: used to view all of the profiles that are included in each project in the system (Selected Device Profiles sub-tab); profiles are listed with their descriptions
- Customize Tag Names: used to customize tag names (for example, instead of Current A, you might need to use Current Phase A) within a single project (Customize Tag Names sub-tab)

To add or edit project information, click Add/Edit. The Add/Edit Project screen displays. See *Adding, Editing, or Deleting a Project* on page 50 for more information.

NOTE: Below the Project name field, the (optional) project description displays. This is for information only. This field is created when you add a project.

To export a project, select it from the Project drop-down menu, then click Export. See *Customizing Tag Names* on page 53 for more information.

To specify a tag to be used for onboard waveform captures, click the Onboard Alarm Tags sub-tab. Locate the tag and check the Waveform box.

To view the most recently exported project, click the folder button to the right of the Export button:

Adding, Editing, or Deleting a Project

Use the Add/Edit Project screen to begin adding, editing, or deleting projects. A project includes all of the tags that belong to the device profiles that you have created and added to the project. From this screen you also export individual projects to the format that can be added to PowerLogic SCADA (using the Profile Wizard).

Adding a Project

To add a project:

1. First ensure that you have set up the tags, device types, and device profiles that you want to include. Also, add at least one unit template.
2. In the Project Options box, click Create New or Create From.

   If you are creating from another project, choose it from the Project to Create From drop-down list.
3. Type a Project Name: maximum 32 characters; name must be alpha-numeric only, must begin with an alpha character, no punctuation allowed. Do not use:

   \ / : * ? < > |

4. To view a list of projects that have already been added to PowerLogic SCADA, click the Display Projects button:

   A list displays with the projects that have been added to PowerLogic SCADA (greyed out if there are no projects yet or if the Profile Editor is not on the same computer as the server). To open a project for editing, select it and click OK.
5. If you want to lock the project, preventing anyone from editing it, check the Lock this Project box. This action cannot be undone. If you wish to edit a locked project, you must use the Create From feature to add a new one, then delete the locked one.
6. Type a Description for the project. This description displays as a tool tip when you hover over the project name on the main Set Up Project tab.
7. Select a Unit Template from the drop-down list. Unit templates are created on the Units screens. See *Setting Up Engineering Templates and Selecting Conversions* on page 23 for instructions on creating templates.

   To add a new unit template, click Set Up Eng. Unit Templates. The Set Up Engineering Unit Templates page displays. See *Adding or Editing a Base Engineering Unit or Conversion* on page 25 for help.
8. From the Device Profiles box, select the profiles you want to include in this project. Highlight one at a time, then click the right arrow to move it to the Selected Profiles box.
9. When all profiles are added, click Save to save the change, or click Save & Exit to save changes and close the screen.

Editing a Project

You can only edit projects that are unlocked.

To edit a project:

1. Open the Add/Edit Project tab (from the Set Up Project tab, click Add/Edit).
2. In the Project Options box: Select the project to be edited from the drop-down list, then click Edit Existing.
3. You can change any attribute of the project.
4. Click Save to save the change, or click Save & Exit to save changes and close the screen.

Deleting a Project

You can only delete unlocked projects.

To delete a project:

1. Open the Add/Edit Project tab (from the Set Up Project tab, click Add/Edit).
2. In the Project Options box: Select the project to be deleted from the drop-down list, then click Delete Existing.
3. Click Delete.

Exporting a Project

After your project is complete, you will export it to the PowerLogic SCADA project. If the Profile Editor is on the same computer with PowerLogic SCADA, and if you have

created a matching project in the PowerLogic SCADA project, this process will copy all project data—device tags, device types, and device profiles—into the PowerLogic SCADA project. If the Profile Editor is not on a computer with PowerLogic SCADA, you need to manually move the exported file to the PowerLogic SCADA server. See **Moving Files if the Profile Editor is not with the Server**, below.

Before you export

If you are exporting a project for the first time to the PowerLogic SCADA project, you need to create a matching project in PowerLogic SCADA. To do this:

1. Launch the instance of PowerLogic SCADA to which you will be exporting data: Click Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Explorer.
2. From the Citect Explorer window, click File > New Project.
3. In the New Project Window, add the project, ensuring that the name matches exactly the one in the Device Profile. Be sure that Template Resolution is SXGA.

   If you have questions about any of the fields, click Help.

Exporting

To export a Profile Editor project to the PowerLogic SCADA project:

1. Open the Set Up Project tab.
2. From the drop-down list, select the project to be exported.
3. Click Export Project.

   NOTE: If you have added custom tags to devices, but the tag addressing is incomplete, a message displays with the name(s) of the device profile(s) that contain the tag(s). Return to the Define Device Type Tags tab. Locate any tags for which “Edit...” is red. Click Edit... to open the Edit Address screen. Make the necessary changes. Then, from the Set Up Project tab, refresh the tags for those profiles. Then try exporting again.

   A progress bar displays while the various profiles are saved, The resulting files are exported to these locations in the Profile Editor (assuming that you accepted the default locations during installation):

   — Each Project file, used by the Profile Editor, is stored in Documents and Settings\All Users\Application Data\Schneider Electric\Profile Editor\v1.0\Projects (for Vista operating systems, it is ProgramData\Schneider Electric\Profile Editor\v1.0\Projects).

   — Each Profile Wizard profile file is stored in Documents and Settings\All Users\Application Data\Schneider Electric\Profile Editor\v1.0\WizardProfiles\[project name] (for Vista operating systems, it is ProgramData\Schneider Electric\Profile Editor\v1.0\WizardProfiles\[project name]). A single file for each included profile.

   — The Equipment.Profiles file (contains all of the Profile Wizard profile information and the base profile information used by the Profile Wizard) is stored in Program Files\Schneider Electric\Profile Editor\v1.0\WizardProfiles\[project name]\Profile Wizard.

   Both the Application Data directory (on Windows XP) and the ProgramData directory (on Vista) are hidden folders by default.

   In PowerLogic SCADA, files are located in the following folders:

   — DeviceProfiles contains .XML files for every profile (these are used by the Profile Editor).

— DeviceTypes contains .XML files for all device types (these are used by the Profile Editor).
— Projects contains all .XML files for all projects (these are used by the Profile Editor).
— DeviceWizardProfiles contains the exported device profiles and equipment profiles files, organized by project (these are used by the Profile Wizard).

4. On the Project Editor window, use the Profile Editor to add device information.

Moving Files if the Profile Editor is not with the Server (Reusing a Project)

When the Profile Editor is not on the same computer as the PowerLogic SCADA server, you need to move the export file to the server computer. Follow these steps:

1. Export the project from the Profile Editor.
2. Click the folder icon beside the Export link:

3. Copy the file (Equipment.profiles) that displays and move it to a portable drive.
4. On the PowerLogic SCADA server computer, paste Equipment.profiles in this location:

   [Drive Letter]:\Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\User\[Project] (for Vista operating systems: [Drive Letter]:\ProgramData\Schneider Electric\PowerLogic SCADA 7.10\User\[Project])

   where:

   [Drive Letter]: is the drive on which you installed the PowerLogic SCADA server

   the Application Data and ProgramData folders cannot be hidden (set the folder view for “view hidden folders”)

   [Project] is the name of the project you are creating; you must have already added this project to PowerLogic SCADA (see Before You Export, above).
5. From the Citect Project Editor, click Tools > Profile Wizard to begin adding device information to the PowerLogic SCADA project.

Editing and Deleting Information in a PowerLogic SCADA Project

After you have exported a project to a PowerLogic SCADA project, you still need to use the Device Creation Wizard to add system information to a PowerLogic SCADA project. See *Before You Use the Profile Wizard* on page 68 for information about this process.

Customizing Tag Names

From the Set Up Project tab, click the Customize Tag Names sub-tab. You can add a custom name for any tag in the system, predefined and custom tags. The customized name will be used anywhere the original name would be used, but only for the project that is selected in the drop-down menu. When you use the Export option, it will be used by the Profile Wizard.

Understanding Tags

PowerLogic SCADA includes a variety of tag types: real-time, alarm, and trend. Most of the tags that you will need are already added. However, you can add custom tags to suit special needs. This section describes how tags are constructed and provides further specific information about the construction of format codes, logic codes, and addresses.

In this section, you can learn about:

IEC61850 Tag Construction

Defining an Enumeration

Format Code Definitions

Logic Codes

IEC61850 Tag Construction

IEC61850 tags have a great deal of flexibility, which allows them to specify the way that functions are implemented in devices. The IEC61850 tag was developed for medium-voltage and high-voltage applications, such as monitoring, control, and substation automation.

Some of our devices include data and functionality that are not yet covered by IEC61850. For these devices, the general IEC61850 formatting was followed when creating tags.

If you are writing Cicode (see *Using Cicode to Customize a Project* on page 99). You will need to know the IEC61850 tag name that you added to the device profile for that device. You can print the .csv file to view tag names (see *Printing the .CSV File* on page 34). Apart from that, you would only need to add tags if you are installing a third-party device that is not standard to PowerLogic SCADA. If you do need to add tags, create any category you wish, and follow the format shown below.

Tags shall be no longer than 29 characters. Use a backslash as a separator between tag parts. Tags are constructed in this manner:

EquipmentName\Logical_Node\Data Object\Data Attribute (may have more than one)

The following table lists the main categories for the common logical nodes. After the table, the most commonly used category (Mxxx: metering and measurement) is described.

Category Name	Description
Axxx	automatic control; e.g., ATCC (tap changer), AVCO (voltage control)
Cxxx	supervisory control; e.g., CILO (interlocking), CSWI (switch control)
Gxxx	generic functions; e.g., GGIO (generic I/O)
Ixxx	interfacing/archiving; e.g., IARC (archive), IHMI (HMI)
Lxxx	system logical nodes; e.g., LLNO (common), LPHD (physical device)
Mxxx	metering and measurement; e.g., MMXU (measurement), MMTR (metering), MSTA (metering statistics), MSQI (sequence and imbalance), MHAI (harmonics and interharmonics)
Pxxx	protection; e.g., PDIF (differential), PIOC (instantaneous overcurrent or rate of rise.), PDIS (distance), PTOV (time-overvoltage)
Rxxx	protection related; e.g., RREC (auto reclosing), RDRE (disturbance)
Sxxx	sensors, monitoring; e.g., SARC (arcs), SPDC (partial discharge)
Txxx	instrument transformer; e.g., TCTR (current), TVTR (voltage)

Category Name	Description
Xxxx	switchgear; e.g., XCBR (circuit breaker), XCSW (switch)
Zxxx	other equipment; e.g., ZCAP (cap control), ZMOT (motor)

The following example illustrates the IEC61850 tag for current A:

EquipmentName\MMXU1\A\PhsA

where:

M = the category

MXU = measurement of currents, voltages, power, and impedances

1 = the instance (there could be multiple MMXU tags)

A = the data object, current

PhsA = the attribute that further defines the data object, phase A

All of the tags that are currently used in the system can be viewed from the Profile Editor, Define Device Type Tags tab. Click Settings > Display Advanced Properties to display the full tag names.

Defining an Enumeration

An enumeration is a single value (0-15) that is used to define a condition that is determined by multiple-bit input. You will add enumerations to handle scenarios that are more complicated than simply true-false, to allow for dynamic contingencies. For example, when you need to use multiple bits to describe the position of a circuit breaker, you might do the following:

Bit *y* (closed) | Bit x (open). Note that the least significant bit is register 1.

Bit x | Bit y	Status	Circuit Breaker Position	Returned Value
0 | 0	Indeterminate	Circuit breaker is neither open nor closed	0
0 | 1	Open	Circuit breaker is open.	1
1 | 0	Closed	Circuit breaker is closed.	2
1 | 1	Error	Circuit breaker is reporting both open and closed condition. Possible device/wiring error	3

Using the enumerated status, we place the register and bitmask for the open position in register 1 (least significant) and the register and bitmask for the closed position in register 2 (most significant).

Using Special Tags to Control Circuit Breaker Status

When you want to include a device that does not have a pre-defined device profile (such as a third-party circuit breaker), you must identify the registers that the device uses for the operations you want, then choose the correct tags and tag addresses to write to these registers. Finally, when creating the one-line on the graphics page, you will choose the appropriate genie:

1. Determine the device registers used for the open and close operations on the circuit breaker.

2. In the Profile Editor, choose the tag needed for each operation.
3. Ensure that tag address references the correct action and register(s). See *Editing Tag Addresses* on page 40 for instructions on editing the address,
4. When adding a genie for the circuit breaker on the graphics page, choose from the default library (see *Appendix 1: Default Genie Library* on page 129), or create a custom genie (see *Creating and Editing Genies* on page 90).

Format Code Definitions

The address field is part of the tag. It includes a variety of attributes, some of which are required, and some optional. The following tables list the attributes, whether they are required, and their possible modifiers. All parts of a tag are case sensitive. The order of the fields is fixed; and all fields are separated by semi-colons. See *Logic Codes* on page 59 for templates of constructed tags.

Real-Time Format Code Definitions

Attributes	Modifiers	Comments
T (type) Required	SS = single status DS = double status enumeration ST = string UT = UTC time MV = measured value (float) CM = complex measured value (float) BC = binary counter (integer)	Temporarily, this may return a string; when PowerLogic SCADA is upgraded to handle large integers, this will change.
D (module—Micrologic devices)	B = BCM P = PM M = MM C = CCM	
M/m/S/s/C/c/I/i (register type)	M = holding registers in hexadecimal m = holding registers in decimal S = input coil (status register) in hexadecimal s = input coil (status register) in decimal C = output coil (writable only) in hexadecimal c = output coil (writable only) in decimal I = input register (read only) in hexadecimal i = input register (read only) in decimal	
Register Number Modifiers (register number from 1–4)	u## = ## registers are unsigned, ## is a decimal s## = ## registers are signed; ## is a decimal	After the modifier, there may be a number indicating scaling factor. See "V," below in this table. Used for conversion to SI units, this number will be: RegisterValue x scale For SS and DS: there must be a 1U default; the modifier will be a bitmask: - The mask must use hex only, 16 bits/register - Attach the ones, then the zero mask, to the register; if you only have ones masks, just attach them - Only one register cases can be inverted. Add **:I** after the masks for inversion.
N (scale)	numerical entries; range is -10 to 10	N defines a constant scale; the logic code knows how to use it.
R (scale register)	the register number in decimal	R defines the holding register where the scale is held; the logic code knows how to use it.
E (priority)	single digit: 1, 2, or 3; default 2 is used if this is not included (1 = high, 2 = normal, 3 = low)	Defines the priority PowerLogic SCADA uses in processing data.
V (conversion factor)	Use scientific notation without the decimal.	Examples: 354E-3 = 0.354 354E1 = 3540 Will be multiplied before the value is returned.

Attributes	Modifiers	Comments
L:P (logic code) Required	The number that is used comes from the Logic Codes table.	L:P is the logic code for PowerLogic. Other codes may follow, such a L:I for ION. For logic code descriptions, see *Logic Codes* on page 59

Alarm Format Code Definitions

Attributes	Modifiers	Comments
T (type) Required	ALM = alarm	
D (module—optional for Micrologic devices)	B = BCM P = PM M = MM C = CCM	BCM is straight addressing, and therefore, optional.
F (file) Required	File number will be in decimal, up to 5 digits	
Q (unique ID) Required	Unique ID will be in decimal.	This number can be huge.

Control Format Code Definitions Rules of Operation

These rules are true for predefined and custom codes:

Address structure	Result
C:N;(action)	If 1, perform action. If 0, undefined.
C:N;(action1);(action2) C:NO;(action1);(action2)	If 1, perform action1. If 0, perform action 2.
C:NC;(action1);(action2)	If 1, perform action2. If 0, perform action1

Predefined Control Format Codes

Attributes	Modifiers	Comments
C (command) Required	NO = normally open NC = normally closed N = normal operation	Normal operation does not have a closed/open status.
OPERATE (command word)	n/a	Two required for NO and NC.

Predefined Reset Format Codes

Attributes	Modifiers	Comments
Reset (command word)	n/a	Entering a one to this tag causes the reset to take place.

Custom Control and Reset Format Codes

Attributes	Modifiers	Comments
C (command) Required	NO = normally open NC = normal closed N = normal operation	Normal operation does not have a closed/open status.
Followed by one or two entire “write” addresses; used only for logic codes 101, 102, 103. For logic code descriptions, see *Logic Codes* on page 59. Write Address format: T:SS;m:##:#;L.:P:101 Example: C:NO;T:SS;m:1234:1;L:P101;T:SS;m:3456:1;L:P101		

Logic Codes

Logic codes tell the SCADA program how to mathematically operate on the values in device registers to give users the desired values. The following table lists each logic code with its related information.

Logic Code	IEC Type	PowerLogic SCADA Data Type	Register Definition	Scaled Required? (register or fixed)	Device Specific?	Notes
READS						
Invalid Block (L:P:0)	IB	LONG	Up to 1,000 sequential registers	No	Generic only	Defines invalid blocks of memory in the device. The driver does not include these registers in block reads.
Date / Time (L:P:1) (3 register)	UT	LONG	3 sequential registers	No	Generic – if it fits	**Register N**: High byte = Month 1–12 Low byte = Day 1–31 **Register N+1**: High byte = Year 0–199 (+1900) Low byte = Hour 0–23 **Register N+2**: High byte = minutes 0–59 Low byte = seconds 0–59
Date / Time (L:P:2) (6 register)	UT	LONG	6 sequential registers	No	Generic - if it fits	**Register N**: Seconds 0–59 **Register N+1**: Minutes 0–59 **Register N+2**: Hours 0–23 **Register N+3**: Day 1–31 **Register N+4**: Month 1–12 **Register N+5**: Year 0–199 (+1900)
Date / Time (L:P:3) (3 or 4 register -- Circuit Monitor/ Power Meter)	UT	LONG	3 or 4 sequential registers	No	CM/PM	**Register N**: High byte = Month 1–12, Low byte = Day 1–31 **Register N+1**: High byte = Year 0–199 (+1900) Low byte = Hour 0–23 **Register N+2**: High byte = minutes 0–59 Low byte = seconds 0–59 **Register N+3**: msec = 0–999 (unused)
Date / Time (L:P:4) (3 or 4 registers SEPAM)	UT	LONG	3 or 4 sequential registers	No	SEPAM	**Register N**: Bits 0–6 = Year: 0 –70 (2000– 2070) 71 – 99 (1971–1999) **Register N+1**: Bits 8-11 = Month Bits 0-4 = Day **Register N+2**: Bits 8-12 = Hour Bits 0-5 = Minutes **Register N+3**: msec = 0-59,999 (seconds are ms/1000)
Date/Time (L:P:5) 3-register Micrologic	UT	LONG	3 sequential registers	No	micro	**Register N**: High byte = Month 1–12, Low byte = Day 1–31 **Register N+1**: High byte = Year 0–69 (+2000), Year 70–99 (+1900) Low byte = Hour 0–23 **Register N+2**:

Logic Code	IEC Type	PowerLogic SCADA Data Type	Register Definition	Scaled Required? (register or fixed)	Device Specific?	Notes
						High byte = minutes 0–59 Low byte = seconds 0–59
Date/Time (L:P:6) 4-register Micrologic	UT	LONG	4 sequential registers	No	micro	**Register N**: High byte = Month 1–12, Low byte = Day 1–31 **Register N+1**: High byte = Year 0–69 (+2000), Year 70–99 (+1900) Low byte = Hour 0–23 **Register N+2**: High byte = minutes 0–59 Low byte = seconds 0–59 **Register N+3**: msec = 0–999 (unused)
Date/Time (L:P:7) 2-register Argos	UT	LONG	2 sequential registers	No	Argos	The number of seconds since 01/01/2000 (00:00:00) register 1 = MSB register 2 = LSB
Modulo 10k (L:P:10)	BC	STRING	Up to 4 registers	No	generic	Result is a string representation. Range is 0 to 9,999,999,999,999,999 Each register has a range of 0 to 9,999 Result is: – R4*10,000^3 + R3*10,000^2 + R2*10,000 + R1
Modulo 10k Val (L:P:11)	BC	REAL	Up to 4 registers	No	generic	Result is a string representation. Range is 0 to 9,999,999,999,999,999 Each register has a range of 0 to 9,999 Result is: – R4*10,000^3 + R3*10,000^2 + R2*10,000 + R1
Modulo 10k Energy (L:P:12)	BC	STRING	Up to 4 registers	No	generic	Result is a string representation. Range is 0 to 9,999,999,999,999.9 Each register has a range of 0 to 9,999 Result is – (R4*10,000^3 + R3*10,000^2 + R2*10,000 + R1)/1000
Modulo 10k Energy Val (L:P:13)	BC	REAL	Up to 4 registers	No	generic	Result is a string representation. Range is 0 to 9,999,999,999,999.9 Each register has a range of 0 to 9,999 Result is – (R4*10,000^3 + R3*10,000^2 + R2*10,000 + R1)/1000
PL Digital Input SS (L:P:20)	SS	LONG	2 registers	No	CM/PM	First register (100–199 inclusive) indicates that this is a digital input register. Second register is masked to test for either one 1 or one 0. Result is: 0 = off and 1 = on. This result can be inverted.
PL Digital Input DS (L:P:21)	DS	LONG	2 registers	No	CM/PM	Same as PL Digital Input SS except: Result is 0 = intermediate, 1 = off, 2 = on, 3 = bad-state. Inversion will invert only off and on states.

Logic Code	IEC Type	PowerLogic SCADA Data Type	Register Definition	Scaled Required? (register or fixed)	Device Specific?	Notes
PL Digital Input TF (L:P:22)	SS	DIGITAL	2 registers	No	CM/PM	Same as PL Digital Input SS except: Result is: 0 = false and 1 = true. This result can be inverted.
PL Digital Output SS (L:P:23)	SS	LONG	2 registers	No	CM/PM	First register (200–299 inclusive) indicates that this is a digital output register. Second register is masked to test for either one 1 or one 0. Result is: 0 = off and 1 = on. This result can be inverted.
PL Digital Output DS (L:P:24)	DS	LONG	2 registers	No	CM/PM	Same as PL Digital Output SS, except: Result is: 0 = intermediate, 1 = off, 2 = on, 3 = bad-state. Inversion will invert only off and on states.
PL Digital Output TF (L:P:25)	SS	DIGITAL	2 registers	No	CM/PM	Same as PL Digital Output SS except: Result is: 0 = false and 1 = true. This result can be inverted.
Status SS (L:P:26)	SS	LONG	Up to 4 registers	No	Generic	Each register is compared to a ones' mask. Optionally it can be compared to a zeros' mask. (Use the Edit Address screen in the Profile Editor to create masks for the user.) Result is: 0 = off and 1 = on. If there is only one register, the result can be inverted.
Status OR SS (L:P:226)	SS	LONG	2 to 4 registers	No	Generic	Each register is compared to a ones' mask. These results are OR'ed together. Optionally, it can be compared to a zeros' mask. (Use the Edit Address screen in the Profile Editor to create masks for the user.) Result is: 0 = off and 1 = on.
Status DS (L:P:27)	DS	LONG	Up to 4 registers	No	Generic	Same as Status SS except: Result is: 0 = intermediate, 1 = off, 2 = on, 3 = bad-state. Inversion will invert only off and on states.
Status OR DS (L:P:227)	DS	LONG	2 to 4 registers	No	Generic	Same as Status OR SS except: Result is: 0 = intermediate, 1 = off, 2 = on, 3 = bad-state.
Status TF (L:P:28)	SS	DIGITAL	Up to 4 registers	No	Generic	Same as Status SS except: Result is: 0 = false and 1 = true. This result can be inverted.
Status OR TF (L:P:228)	SS	DIGITAL	2 to 4 registers	No	Generic	Same as Status OR SS except: Result is: 0 = false and 1 = true.
Status Int (L:P:29)	BC	LONG	1 register	No	CM/PM	One register is bit anded with one mask. The result will be an integer that can be used to choose the appropriate enumeration.
Status Enumeration (L:P:229)	EN	LONG	1 to 4 registers	No	Generic	Each register is compared to a ones' mask. Optionally it can be compared to a zeros' mask. (Use the Edit Address screen in the Profile Editor to create masks for the user.) Result is a combination of the results for each register, using this formula: result for register 1 * 2^0 + result for register 2 * 2^1 + result for register 3 * 2^2 + result for register 4 * 2^3

Logic Code	IEC Type	PowerLogic SCADA Data Type	Register Definition	Scaled Required? (register or fixed)	Device Specific?	Notes
PL Analog Input (L:P:30)	MV/CM	REAL	3 registers	No	CM/PM	First register (300–399 inclusive) indicates that this is a analog input register. Second register is treated as a signed value. Third register can contain a value from –3 to 3 and will be used to scale the second register (R2*10^R3).
PL Analog Output (L:P:31)	MV/CM	REAL	2 registers	No	CM/PM	First register (400–499 inclusive) indicates that this is a analog output register. Second register is treated as a signed value.
Scaled Register Signed (L:P:32)	MV/CM	REAL	1 or 2 registers	Either (optional)	Generic	For a single register: treated as a signed value from –32,767 to +32,767. (-32768 will result in a NA) For two registers: the registers will be concatenated together, the first register filling bits 16–32 and the second register filling bits 0–15. Values will range from –2,147,483,648 to –2,147,483,647. Values can be scaled using a fixed scale or a scale register.
Scaled Register Unsigned (L:P:33)	MV/CM	REAL	1 to 4 registers	Either (optional)	Generic	For a single register: treated as an unsigned value from 0 to 65,535. For two registers: the registers will be concatenated together, the first register filling bits 16–32 and the second register filling bits 0–15. Values will range from 0 to 4,294,967,295. Values can be scaled using a fixed scale or a scale register
Scaled Register Signed No NA (L:P:34)	MV/CM	REAL	1 or 2 registers	Either (optional)	Generic	Same as Scaled Register except that a single register with value -32768 is acceptable and will be reported as such.
Scaled Register Signed SEPAM A (L:P:35)	MV/CM	REAL	2 registers	Either (optional)	Generic	Same as Scaled Register except that 0xFFFFFFFF or 0x00007FFF will be NA.
Scaled Register Signed SEPAM B (L:P:36)	MV/CM	REAL	2 registers	Either (optional)	Generic	Same as Scaled Register except that 0xFFFFFFFF will be NA.
IEEE 32 Real (L:P:37)	MV/CM	REAL	2 sequential registers	No	Generic	Uses the IEEE standard for floating-point arithmetic (IEEE 754); register 1 is MSB, register 2 is LSB
Scaled Register Signed SEPAM 2000 Format B (L:P:38)	MV/CM	REAL	1 register	Either (optional)	Generic	For a single register: treated as a signed value from –32,767 to +32,767: From the value of the unsigned register, subtract 32768; then apply the scale. 0000 or FFFF will be NA.
PL String (L:P:39)	ST	STRING	1 to 10 sequential registers	No	Generic	Each register can represent up to two ASCII characters.
Sum Registers (L:P:40)	MV/CM	REAL	1 to 4 registers	Either (required)	Generic	Result is: R1 + … + Rn * 10^scale

Logic Code	IEC Type	PowerLogic SCADA Data Type	Register Definition	Scaled Required? (register or fixed)	Device Specific?	Notes
Divide Registers (L:P:41)	MV/CM	REAL	3 registers	Either (required)	Generic	Result is: R1/R2 * R3 * 10^scale If R2 is zero, result will be #COM
Multiply Registers (L:P:42)	MV/CM	REAL	1 to 4 registers	Either (required)	Generic	Result is: R1 * … * Rn * 10^scale
Average Registers (L:P:43)	MV/CM	REAL	1 to 4 registers	Either (required)	Generic	Result is: Avg(R1 … Rn) * 10^scale
Average Registers WF (L:P:44)	MV/CM	REAL	2 to 4 registers	Either (required)	Generic	Result is: Avg(R1 … Rn-1) * Rn * 10^scale
Sum with Scale (L:P:45)	MV/CM	REAL	2 registers	Either (required)	CM/PM	Result is: (R1 * 10^scale) + R2
Sum with Scale Unsigned (L:P:46)	MV/CM	REAL	2 registers	Either (required)	CM/PM	Result is same as above, except unsigned.
Vector Math (L:P:47)	MV/CM	REAL	2 registers	Either (required)	Generic	Result is: sqrt (R1^2 + R2^2) x scale
Vector Math IEEE (L:P:48)	MV/CM	REAL	4 registers	Either (required)	Generic	Result is: sqrt ([R1 R2]^2 + [R3 R4]^2) x scale where [ ] indicates IEEE32 representation
CM4 Power Factor IEEE (L:P:50)	MV/CM	REAL	1 register	No	CM4	Returns the IEEE power factor.
PM8 Power Factor IEEE (L:P:51)	MV/CM	REAL	1 register	No	PM8	Returns the IEEE power factor (converted from IEC mode as necessary). The device may be in IEEE or IEC mode if the device firmware version is 11.6 or higher. If the device firmware version is below 11.6, IEC mode is not supported.
SP40 Power Factor IEEE (L:P:52)	MV/CM	REAL	1 register	No	SEPAM 40	Returns the IEEE power factor (converted from IEC mode).
ML Power Factor IEEE (L:P:53)	MV/CM	REAL	2 registers	No	ML	Returns the IEEE power factor (converted from IEC mode as necessary). The second input register must be the associated Reactive Power for the Power Factor requested.
Generic Power Factor (L:P:54)	MV/CM	REAL	2 registers	No	Generic	R2/sqrt (R2^2 + R1^2) where: R2 = real power R1 = reactive power

Logic Code	IEC Type	PowerLogic SCADA Data Type	Register Definition	Scaled Required? (register or fixed)	Device Specific?	Notes
Generic Power Factor - IEEE32 variation (L:P:55)	MV/CM	READ	4 registers	No	Generic	[R3 R4]/ sqrt ([R3 R4]^2 + [R1 R2]^2) where: R3 = real power IEEE32 MSR R4 = real power IEEE32 LSR R1 = reactive power IEEE32 MSR R2 = reactive power IEEE32 LSR
SP2000 Power Factor IEEE (L:P:56)	MV/CM	REAL	1 register	No	SEPAM 2000	Returns the IEEE power factor (converted from IEC mode).
WRITES (these are write-only; see below for Read/Write codes) NOTE: If the device is capable of preventing (blocking) writes to its registers, verify that the "block" feature is disabled before you implement the write.						
Status Write Register (L:P:101)	SS	LONG	1 register	No	Generic	If you input 1 to this tag it will write the **MASK** value to the register.
Status Write Register AND (L:P:102)	SS	LONG	1 register	No	Generic	If you input 1 to this tag it will read the register and **AND** the **MASK** with the register (This puts a 0 whereever there is a 1 in the mask and leaves the rest alone).
Status Write Register OR (L:P:103)	SS	LONG	1 register	No	Generic	If you input 1 to this tag it will read the register and **OR** the **MASK** with the register (This puts a 1 whereever there is a 1 in the mask and leaves the rest alone).
Write Register Unsigned (L:P:110)	MV/CM	REAL	1 register	Either	Generic	This will take the input value read in and divide out the scale factor and the conversion factor. It will then round to the nearest whole number and if it is a value from 0 to 65535 it will put this value in the register.
Write Register Signed (L:P:111)	MV/CM	REAL	1 register	Either	Generic	This will take the input value read in and divide out the scale factor and the conversion factor. It will then round to the nearest whole number and convert the signed value to an unsigned value from 0 to 65535. It will put this value in the register.
READ/WRITES						
Read/Write Holding Register (L:P:120)	MV/CM	LONG	1 register	No	Generic	You can write any value from 0 to 65535 and read an unsigned value from the same register.
Read/Write Coil Register (L:P:121)	SS	DIGITAL	1 register	No	Generic	You can write 0 or 1 and read a value from the same register.

Using Block Writes

Block writes represent blocks of registers that are updated in a single write operation. There are two types of block writes:

- **fixed**: fully specified and compiled before run time. Writing the value of '1' to such a variable tag causes the specified fixed values to be written to the specified registers.
- **variable**: specified on the fly. The registers and the values to be written are not fixed; they are specified during run time by the user.

Fixed block writes have the following format:

`T:BWF;[D:{B|C|M|P};]S:<start_register>,<values>`

where

`B`, `C`, `M`, or `P` are applicable only to Micrologic devices (otherwise the D: section is omitted) and is the module (manager) identifier (Circuit Breaker, Chassis, Metering, Protection).

`<start_register>` is the first register number for a contiguous block of registers.

`<values>` is a comma-separated list of up to 10 values that will be written to the registers starting from <start_register>.

For example:

`T:BWF;S:100,1,2,3,4,-5`

Variable block writes have the following format:

where

`B`, or `C`, or `M`, or `P` is applicable only to Micrologic devices (otherwise the D: section is omitted altogether) and is the module (manager) identifier (Circuit Breaker, Chassis, Metering, Protection)

For example:

`T:BWV;`

The start register and the values to be written follow exactly the same rules and syntax as the definition for the Fixed Block Write, however, these are specified at the time the write operation is performed. For example, specifying "`S:100,1,2,3,4,-5`" as the write value for the tag "`T:BWV;`" would write values 1,2,3,4, and -5 to the registers 100, 101, 102, 103, and 104.

The Profile Wizard

This chapter discusses how to use the Profile Wizard to add an individual device profile to, or remove it from, a PowerLogic SCADA project. From the Profile Editor, you export profile data into a file that is available to be imported into a PowerLogic SCADA project. From there, use the Profile Wizard to add the device profile into the PowerLogic SCADA project. Real-time tags, control tags, and alarm tags in the Profile Editor become variable tags in PowerLogic SCADA. Alarm tags will also have a second tag in time-stamped digital alarms or time-stamped analog alarms, depending on whether they are on-board or PC-based.

To add a small number of profiles to the PowerLogic SCADA project, it makes sense to use the Profile Wizard. However, when you want to add a very large number of profiles, use the Automation Interface to save time. This feature allows you to add multiple devices in a single step. For information, see *Adding Multiple Devices through the Automation Interface* on page 74 for information.

In this section, you will find these topics:

Before You Use the Profile Wizard

- *Adding the Cluster, Network Address, and Servers*
- *When Setting Up More than two I/O Servers per Cluster*

Using the Wizard to Add and Remove Devices

- *Adding a Device to a PowerLogic SCADA Project*
- *Removing a Device From the PowerLogic SCADA Project*
- *Editing Devices in a PowerLogic SCADA Project*
- *Adding a Device to a PowerLogic SCADA ProjectAdding Multiple Devices through the Automation Interface*

Using the OLE Automation Interface to Add and Remove Devices

- *Examples*
- *Command Line Interface*
- *Interface Workflow*

Before You Use the Profile Wizard

Before you can add device profiles, you need to establish the folder structure and network addresses in the PowerLogic SCADA project: Launch the PowerLogic SCADA software: click Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Explorer. Three windows display:

- Citect Project Editor
- Citect Explorer
- Citect Graphics Builder

Have a copy of each device's connection information: communications protocol and IP address. You will enter this information when you add the devices.

In Citect Explorer, add a project that has the same name as the one set up in the Profile Editor (File > New Project). Then add the following configuration records in the Project Editor (Servers):

- at least one **cluster**; A cluster is a group of alarms servers, trends servers, reports servers, and I/O servers. For additional information about clusters, see Clustering in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help).
- **a network server address for each cluster**: The name added here is linked to the IP address. This is the address to be used by all of the servers in the cluster. When using the host computer as the server, you can use the IP loopback address: 127.0.0.1.
- at least one of each relevant type of **server**: There are four types of server: alarm, report, trend, and I/O:

  An **alarm server** monitors all alarms and displays an alarm on the appropriate client(s) when an alarm condition becomes active.

  Set 'Publish Alarm properties' to TRUE when setting up the alarm server. This is must otherwise you will not be able to edit the setpoints page. To view this field, open the Alarm Servers window (Servers > Alarm Servers). Press F2 to display the extended screen. choose TRUE in the Publish Alarm Properties field.

  A **reports server** controls report processing; you can request reports at any time or when specific events occur.

  A **trends server** controls the accumulation and logging of trend information; this information provides a current and historical view of the system, and can be processed for display on a graphics page or printed in a report.

  An **I/O server** is a dedicated communications server that exchanges data between I/O devices and PowerLogic SCADA Control Clients.
- a **project** to PowerLogic SCADA: Then compile and correct any errors. After the first compile, run the Computer Setup Wizard (Project Editor > Tools > Computer Setup Wizard. This wizard prepares the computer for use with PowerLogic SCADA.

Adding the Cluster, Network Address, and Servers

Follow these steps to add the network address, which consists of at least one cluster, servers, and the related IP address. Each project must have at least one cluster, a network address, and one of each type of server that will be used. Clusters are used to logically and/or physically separate the servers in a system. Having multiple clusters on a large project (called scaleability) will help maintain system performance.

1. From the Citect Explorer screen, display the project: In the upper left corner of the screen, choose the project from the drop-down menu.
2. From the Project Editor screen, click Servers > Clusters.
3. At the Cluster screen, type the cluster name.

For the cluster to be able to communicate with system devices, its name must be between two and 16 characters, no spaces. It must be unique to the project.

If there is only one cluster, the system will always automatically choose it for a project. If there are multiple clusters, you will always have to select the one that you want.

4. If desired, type a description in the Comment field.

   This field can be used for translation purposes. See *Entering Copy for Translation* on page 71 for more information.

5. Click Add.

   The new cluster name displays in the window.

6. Close the Cluster window.

   If a cluster already exists, you can add a new one: with the existing name displayed, click Replace to create a duplicate of the existing one; overwrite the name of the one you just added; finally click Add to save the name that you just typed. Both clusters will be in the system. To view all of the clusters added to this project, use the scroll bar on right side of the window.

7. Still on the Citect Explorer screen, click Servers > Network Addresses.

   Adding network servers enables you to handle data from large installations, and to use multiple servers.

8. At the Network Addresses screen, type a name for the network server; then enter the IP address for this server. This is the address that all of the servers in this project will use. If the servers are on the local machine, you can use the loopback address (127.0.0.1).

   The Comment field can be used for translation purposes. See *Entering Copy for Translation* on page 71 for more information.

9. Click Add, then exit the screen to save the server name.
10. Note the name so that you can re-use it for the remaining servers.
11. To add an alarm server, click Servers > Alarm Servers.
12. At the Alarm Servers screen, follow these steps:
    - At the Cluster Name screen, choose the cluster you will use for this server.
    - At the Network Addresses field, choose the network name that you added for this system.
    - In the Mode field, you can specify whether this server is primary or standby. The standby server will act as a backup, providing redundancy in case the primary server goes down.
    - In the Port field, select whether this is a primary or standby server (default: primary). You can have one primary and one standby server for each cluster.

      An additional set of prompts displays when you “expand” the screen. To expand, click F2 when the Alarm Servers screen is active. In the Publish Alarm Properties, choose whether alarm properties will be viewable as variable tags (default = false). If you set it to true, the alarm server monitors the port you specify as if it were an I/O connector.
    - Click add to save the new alarm server. Exit the window.
13. Repeat steps 11 and 12 for all other server types that you need to add.

    NOTE: When you delete a cluster, you need to also click File > Pack to completely delete it from the system.

14. Click File > Compile.
15. If there are errors and/or warnings after the page is compiled, do the following:

a. At each error, click GoTo, which opens the location where the error has occurred.
b. Using the information in the error message, correct the error.
c. After all errors are addressed, re-compile to verify that the errors have been removed.

For additional information, click Help at the error screen.

When Setting Up More than two I/O Servers per Cluster

If you need to add more than two I/O servers to a cluster, you need to define a redundant I/O device called *NetworkTagsDev* for each of the servers. If you do not do this, you can lose device status information during runtime.

If the cluster includes only one or two I/O servers, the I/O devices are automatically added when you add the cluster during Profile Wizard configuration (see *Adding the Cluster, Network Address, and Servers* on page 68). If a system has more than two I/O servers in a cluster, you must manually add the *NetworkTagsDev* I/O device for the remaining servers (after the first pair).

To create the board, port, and *NetworkTagsDev* I/O device, ensure the following in PowerLogic SCADA:

- all redundant *NetworkTagsDev I/O* devices have the same number
- the Startup Mode field is set to Standby; do this for all standby *NetworkTagsDev* I/O devices, including the one created by the Profile Wizard
- the Equipment field is set to <Cluster>_NetworkTagsDev

The field values for the forms in each of the I/O servers should be:

Boards Form

Board Name: <any unique name> (suggestion: BOARDy_SVRz)

Board Type: DISKXML

Address: 0

Leave everything else blank.

Ports Form

Port Name: <any unique name> (suggestion: PORTx_BOARDy_SVRz)

Port Number: <any unique number within the I/O server> (suggestion: x)

Board Name: <use the board name defined above>

Leave everything else blank.

I/O Devices Form

Name: NetworkTagsDev

Number: <same number as the one defined in the corresponding device>

Address: NetworkTagsDev

Protocol: DISKXML

Port Name: <use the port name defined above>

Startuo Mode: Standby

Equipment: <Cluster> _NetworkTagsDev

Leave everything else blank.

Notes:

- Startup Mode is only visible when in extended form mode (press F2 to toggle between simple form mode and extended form mode, while in the I/O device form).

- The Equipment field is hidden by default. To change it to visible, open units.dbf (in the project folder) in Excel.
- If the system has one or two I/O servers per cluster, the startup mode of the standby *NetworkTagsDev* I/O device could be set to StandbyWrite in the Profile Wizard. If the system has more than two I/O servers per cluster, the startup mode of all standby *NetworkTagsDev* I/O devices must be set to Standby.

  One side effect of this is that, when the system "fails over to a redundant I/O server," affected devices will momentarily be in communication failure as the system transitions to the redundant server.

Entering Copy for Translation

There are several “description” or “comment” fields throughout this tool. These fields can be used to create copy for translation purposes. If you type a Comment in this format:

@(XXX), where XXX = the copy that is to be translated,

the copy you typed in the comment field is added to the file named English.DBF, located in Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\User\[your project](for Vista operating systems: ProgramData\Schneider Electric\PowerLogic SCADA 7.10\User\[your project]). English.dbf contains terms that will be translated. After translation, there will be an additional .dbf file for each translated language. At runtime, the user will be able to choose the .dbf file that is to be used in the display, thus changing the language used in the display.

Using the Wizard to Add and Remove Devices

This section describes how to use the Profile Wizard to manage device information in a PowerLogic SCADA project. Before you begin adding devices, make sure you have completed the tasks described in *Before You Use the Profile Wizard* on page 68.

You will use these procedures to add remove, or edit device data:

Adding a Device to a PowerLogic SCADA Project

Removing a Device From the PowerLogic SCADA Project

Editing Devices in a PowerLogic SCADA Project

Adding a Device to a PowerLogic SCADA Project

The Profile Wizard allows you to add one device at a time. To add more than one device at a time, you can use the Automation Interface. See *Adding Multiple Devices through the Automation Interface* on page 74 for instructions.

1. From the Citect Explorer screen, display the project to which you want to add the device: In the upper left corner of the screen, choose the project from the drop-down menu.
2. From the Citect Project Editor, click Tools > Profile Wizard.

   The Profile Wizard welcome screen displays.

   NOTE: When entering a required value, for example device name and address, a red exclamation mark (**!**) displays when the name is not valid or is missing.
3. Cluster Setup: Before you add devices, you need to set up each cluster that was created for the project (you must have added at least one cluster to PowerLogic SCADA before you do this):

   a. From the Profile Wizard, under System Devices, click Cluster Setup. Click Next.

   b. At the Enter Instance Information screen, a cluster name displays. Click Next.

   c. If there are multiple clusters, the Select cluster screen displays. Choose the cluster you want to set up. Click Next.

   d. If there are multiple I/O servers in the cluster, the Select I/O servers screen displays. Check *Supports Redundancy* and select the I/O servers to which you want to add the device. Click Next. If there are more than two IO servers per cluster, you must set them up manually (see *When Setting Up More than two I/O Servers per Cluster*).

   e. At the Ready to perform action screen, click Next.

   If you have more than one cluster to add, repeat steps 3a through 3e for each cluster.

   When you are finished adding clusters and I/O servers, you return to the Profile Wizard welcome screen.
4. If you are adding devices to the project for the first time, click Next.

   or

   If devices have been added, click Create an I/O Device in the project, then click Next.
5. At the Choose profile screen, note that the devices are organized according to the category/subcategory to which they were added. Select the first device profile that you want to use to add a device to the PowerLogic SCADA project. Click Next.
6. At the Enter Equipment information screen, type a descriptive device name, for example: *CM4Bay1Circuit1* (no spaces or punctuation; to allow space in PowerLogic SCADA, the preferred limit is 16 characters). The Comment field is stored in the equipment.dbf file.

NOTE: There are several “description” or “comment” fields throughout the PowerLogic SCADA product. These fields can be used to create copy for translation purposes. If you type a Comment in this format:

@(XXX), where XXX = the copy that is to be translated,

the copy you typed in the comment field is added to the file named English.DBF ([Project Directory]\Schneider Electric\PowerLogic SCADA 7.10\User\[your project]), which contains terms that will be translated. After translation, there will be an additional .dbf file for each translated language. At runtime, the user will be able to choose the .dbf file that is to be used in the display, thus changing the language used in the display.

Click Next.

7. At the Select Communications Method screen (does not display if the device has only communication method), select the communications protocol used by this device. Click Next.
8. At the Communications Settings screen, type the requested information. Depending on the communication method selected, you may see different prompts. Click Next.
9. At the Ready to perform action screen, click Next.

   After the device is added, a screen displays telling you that the project was updated successfully.

   To view a detailed list of all of the device profiles and all operations performed in the project, check the View audit log box. The list displays after the device is added.

   To continue adding or removing device profiles, click Next. Repeat steps 3 through 8.
10. When you have finished adding devices, uncheck the Add/remove more equipment option, then click Finish.

    If you checked the View audit ... box, the list displays.

    If you did not check the Add/remove... box, the Profile Wizard closes. If you checked the Add/remove ... box, the Welcome screen displays again.
11. From any of the three PowerLogic SCADA screens, compile the project.
12. When all errors are corrected, click File > Run to view the runtime environment.

Removing a Device From the PowerLogic SCADA Project

1. From the Citect Explorer screen, display the project from which you want to remove the device: In the upper left corner of the screen, choose the project from the drop-down menu.
2. From the Citect Project Editor, click Tools > Profile Wizard.

   The Profile Wizard welcome screen displays.
3. Click Remove a device from the project, then click Next.
4. At the Remove a device screen, click the dropdown menu to display the equipment names that were used when the device profiles were added in the Profile Wizard. From this list, select the device that you want to remove. Click Next.
5. At the Ready to perform action:

   To compress the project files after removing this profile, click Pack databases.

   Click Next.

   After the device is deleted, a screen displays telling you that the project was updated successfully.

   To view a detailed list of devices that you have added or deleted, check the View audit log box. The list will show all of the device data that has been added, as well

as the data that has been removed in this session. The list displays after you click Finish.

6. To continue removing devices, check Add/remove more devices, then click Next. Repeat steps 4 and 5.
7. When you have finished removing devices, uncheck Add/remove more devices, then click Finish.

   If you checked the View audit ... box, the list displays.

   If you did not check the Add/remove... box, the Profile Wizard closes. If you checked the Add/remove ... box, the Welcome screen displays again.

Editing Devices in a PowerLogic SCADA Project

Editing a Profile and Adding it Back to the Project

After you have added devices to the PowerLogic SCADA project, and you need to make major changes to the device profile (e.g., add a large number of tags), do the following:

1. Delete the device from the PowerLogic SCADA project: Use the “Remove a device from the project” feature in the Profile Wizard.
2. Edit the device profile in the Profile Editor, and run the Export feature,
3. In the Profile Wizard, add the device back to the PowerLogic SCADA project.

Editing a Device in PowerLogic SCADA Only

If, when you added the device to the PowerLogic SCADA project, you entered incorrect information (e.g. wrong IP address), do the following:

1. Delete the device from the PowerLogic SCADA project: Use the “Remove a device from the project” feature in the Profile Wizard.
2. In the Profile Wizard, add the device back to the PowerLogic SCADA project.

Adding Device Data in PowerLogic SCADA Only

If you need to add a small amount of data to a device that is in the PowerLogic SCADA project (e.g., add a single tag), add it directly in PowerLogic SCADA. Be sure that you also add it to the device in the Profile Editor so that it is available for other devices in the future.

Adding Multiple Devices through the Automation Interface

The Automation Interface allows you to add multiple devices at a time, instead of using the Profile Wizard, which only adds one device at a time. Follow these instructions to add large quantities of devices into the PowerLogic SCADA project. (This example is for MODBUS TCP. See the table at the end of these instructions for other comms methods.)

NOTE: If you install PowerLogic SCADA on a computer that does not have Microsoft Excel installed, you will not be able to run the Automation Interface. When you try to run the macro, you will see an "out of memory" message. The solution is to uninstall both Powerlogic SCADA and Excel; then, first reinstall Excel, then reinstall PowerLogic SCADA.

1. For an existing project: Before you begin, make a backup copy of your project. See *Backing Up a Project* on page 96 for instructions.

   For a new project: In the Citect Explorer, add a new project. From the Project Editor, define a cluster; and add alarm, trend, and I/O servers (see *Before You Use the Profile Wizard* on page 68).

2. In the Profile Editor, create a project that includes the devices included in this installation. Give the project the same name you used in step 1.
3. Open Excel 2003 or Excel 2007 and enable macros (Click the Developer tab, then Macro security. Click Enable all Macros).
4. In Excel, open *Example Automation.xls* (Program Files\Schneider Electric\PowerLogic SCADA 7.10\Bin). This sample spreadsheet can be edited to include the communications methods you will use. Then, enter the devices that you want to add to the PowerLogic SCADA project in the spreadsheet.
5. From the Setup worksheet at the bottom of the spreadsheet, make these changes:
   a. Cell B2: Enter the path to Citect.ini:

      For Windows XP: Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\Config.

      Vista operating systems: ProgramDataSchneider Electric\PowerLogic SCADA 7.10\Config.
   b. Cell B3: Type the project name.
   c. Cell B4: Enter the path to the location of *equipment.profiles:*

      For Windows XP: Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\User\ProjectName).

      For Vista operating systems: ProgramData\Schneider Electric\PowerLogic SCADA 7.10\User\ProjectName).
6. From the Input worksheet at the bottom of the spreadsheet, enter the following information for each device that you want to add:
   a. Profile Name: Type the complete name (to view exact addresses, open *equipment.profiles* in Notepad), e.g.: SchneiderElectric.Protection Device.Sepam S42 Full
   b. Name: Enter the device name (no spaces or punctuation, limit of 32 characters). This becomes the equipment name in the PowerLogic SCADA project. You will use this when adding genies to drawings.
   c. Cluster: Type the cluster name from step 1.
   d. Primary IO Server Name: Type the I/O server added in step 1. It becomes the primary server.
   e. CommsMethod: Type the communications protocol being used, e.g., MODBUS/RTU via Gateway. See table below for alternate communication connections.
   f. Additional Prompts will display, depending on the communication connection being used. See the table below for a list of data required for each comms method.
   g. EquipmentAddress: Type the device address (required only for MODBUS/RTU and MODBUS/RTU via Gateway; optional for other comms methods).
7. Close Citect Explorer if it is open.
8. In Example Automation.xls, click Tools > Macro > Macros. Highlight the macro called *ThisWorkbook.AddDevices*, then click Run.

   Devices are created, according to the information you added. When device creation is complete, a message displays.

   NOTE: For large systems, this can take quite a long time. While the macro is running, there is no feedback on the screen.
9. Launch PowerLogic SCADA and verify communication for all of the devices listed in the spreadsheet.

If *Example Automation.xls* does not include a communications connection that you want to use, you can add the new connection. The following table lists connections and the required information for each.

CommsMethod	ComPort	Baud Rate	DataBits	StopBits	Parity	IP Address	Equipment Address	
Generic Serial [1]	X	X	X	X	X		X	2
MODBUS/RTU	X	X	X	X	X		X	
Generic TCP						X	X	2
MODBUS/RTU via Gateway						X	X	
MODBUS/TCP						X	X	2

1 Generic Serial comms must be set up via Gateway in this release.

2 Optional

Using the OLE Automation Interface to Add and Remove Devices

In this section, you will find these topics:

Examples

Command Line Interface

Interface Workflow

There is also an OLE Automation interface that allows you to add and remove devices from a project. The interface can be used by VB, .NET, and by C++ clients. It is the building block for building a bulk interface based on Excel.

IEquipmentAutomation. The equipment automation interface facilitates reading equipment profiles, selecting a profile by name, setting the required information and saving the profile to a project.

INameValueCollection. The **INameValueCollection** interface provides the means to set the attributes and comms properties. The interface provides the method **SetValue** for setting a named value for the currently selected profile. If the name is not recognised, the method will return false and errors will be handled by the current error output method. In the case where an error has occurred, the **GetErrorMessages** may be called on the **EquipmentAutomation** interface to get more information about what went wrong.

Examples

C#

After a C# project is created, a reference to the EquipmentAutomation.dll should be established.

Using the equipment automation will be made easier by using the correct namespace as shown:

```
using Citect.CitectSCADA.Equipment.Automation;
```

The following is an example of some typical C# code that will select a profile, set it up and save it.

```
EquipmentAutomation myAuto = new EquipmentAutomation();
myAuto.IniFile =
@"C:\Documents and Settings\All Users\Application Data\Schneider
Electric\PowerLogic SCADA 7.10\Config\citect.ini";

myAuto.ReadProfiles(
@"C:\Documents and Settings\All Users\Application Data\Schneider
Electric\PowerLogic SCADA 7.10\User\TestProfiles\equipment.profiles");

string profileName = "Schneider Electric.Protection Device.Branch Circuit Monitor Full";

if (myAuto.SelectProfile(profileName))
{
    If (myAuto.SelectProfile(profileName))
    {
        NameValueCollection myAttributes = myAuto.ProfileAttributes;
        myAttributes.SetValue("NAME", "yourEquipName");
        myAttributes.SetValue("CLUSTER", "yourCluster");
        myAuto.ServerNamePrimary = "PrimaryServer";
        myAuto.SetCommsMethod("Modbus/RTU via Gateway");
```

```
            INameValueCollection myCommsProperties =
            myAuto.CommsMethodPropertiesPrimary;
            myCommsProperties.SetValue("IPAddress", "10.0.0.1");
            myCommsProperties.SetValue("EquipmentAddress", "1");
            equipmentAutomation.SaveProfile("TestProfiles");
        }
    }
```

Excel VBA

An example spreadsheet, called “Example Automation.xls,” is provided in the bin directory. This spreadsheet has a “Setup” and an “Input” sheet. The Setup sheet contains information about the ini file location, the project name, and the profile that is used. The Input page contains information that is used by the macro to select a profile, set the attributes, set the IO server name, select the comms method and set the appropriate comms properties.

Command Line Interface

The Device Wizard allows the wizard to be launched from the command line. It uses the flag /i to specify the location of the Power SCADA Ini file (usually found in the config directory of the user/data directory), and the flag /p to specify the project name. The arguments to these flags must immediately follow the flag without a space. It is recommended to use quotes around the ini filename.

In addition to launching the wizard, the Device Wizard allows a merge of a device profile XML file with the current PowerLogic SCADA “master” device profile XML file. The given file must conform to the profiles schema.

Additional help may be obtained by using the /h flag.

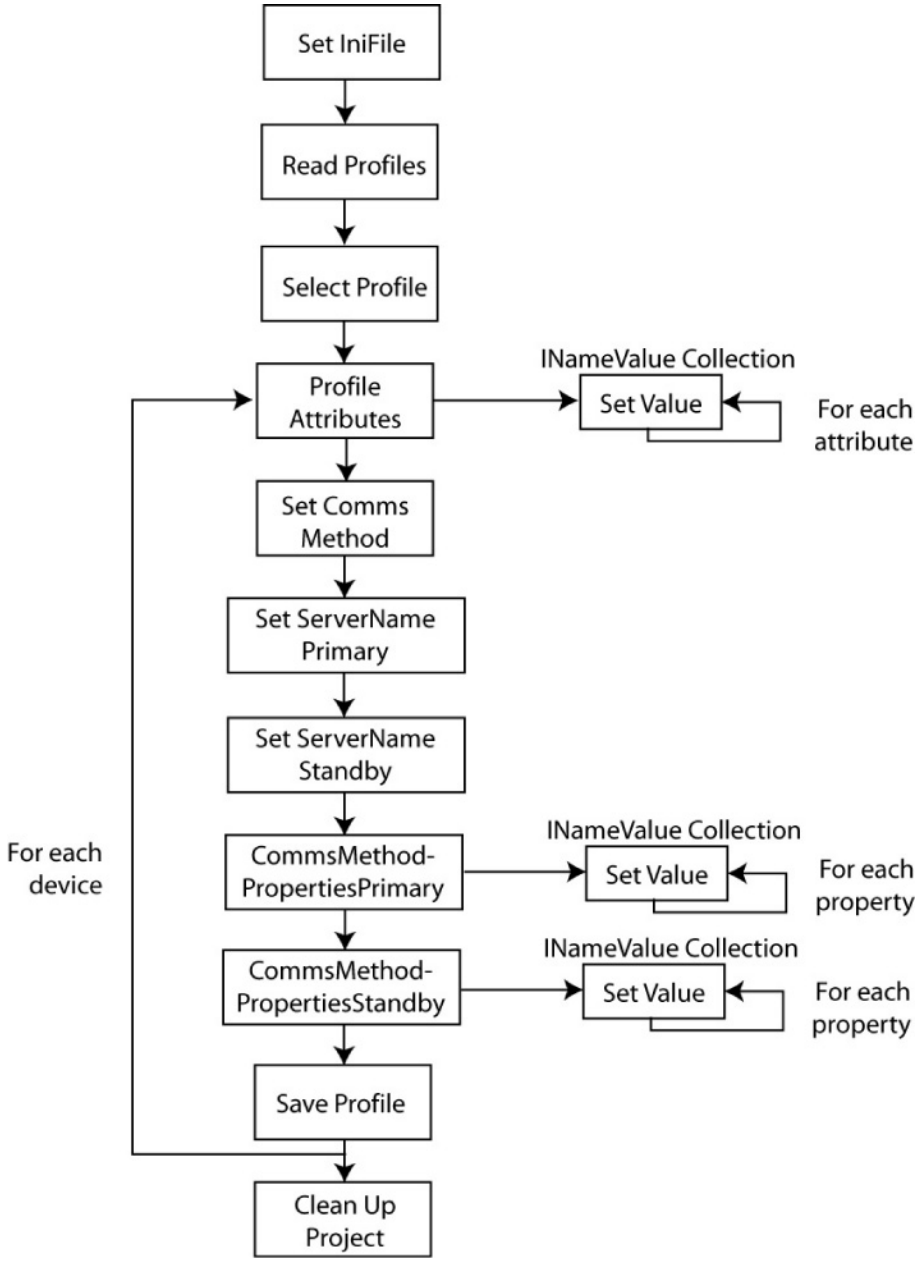
EquipmentWizard.exe **/?**

EquipmentWizard.exe **/p**”ProjectName” **/i**”iniFile”

EquipmentWizard.exe **/p**”ProjectName” **/i**”iniFile” **/m**”MergeProfileName”

Interface Workflow

The general workflow, when adding devices via the automation interface, is illustrated in the following flowchart. Success values can be checked at each step in the process by checking the Boolean return value or by checking the engine errors.



Prerequisite information

Before creating a profile in a project, key information about the profile and the project in which it is being added should be established. This information can be established by examining the profile definition in the equipment.profiles file for the project. Attribute names and the supported comms methods must be known in order to know what needs to be set. The cluster and the IO server names from the project that the equipment is being added to should also be established.

Error messages can also provide useful feedback, and a way of discovery for what is required. The profile will not be saved until all required information is determined to be valid and complete.

Interface

The automation interface is designed to lead the developer through the workflow, providing useful feedback along the way in the form of error messages as to what was missing or incorrect. The interface is a COM automation interface that allows multiple profiles to be saved to a project.

IniFile [Property] — The equipment engine relies on the location of the ini file to determine where projects are stored and for validating that a profile can be successfully imported into the selected project.

ReadProfiles —The equipment engine reads the given profile file into memory and ensures that it is matches the expected schema.

SelectProfile — Selecting a profile initialises the engine for a specific profile. A profile is selected by referring to the qualified name. The qualified name is made up of the categories that contain the profile separated by a '.'.

ProfileAttributes [Property] — Attributes are set by first retrieving an **INameValueCollection** from the **ProfileAttributes** property. The **INameValueCollection** provides an interface method to set a named value. This will set the attribute with the given name for the selected profile in the equipment engine. SetValue returns a Boolean value indicating success.

ServerNamePrimary and ServerNameStandby [Properties] — The name of the primary IO server is required to be set (ServerNamePrimary) before saving a profile. A standby server name may also be optionally specified (ServerNameStandby). The server names must be valid server names in the project that the profile is being imported into.

SetCommsMethod — Each profile specifies which comms methods it supports. Setting the comms method requires that the currently selected profile supports the comms method specified.

CommsMethodPropertiesPrimary and CommsMethodPropertiesStandby[Properties] — Comms properties are also set via **INameValueCollection** which is returned when calling **commsMethodPropertiesPrimary** or **commsMethodPropertiesStandby**. If a standby server was specified, the comms properties for the standby server are required to be set. The comms properties required depend on the comms method selected. Below is a table listing the comms properties required for each of the comms methods.

CommsMethod	ComPort	Baud Rate	DataBits	StopBits	Parity	IP Address	Equipment Address	
Generic Serial[1]	X	X	X	X	X		X	2
MODBUS/RTU	X	X	X	X	X		X	
Generic TCP						X	X	2
MODBUS/RTU via Gateway						X	X	
MODBUS/TCP						X	X	2

1 Generic Serial comms must be set up via Gateway in this release.

2 Optional

SaveProfile — SaveProfile takes the name of the project to which the profile is being saved. Saving a profile involves writing all output values defined in the profile definition to the specified project. A two-phase commit process is used to ensure that data is

valid before committing changes permanently. Saving profiles to a different project will result in an automatic cleanup of the first project. See CleanupProject for more information.

ResetSelectedProfile — A call to ResetSelectedProfile will clear data associated with the selected profile, as well as clearing any accumulated errors. After saving a profile successfully, this method will be called automatically to allow the next profile to be saved.

DeleteEquipment — DeleteEquipment requires the project name that the equipment is to be deleted from as well as the name of the equipment that is being deleted. It will scan through the project looking for records that were added for a specific piece of equipment and will remove them accordingly. Removal of file outputs which were defined in the profile from which the equipment was added is not performed as these files may contain sensitive data that has since been added to these files. Additionally, those dbf entries in the profile under dbf-file that do not have an equip field are not removed, as there is no way of determining if they are exclusively used by this piece of equipment.

CleanupProject — CleanupProject performs operations such as re-indexing of dbf files and packing dbf files. Errors detected by this phase are reported as warnings and do not trigger a rollback. This operation should be called after all save operations have been performed on a project.

NoErrors [Property] — This is a Boolean property that can be used to determine if any errors have been collected.

GetErrorMessages — Detailed error messages may be obtained by calling **GetErrorMessages**, which returns a string with errors and warnings on multiple lines.

ErrorOutput [Property] — The error handling approach be set with the **ErrorOutput** property. Error Handling.

The values used for the error output are:

- None = 1
- Dialog = 2 [default]
- Console (silent if run from excel) = 4

Error Handling — Each interface method returns a Boolean indicating the success of that operation. Accumulated errors will prevent progression through the workflow and as such, this return state should be examined at each stage. A call to **ResetSelectedProfile** will clear accumulated errors, but it is recommended that errors are addressed before you proceed.

Information about errors and warnings can be obtained through a call to **GetErrorMessages**. By default, a dialog will display with error information at each step of the workflow, but this can be controlled with the **ErrorOutput** property.

Graphics: Using the Design Time Environment

This chapter discusses how to use genies to create one-line diagrams for customers. In this section, you will find these topics:

Typical Workflow
The PowerLogic SCADA Pages
Creating a PowerLogic SCADA Project
- *Reusing Projects Created in the Profile Editor*
- *Adding a New Graphics Page*
- *Creating a One-Line on a Graphics Page*
- *Using Menu.dbf to Change the Graphics Page Appearance*
- *Creating and Editing Genies*
- *Deleting Information from PowerLogic SCADA*

Working with Alarms
- *Alarms Overview*
- *Adding Setpoints and Delays*
- *Making a PC-Based Alarm Digital*
- *Setting Up an Alarm Based on an Enumeration*
- *Changing an Alarm Severity*
- *Adding an Onboard Alarm Tag*
- *Setting Up Audible Alarms*
- *Setting Up Audible Alarms*

Restoring and Backing Up Projects
- *Restoring a Project*
- *Backing Up a Project*

The Project Color Change Tool
- *Adding a Custom Genie for Use in the Color Change Tool*
- *Changing the Colours of Busbars and Circuit Breakers*

Using Cicode to Customize a Project

Typical Workflow

When creating a graphics page, these are the steps you will usually follow:

1. Before you begin, make sure you have:
   a. created a project in the Profile Editor
   b. added a project with the same name to PowerLogic SCADA; added at least one cluster, network address, and server
   c. exported the project from the Profile Editor
   d. used the Profile Wizard to add devices to PowerLogic SCADA
2. From the Graphics Builder window, add a new page, using the style and page template you prefer (instructions assume that you will use the default template).
3. Add genies to the page to create the one-line drawing.
4. Compile and run the project. Note any errors and warnings; correct all errors. Review warnings for problems such as missing tags.
5. Update the menu.dbf file for the project.
6. Run the project to view the graphics page in the runtime environment.

The PowerLogic SCADA Pages

When you launch PowerLogic SCADA (Start > Programs > Schneider Electric > PowerLogic SCADA Explorer), three windows are opened:

- **Citect Explorer**: Use this application to explore and add projects. PowerLogic SCADA is shipped with a project that has example page configuration. Project names are listed in the left-hand pane.
- **Citect Project Editor**: Use this application to edit the SCADA configuration. For example, use it to add clusters, network addresses, and servers; and to launch the Profile Wizard and the Project Color Change tool.
- **Citect Graphics Builder**: Use this application to build graphic display pages for a project. A standard PowerLogic library of genies are provided. The genies contain logic such as control outputs and rack-in/rack-out. Once assembled and opened in the runtime environment, they provide an interactive view of an installation's power system.

Creating a PowerLogic SCADA Project

NOTE: The information in this section is based on the Normal template from the PLS_Style_1 library. The Normal template includes all of the tabs and buttons that are described in the topics here. If you want to edit selections on the tabs and buttons, use the menu.dbf file. For instructions on editing templates with menu.dbf, see *Using Menu.dbf to Change the Graphics Page Appearance* on page 89.

When you modify the template or create a completely new template, the information provided here for creating graphics pages may only apply as a guide.

In this section, you will find these topics:

Reusing Projects Created in the Profile Editor

Adding a New Graphics Page

Creating a One-Line on a Graphics Page

Using Menu.dbf to Change the Graphics Page Appearance

Creating and Editing Genies

Deleting Information from PowerLogic SCADA

Reusing Projects Created in the Profile Editor

You may want to create a “standard” project that can be reused for different installations. For instructions on saving and then reusing projects, follow these steps:

1. Export the project from the Set Up Project tab of the Profile Editor.
2. Click the folder icon beside the Export link:

3. Copy the file (Equipment.profiles) that displays. If you need to use this file to another computer, you can move it to a portable drive.
4. On the PowerLogic SCADA server computer, paste Equipment.profiles in this location:

   Windows XP: Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\User\[Project]

   Vista operating systems: [Drive Letter]:\ProgramData\Schneider Electric\PowerLogic SCADA 7.10\User\[Project]

   where:

   [Drive Letter]: is the drive on which you installed the PowerLogic SCADA server

   the Application Data and Program Data folders are not hidden (set the folder view for “view hidden folders”)

   [Project] is the name of the project you are creating; you must have already added this project to PowerLogic SCADA (see Before You Export, above).
5. Be sure you have created the files described in *Before You Begin* on page 85.

Before You Begin

Before you start adding data in the PowerLogic SCADA project, make sure that you have:

- used the Profile Editor to add all of the device types, device profiles, and projects
- from Citect Explorer, created a project in PowerLogic SCADA; from the Citect Editor, added clusters, network addresses, and servers
- exported devices from the Profile Editor
- added devices into the PowerLogic SCADA project, using the Profile Wizard

Adding a New Graphics Page

You need to add a project in two places before you add pages. First add the project to PowerLogic SCADA, from the Citect Explorer screen (see steps below). Then add it to the Profile Editor. See*Adding, Editing, or Deleting a Project* on page 50 for instructions.

1. From the Citect Explorer screen, choose the project for which you will create a graphic.
2. On the Graphics Builder screen, click File > New > Page.
3. On the Use Template popup, set the resolution to SXGA (or default), and uncheck the Title Bar option. Select the page template (for example, normal). Click OK.

   NOTE: If you use any style that is not preceded by “pls,” the PowerLogic SCADA graphics features will not be included. If you want to change the attributes of the default style, copy *pls_style_1* and paste it in your project; rename it, and make the desired changes. The new template can then be used for your pages.

   The graphics page displays in design-time mode.
4. To make changes to the menus and tabs that display on the screen, use the menu.dbf file. See *Using Menu.dbf to Change the Graphics Page Appearance* on page 89 for instructions on making these changes.
5. Save the graphics page, giving it the desired name and selecting the project that you have just added.

Setting a new page as the project's startup page

After you add a new page, follow these steps to make it be the system starting page. If you do not make these settings correctly, the system will not properly execute the startup routine, and the browser navigation buttons will not work.

1. In the [MultiMonitors] StartupPage parameter, add the new page:
   a. Open the Parameters dialog (Project Editor > System > Parameters).
   b. In the Section Name field, type MultiMonitors.
   c. In the Name field, type StartupPage.
   d. In the Value field, type the name of the new page.
2. In the Computer Setup Wizard (General Options Page), leave the ini settings as <default>.

Creating a One-Line on a Graphics Page

You will build a one-line by adding genies to the new page. For more information about genies, see Understanding Genies in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help).

To begin creating a one-line, start adding genies to the page:

1. Click the “genie” icon on the toolbar:

2. From the Paste Genie screen, choose a library (to ensure that all PowerLogic SCADA features are available, choose a library that begins with “pls”). If you want to edit a genie or create a new one, see *Creating and Editing Genies* on page 90 for instructions.

3. From the available icons in the selected library, select a genie; click OK to paste the genie on the graphics page.
4. A popup window displays for you to enter genie properties. Enter the requested information (descriptions below for circuit breakers, switchgear, and busbars only):

Circuit breakers, both ANSI-type and IEC-type

a. Equipment: From the drop-down list, choose the name of equipment represented by the genie (this is an equipment name that you entered in the Profile Wizard or the Automation Interface, e.g., *CM4Main*). To view the equipment you can use for this page, you must have the project selected in the Citect Explorer window.

b. Busbar Connections: Where you are connecting to busbars, type the numbers for the connection source and destination. These are arbitrary, and only indicate how the genies in the one-line fit together on the page. Busbar numbers can be from 0 to 74. Numbers must be unique within each page. We recommend that they be consistent across pages for future compatibility.

c. Label (currently, labels are only available on larger circuit breakers): If applicable, type the name you want to appear on upper left corner of the runtime page.

d. Value Type: From the drop-down list, select the tag you want to use for this genie. This information displays in the lower left corner of the genie in run time.

The tag you choose here causes real-time data (for example, “currents”) to display on the runtime environment of the Graphics page. If you do not select a tag here, the status screen will have a blank in that spot. This tag also determines what displays in the popup (volt, amp, watts, etc.) See *Using the Genie Status Page* on page 117 for more information.

e. Units: From the drop-down list, select the unit (created in the Profile Editor). Units will display beneath the Value Type on the run time screen.

f. Multiplier: Enter the multiplier you want to use for the units in this graphic.

Switchgear

a. Busbar Connections: Where you are connecting to busbars, type the numbers for the connection source and destination. These are arbitrary, and only indicate how the genies in the one-line fit together on the page. Busbar numbers can be from 1 to 100. Numbers must be unique within each page. We recommend that they be consistent across pages for future compatibility.

b. Label (currently, labels are only available on larger circuit breakers): Type the name you want to appear on upper left corner of the page in the runtime environment.

c. Equipment: From the drop-down list, choose the name of equipment represented by the genie (this is an equipment name that you entered in the Profile Wizard or the Automation Interface, e.g., *CM4Main*). To view the equipment you can use for this page, you must have the project selected in the Citect Explorer.

Busbars

a. Busbar Number: Use busbar numbers to associate devices within the drawing; they are also an aid to help you set up animation for genies in the drawing. The busbar number you enter here is also used on the associated device. We recommend that they be consistent across pages for future compatibility.

b. Busbar Voltage: Type the number for the busbar’s voltage level, determined by the standard used in this installation. (Colours for busbar voltage levels are determined in the Project Color Change tool. See *Changing the Colours of Busbars and Circuit Breakers* on page 97 for more information.) To ensure that

the Project Color Change tool operates correctly, voltages need to be consistent across the project.

c. Line Active: From the drop-down menu, select the tag for the incoming voltage for device specified in the connected equipment. This can also be a Cicode function of expression (such as MyTag1 > 0).

As an expression, use Line Active to link multiple-page drawings: on the incoming busbar on page 2, enter a line active statement that is equal to the tag used for the last device on page 1.

NOTE: If a field condition is available that can be used to indicate the actual busbar active state, it should be entered in the Line Active field.
Also, if there are transformers in the one-line, the busbar may not change colour smoothly after state changes. To minimize any delay in colouring busbars, you should, if possible, enter a valid expression in the Line Active field. For example, if you choose phase A voltage for line active of “MYCM4,” and the incoming voltage is 440V, Line Active might read: MYCM4\MMXU1\PhV\phsA > 400. It is especially important to enter a line active expression for busbars that are directly upstream from the transformer.

5. Click OK to save the genie to the page.
6. If a message displays telling you that variable tags are not found, and asking if you want to add the unknown tags, this is not a problem. The genie is expecting to see a variable, but one may not exist in the equipment (for example, it could be looking for a “racked out” variable in a Sepam). Click No to this message.

   NOTE: To disable this message, go to the Project Editor page. Click Tools > Options. From the Options window, de-select “Prompt on tag not exist.”

   The genie is added to the page.
7. Continue adding the genies to make up the graphics page.

   After you create the graphics page, you need to compile it. You can run the page to review its elements and correct any errors.
8. On the Graphics Page, click File > Compile.

   While the page is being compiled, the PowerLogic SCADA Compiler displays each file name as it is being compiled. A message then displays, telling you that compilation was successful.
9. If there are errors and/or warnings after the page is compiled, a message box displays, telling you the number of errors and warnings. You must understand all warnings and correct all of the errors. To begin, do the following:

   a. Click OK at the PowerLogic SCADA Compiler message.

      The Compile Errors window displays the first error.

   b. Note the error message. For more information on each error, click Help; note the Description for that Error Message.

   c. To correct the error, click the GoTo link.

      The appropriate window displays for you to correct that error.
10. Correct each error. Warnings do not need to be corrected in order to run the project; but they should be checked to see if they impact the project (such as a tag that is not defined).
11. If you delete any entries, click File > Pack to clean up the files.
12. Re-compile and verify that all of the errors have been addressed.

    NOTE: You can also view the Compile Errors window from the Project Editor (File > Compile Errors).
13. After all errors are corrected, run the project (File > Run).

The Graphics page displays in the runtime environment. Note: Until you edit the menu.dbf file (see step 14), only basic tabs will display on the Graphics page.

14. Before you can view your one-line, you need to edit the menu.dbf file, which controls the appearance of the graphics page in the runtime environment. See *Using Menu.dbf to Change the Graphics Page Appearance* on page 89 for help.

Using Menu.dbf to Change the Graphics Page Appearance

The menu.dbf template controls certain aspects of the runtime environment screen appearance: among other things, tabs and menus on the screen. Entries in menu.dbf can also specify actions to be taken when an option is selected.

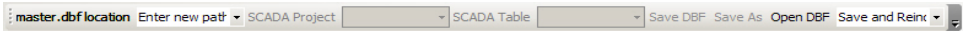
Because it takes much effort to create the items that display on a graphics page, you can use menu.dbf from the Power SCADA pls_example project as a template for your project template. From it, you can copy and paste settings into an individual project's menu.dbf file.

You will use the DBF Add-In tool, which should have been installed during the PowerLogic SCADA installation. These instructions assume that you are copying from pls_example and pasting into your project's DBF file.

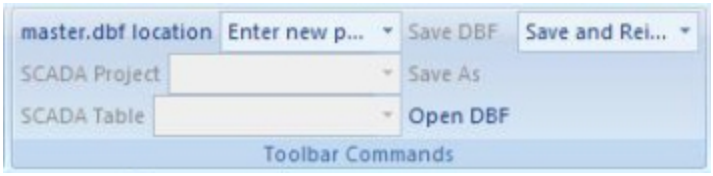
NOTE: If you did not have Microsoft Excel installed before you installed PowerLogic SCADA, this toolbar can be installed separately, using the *Project DBF Add-in for Excel* installation from the installation disk (\Citect\ProjectDBFAddIn).

1. Ensure that the DBF Add-in tool is installed on the computer. If it is not installed, use the PowerLogic SCADA installation disk to install the tool.
2. Open Microsoft Excel.

   If you have Microsoft Excel 2003 on the computer, the DBF add-in is a floating toolbar:



   If you have Microsoft Excel 2007 on the computer, the DBF add-in is a docked toolbar, in the upper left corner of the Excel window:



3. In the master.dbf location field: If the location for master.dbf does not display, type it here. The default location is [project drive] Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\User.
4. In the SCADA project field, choose the PLS_Example.
5. In the SCADA table field, choose Menu Definitions.

   Menu.dbf displays in Excel.
6. Copy the data you want to use in the project template.
7. Close the PLS_Example menu.dbf.
8. From the Add-in toolbar, change the SCADA project field to your project.
9. Make the changes you want: paste in the data from the PLS_Example menu.dbf and/or type your own entries. See the table below for descriptions of the columns in menu.dbf.

   NOTE: Do not resize the columns; this will corrupt the PowerLogic SCADA project.

10. Click Save DBF and close the master.dbf file.
11. Close Microsoft Excel.
12. Compile the project; then run it again.

    The Graphics page in the runtime environment will now display according to the settings you have made in menu.dbf.

NOTE: You can edit the menu.dbf file while you have the runtime environment open. To update the graphics page, you need to re-load menu.dbf for this project. To do so, click the small reload button () at the top right corner of the runtime environment screen.

The columns in menu.dbf for PLS_Example control the following:

Column	Control
Page	Must be “Generic” for the default template.
MenuName	These are the names of the tabs in the upper row.
MenuItem	These are the names of the tabs in the lower row. If there is not action associated with a menu item, but you wan to enable submenu items: enter an exclamation point in MenuItem
Submenu	When a tab on the lower row contains subtopics, this controls the names of the subtopics.
Action	Enter either: - a graphics page - a Cicode function; add a question mark (?) as a prefix; for example, ?PLSDspShowAlarm 0
Privilege	Not used in this release.
Disabled	Not used in this release.
Checked	Not used in this release.
BtnWidth	Not used in this release.
Icon	These are the symbols that display during run time, to the left of the related text. To add a symbol, reference it this way: library.symbol name

Creating and Editing Genies

The easiest way to create a new genie is to make a copy of a similar genie from the standard PowerLogic SCADA library and edit it in your project. We recommend, when adding layers, that you keep the dimensions of the new layers the same as the original. For a list of all of the standard PowerLogic genies, including all of the smaller parts that you could use to create genies, see *Appendix 1: Default Genie Library* on page 129.

When you modify a genie, the modifications will be in effect for all instances of that genie in the project.

To create a custom genie, follow these steps:

1. Open the genie that you want to use as a template: From Citect Graphics Builder, click File > Open. Click the Genie tab, then select the library and the template genie. Click OK.
2. Save the genie with a new name in your local project.
3. To separate the genie into layers, click the genie and drag a layer to the bottom right.
4. Repeat the process to pull all of the layers apart.

   Each layer is a sub-genie that controls a different aspect of the overall genie.

5. Make the changes that you wish to the genie.

   NOTE: Be careful to maintain the same dimensions for any new layers that you create.
6. To re-assemble the genie, draw a marquis around all of the parts; then click Arrange > Align, and choose Top and Left. Click OK.

The newly created genie, when applied to a page, will display with the generic input form. To create a customized form similar to those found with the default genies, you must create a new .FRM file. Examples are found in the PLS_Include project directory.

You can rotate any of the genies. The descriptive text will not rotate; however, the features, such as when you rack it in or out, will operate correctly.

To create a custom symbol for a custom genie, follow these steps:

PowerLogic SCADA genies are complex PowerLogic SCADA genies with the added ability to animate colour during runtime (showing changing states). These genies are comprised of multiple symbols that have the same attributes (all will work with Project Color Change tool, all call the same Cicode function, etc.).

1. Create the new symbol.
2. Configure the symbol to work with the Project Color Change tool: from the Symbol properties screen, choose the Access tab. In the Description field, enter <PwrLineColor>.
3. Configure the symbol to display the customized colours in the runtime environment by calling the *PLSAnimBusbar()* function, which is provided in PowerLogic SCADA. This function must be called from the Multi-State Condition area of the Symbol properties window.

   Each multi-state symbol can display up to seven colors (see *Changing the Colours of Busbars and Circuit Breakers* on page 97 for details). These symbols are labeled in binary with ABC and are passed to the *PLSAnimBusbar()* function as the first parameter. The second parameter is the busbar number that determines the animated colour. Always use the %variable% notation. Using this notation permits you to reuse genies.

   See the default symbols in the PowerLogic SCADA libraries that begin with PLS_.
4. Save the symbol and use it in a custom genie.

Deleting Information from PowerLogic SCADA

If you find you need to delete any data that you have entered (clusters, servers, genies, etc.), follow the normal procedure to delete the data, then use the Pack command to completely delete it. To do this, click File > Pack on the Project Editor.

Working with Alarms

In this section, you will find these topics:

Alarms Overview

Adding Setpoints and Delays

Making a PC-Based Alarm Digital

Setting Up an Alarm Based on an Enumeration

Changing an Alarm Severity

Enabling Waveforms for Onboard Alarms

Setting Parameters for Event Log Length and Historical Logging of Events

Adding an Onboard Alarm Tag

Setting Up Audible Alarms

Alarms Overview

This section discusses two alarm types: time stamped analog and time stamped digital. To access the alarms, from Citect Explorer, select the project folder, then click Alarms. In the right-hand pane, the alarm types display. Double-click the one you want to view/edit.

PC-Based Alarms

PC-based alarm tags are added in the Profile Editor, when adding each device profile. See *Adding, Editing, or Deleting a Device Profile* on page 45 for instructions. For instructions on entering setpoints and delays, see *Adding Setpoints and Delays* on page 92.

Onboard Alarms

If onboard alarms have been configured in a supported device, you can use the Profile Editor to map these alarms to digital time-stamped alarms in PowerLogic SCADA.

You cannot configure new onboard alarms from PowerLogic SCADA. You must add the alarm at the device, then you can create the alarm tag for it here. See *Adding an Onboard Alarm Tag* on page 94.

Adding Setpoints and Delays

There are two ways to add setpoints and delays for analog alarms. Before you enter setpoints and delays, ensure that you have configured the alarm server so that Publish Alarm Properties is set to TRUE.

- From the Analog Alarms window (accessible from the Project Explorer or Project Editor screens), you can type the setpoint and delay values for each alarm.
- When in the runtime environment, you can edit setpoints/delays that were set by the method above. See *Viewing Waveforms* on page 119 for instructions.

Also, set the following parameter to allow persisting of alarm parameters at runtime.

[Alarm] UseConfigLimits = 1

Making a PC-Based Alarm Digital

All PC-based alarms are by default analog in PowerLogic SCADA. For some devices, you may need to make an analog alarm trigger on a true-false state (digital). Follow these steps:

1. Open the analog alarm in PowerLogic SCADA.

2. To alarm when a value transitions from 0 to 1, set the High value to 0.

   To alarm when a value transitions from 1 to 0, set the Low value to 1.
3. In the Category field, ensure that the correct alarm level is entered (_PLSALM_HIGH, _PLSALM_MEDIUM, _PLSALM_LOW, _PLSALM_EVENT).
4. Replace the alarm.

Setting Up an Alarm Based on an Enumeration

For instructions on defining an enumeration in the Profile Editor, see *Defining an Enumeration* on page 55.

An example of an enumeration alarm is:

0 = unknown

1 = good

2 = warning

3 = alarm

To add an alarm that is based on an enumeration, follow these steps:

1. Open the analog alarm in PowerLogic SCADA.
2. To alarm on states 0, 2, and 3:

   Set Low = 1 (if the value < 1, the alarm indicates an unknown state)

   Set High = 1 (if the value > 1, the alarm indicates a warning)

   Set High High = 2 (if the value > 2, the alarm indicates an alarm)
3. In the Category field, ensure that the correct alarm level is entered (_PLSALM_HIGH, _PLSALM_MEDIUM, _PLSALM_LOW, _PLSALM_EVENT).
4. Replace the alarm.

Changing an Alarm Severity

To change the severity of an alarm, follow these steps:

1. Open the analog alarm in PowerLogic SCADA.
2. In the Category field, ensure that the correct alarm level is entered (_PLSALM_HIGH, _PLSALM_MEDIUM, _PLSALM_LOW, _PLSALM_EVENT).
3. Replace the alarm.

Enabling Waveforms for Onboard Alarms

To enable waveforms, making them available for viewing in the runtime environment, follow these steps:

1. At the device, or via the meter configuration software (PMCU), add the alarm and enable the automatic capture of a waveform when the alarm occurs.
2. In the Profile Editor, on the Create Device Profiles tab, for the same alarm you added in PMCU, check the Waveform box.

   When this alarm occurs at the device, the waveform will be captured. The files will be transmitted to PowerLogic SCADA and will be available for viewing. The amount of time this takes depends on the number of I/O servers you have and the number of serial devices on a chain. On a very large system with numerous serial devices, this could take as much as an hour.

You can view the waveform from the Alarm Log in the runtime environment. See *Using the Genie Status Page* on page 117 for instructions.

Setting Parameters for Event Log Length and Historical Logging of Events

You can use two parameters to determine the maximum number of entries in the Event Log and whether you want to log entries after they are FIFO'd out of the Event Log.

Event Storage: [Alarm]SummaryLength Parameter

The maximum number of alarms that be stored is controlled by the Alarm Summary length parameter, which defines the maximum number of alarm summary entries (Event Log entries) that can be held in memory. You can view these alarm summary entries on the alarm summary page. Each event requires 256 bytes of memory, plus the length of the comment. 32,000 entries will require at least 8 MB of memory. If you have many events, you should ensure that there is enough memory to store them in RAM.

The default value is 5000.

Event Logging: [PLSEventLog] Parameter

Use this parameter to determine whether you will log Event Log entries and the file format that will used for logging. If you choose to log, you must specify the file format: XML or CSV.

[PLSEventLog]

FIleFormat 0 = no file logging, 1 = CSV format, 2 = XML format (default = 0)

EntryCount number of entries logged per disk file (1-3200; default = 250)

ScanRate time in seconds between writing to file, when waiting for a new event (1 to 3200; default = 10); may be ignored if the entry count is reached or if a large number of alarms or events is triggered

The filename will have this format:

<Project Name>_<Cluster Name>_PLSEventLog.YYMMDDHHMMSS.xxx (where xxx is the file format selected). The log files will be stored in the Data folder.

[Path]

PLSEventLog=c:\SharedData\Alarm

To add custom alarm categories to the log file, use:

[AlarmFormat]

EventLog=OnDate | Date, OnTimeMS | Time, Custom1 | Equipment, Name | Description,
SumState | State | Custom2 | Location, UserName | User

Adding an Onboard Alarm Tag

When a device onboard alarm has not been included in PowerLogic SCADA, you can add it from the Profile Editor. You need to follow these steps to include the device's unique identifier. Otherwise, the alarm will not annunciate in the Graphics page.

NOTE: You can only add onboard alarms for devices using the CM4, PM8, Micrologic, or Sepam drivers. CM4, PM8, and Micrologic unique IDs must be decimal; SEPAM unique IDs must be hexadecimal.

1. From the device, obtain the unique identifier for this alarm. Additionally, for MicroLogic, you need to include the unique sub-identifier.
2. You also need the file number in which alarms on stored on the device.
3. From the Profile Editor, add the onboard alarm (see *Onboard Alarm Tag Addresses* on page 41 in Editing Tag Addresses.

Setting Up Audible Alarms

You can use a variety of Windows wave files for audible alarms. To enable this feature, follow these steps:

1. Define the alarm sound to be used and the repeat interval for each priority in the alarm you want to be audible. Enter the following information, either in the project parameters (Citect Project Editor > System > Parameters) or in the Citect.ini file:
   a. [Alarm]
   b. Sound<priority>=<wave file name>
   c. Sound<priority>Interval=<repeating interval in milliseconds>

   If the device is added through the Profile Wizard, the alarm priority will be 1, 2, and 3 for _PLSALM_HIGH, _PLSALM_MEDIUM, _PLSALM_LOW alarms respectively.

   The user can define specific wave files for the sounds, PowerLogic SCADA also supports the following Windows operating system sounds:

   SystemAsterisk

   SystemExclamation

   SystemQuestion

   SystemDefault

   SystemHand

   SystemExit

   SystemStart
2. After alarms are set up: When an alarm occurs, its specified alarm sound will play continually according to the specified interval. The alarm sound will stop when either:
   — the user clicks Silence Alarm on the alarm page
   — the alarm is acknowledged

Restoring and Backing Up Projects

In this section, you will find these topics:

Restoring a Project

Backing Up a Project

Restoring a Project

To restore a project, overwriting its current settings, follow these steps:

1. From the Citect Explorer, select the project you want to restore; click Tools > Restore.
2. In the Restore from box, browse to the location of the file you will use to restore.
3. In the To box, check Current Project.
4. In the Options box, check *Configuration files* to restore backed up INI files and the TimeSyncConfig.xml file (used to store time synchronization settings). If you backed up the sub-directories under the project, the directories will be listed under *Select sub-directories to restore*. You can restore all or no sub-directories, or you can select specific sub-directories to restore.
5. Click OK.

Backing Up a Project

1. From the Citect Explorer, click Tools > Backup.
2. At the Backup Project window, select the project you want to back up
3. Browse to the location where you want to store the backup file.
4. In the Options box, check “Save configuration files.” This saves the citect.ini file.
5. Click OK.

The backup .CTZ file is written to the location you chose. This is a Citect Zip file; you can open it with WinZip.

The Project Color Change Tool

Use this tool to assign colours for busbars and circuit breakers that are generated by genies in graphics pages. You will set separate colours for each project in your system. There are seven states (Error, Intermediate, etc.), for which you can assign colours. You can also select a second “flash” colour for any state; when the busbar or circuit breaker is in that state, the colour will alternate between the two colours.

It is best to set the project colours before you add genies to the project. A large number of standard genies (beginning with PLS_) are already available; however, you can create a custom genie and add it to the project. If you choose to add genies, follow the instructions in *Adding a Custom Genie for Use in the Color Change Tool* on page 97.

To begin changing colours, see *Changing the Colours of Busbars and Circuit Breakers* on page 97.

Adding a Custom Genie for Use in the Color Change Tool

To be able to change colours for a project, all of its genies must be set up in the manner described below. The standard PowerLogic libraries of genies already have this custom setting, but if you need to create a genie or use one that is not in the PowerLogic libraries, follow these steps:

1. From Citect Explorer, choose the project for which you want to set/change colors.
2. From the Graphics Builder, click File > New > Genie.
3. On the Objects toolbar, click the object you want to draw or insert, then click on the graphics page.
4. On the Properties page, set the type to *Multi-state* (this will be on one of two tabs: Appearance or Fill). Set its conditions and states, then click Apply. If you need help adding conditions and states, click Help.

   NOTE: The State colors that are shown on this tab are what you will be changing in the Project Color Change Tool.
5. Click the Access tab. In the Description field, type:

   <PwrLineColor>
6. Click OK to save the genie.
7. Open the graphics page, and drag and drop the genie onto the page.
8. To run the Project Color Change Tool, you need to close any open graphics pages.

Changing the Colours of Busbars and Circuit Breakers

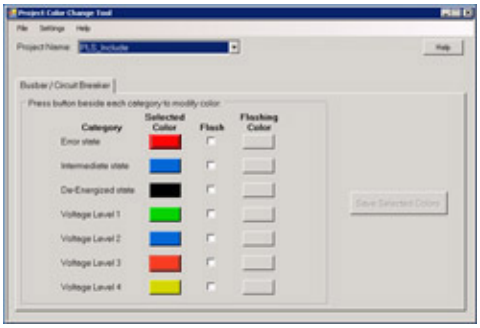
Open the Project Color Change tool: from the Citect Explorer screen, click Tools > Project Color Change Tool. Then follow these steps:

1. If a prompt displays telling you that Citect.ini location has changed, click OK. There is no need to locate the file.

   Colours will be changed for all genies in that project, assuming that the genie was set up with a Multi-state type and that it has <PwrLineColor> as its description on the Access tab.

   NOTE: If no projects display, one of two things is true: either your projects are “hidden” (in the Include folder) or the genies in the project do not have the <PwrLineColor> setting described in *Adding a Custom Genie for Use in the Color Change Tool* on page 97. If any genie on the page does not change colour, check to be sure that you have added the setting for that genie.
2. At the message telling you that PowerLogic SCADA is running and will be hidden, click OK.

The Project Color Change Tool displays (the colours you see may differ):



3. From the Project Name drop-down menu, choose the project for which you want to assign colours.
4. Click the selected colour for the first state or condition that you want to change:

Category	State of the Busbar or Circuit Breaker	PowerLogic SCADA Multi-State Colours*
Error State	The circuit breaker has lost communication.	ABC / **A**BC (colours 1 and 2)
Intermediate state	The circuit breaker is between states, e.g., being racked in or racked out.	A**B**C (colour 3)
De-Energized state	The circuit breaker is de-energized or inactive.	**AB**C (colour 4)
Voltage level 0	These levels correspond to the four voltage levels that (busbar voltage level), entered when adding a busbar to a drawing. These correspond to the levels being used by the standard that applies for each installation.	AB**C** (colour 5)
Voltage level 1		**A**B**C** (colour 6)
Voltage level 2		A**BC** (colour 7)
Voltage level 3		**ABC** (colour 8)

* The Symbol Properties screen is used when creating conditions for multi-state colours. See *Creating and Editing Genies* on page 90 for details on animating a custom genie.

5. From the Windows Color screen, choose the new colour, then click OK.
6. If you want busbars and circuit breakers in the Graphics page to flash when they are in a particular state, click Flash for that state, then add the “flash” colour. For example, if you set the error state colour to red, you might add a flash colour of green.

   When the graphics page is opened for viewing and busbars or circuit breakers are in that state, they will flash continuously between the two colours.
7. Continue to change state colours, or click Save Selected Colors to save your colour changes.
8. To update graphics pages with the new colours, go to the Graphics Builder. Click Tools > Update Pages.

The colours will be used for busbars and circuit breakers in all genies for this project (as long as the genie has *Multi-state* as its type and has the description of <PwrLineColor>.

Using Cicode to Customize a Project

Cicode is a programming language designed for use in PowerLogic SCADA to monitor and control plant equipment. It is a structured language similar to Visual Basic or 'C'. You need no previous programming experience to use it. However, it is assumed that you will have received Cicode training from PowerLogic SCADA before you attempt to use Cicode.

Using Cicode, you can access all real-time data (variables) in the PowerLogic SCADA project, and all PowerLogic SCADA facilities: variable tags, alarms, trends, reports, and so on. You can use Cicode to interface to various facilities on the computer, such as the operating system and communication ports. Cicode supports advanced features including pre-empted multitasking, multi threads, and remote procedure calls.

For information about driver-specific INI parameters that you can configure, see *Appendix 2: Using Citect INI Parameters* on page 137 .

For information about other parameters, see the CicodeReference help file, located in Program Files\Schneider Electric\PowerLogic SCADA 7.10\Bin.

Customizing PowerLogic SCADA Default Behaviors

In this section, you will find these topics:

Time Zone Settings

Time Synchronization

Graphics Library Parameters

Trend Tag Scan Intervals

Calculating Disk Storage for Trends

To learn about the PowerLogic SCADA parameters that you can use to configure drivers, see *Appendix 2: Using Citect INI Parameters* on page 137.

Time Zone Settings

The system allows you to interact with devices located in a different time zone than the PowerSCADA I/O Server. To do this, the system converts any alarm/waveform timestamp as well as the actual time sent within the Set Time command from / to the local time zone. In order to take into account daylight savings properly, the Windows time zones database is used. Thus, time zone names must be taken directly from this database (case-insensitive), otherwise the system will default to using the I/O server's local time zone. The Windows time zone database is located in the Windows registry in HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones\. Examples of time zone names are

- AUS Central Standard Time
- China Standard Time
- Pacific Standard Time (Mexico)

Time zones for remote devices can be specified on two levels:

1. Use section [ProtocolName.ClusterName.PortName.IODeviceName] to specify the time zone for a particular device, e.g.

   [PLOGIC870.Cluster1.PM870_Port.PM870_Device1]

   Timezone = Singapore Standard Time
2. Use general section [POWERLOGICCORE] to specify the time zone for all devices, e.g.

   [POWERLOGICCORE]

   Timezone = Mountain Standard Time

The device-specific time zone specification takes precedence. In other words, if both of the above examples are present in the Citect.ini file, the PM870_Device1 would be located in "Singapore Standard Time" time zone, and all the other I/O devices in the project would be located in "Mountain Standard Time" time zone.

If there is no time zone specification (or if it does not match the time zone from Windows database), the device would be in the same time zone as the machine where the I/O Server is running; thus, no time conversion will be done.

If only the first of the above examples is present within the Citect.ini file, the PM870_Device1 would be located in "Singapore Standard Time," and all the other devices use the current local time zone.

Time Synchronization

Current time can be sent to the corresponding device by means of Set Time command or (in case of Sepam) by writing directly to the corresponding registers within the

device. In addition to the manual procedure, this process can be scheduled to occur periodically (using PowerLogic SCADA events).

Non-manual time synchronization causes the Set Time command to be sent automatically, based on a device state or event originating from within the device.

Automatic time synchronization applies only to Micrologic and PM devices and takes place based on the following rules:

1. For Micrologic devices, the value of the top-most bit of the register 679 is examined (for both the Circuit Breaker Manager and the Chassis Manager). If the bit is equal to 1, it means that the device is out of sync and needs to be synchronized.
2. For PM devices, an alarm 50700 (“Unary Power Up / Reset”) indicates that the device needs to be synchronized. In addition, bit 6 of register 3055 of the device is examined. If this bit is equal to 1, the device has a real-time clock; so automatic time synchronization should never take place.

Graphics Library Parameters

Maximum number of entries that can be held in Event Log

The Alarm Summary length parameter in Citect.ini defines the maximum number of entries that can be held in the Event Log (default = 5000 entries). You can view all events in the Event Log and alarms in the alarm logs (Alarm Log, Unacknowledged Alarms, Disabled Alarms).

Each event requires 256 bytes of memory, plus the length of the comment. 32,000 entries will require at least 8 MB of memory. If you have many events, you should ensure that there is enough memory to store them in RAM.

After the parameter number is reached, older events are FIFO’d out to storage in [Installed Project Directory]\Schneider Electric\PowerLogic SCADA 7.10\Logs

Parameters for Alarm and Event States

[Alarm]

UseConfigLimits = 1

CacheLength = 2500

!Sound1 = <wave file name>

!Sound1Interval = <repeating interval in milliseconds>

!Sound2 = <wave file name>

!Sound2Interval = <repeating interval in milliseconds>

!Sound3 = <wave file name>

!Sound3Interval = <repeating interval in milliseconds>

[AlarmFormat]

EventLog=OnDate | Date, OnTimeMS | Time, Custom1 | Equipment, Name | Description,
SumState | State | Custom2 | Location, UserName | User

[AlarmStateText]

ON=<default text for ACTIVE state>

OFF=<default text for INACTIVE state>

ACK=<default text for ACKNOWLEDGED state>

ENA=<default text for ENABLE state>

DIS=<default text for DISABLE state>

CLE=<default text for CLEAR state>

These parameters are read only when the system starts up. The user must restart PowerLogic SCADA if they change these parameters.

If you do not specify any value for these parameters, these default values will be used, in this order:

Appearance

Disappearance

Acknowledge

Enable

Disable

Clear

[PLSEventLog]

FileFormat=0

EntryCount=250

ScanRate=10

[Path]

[PLSEventLog]=c:\SharedData\Alarm

Trend Tag Scan Intervals

When you select a trend tag for a device profile (Add/Edit Device Profile screen), the tag will be scanned at the “long” interval” (every 15 minutes, with FIFO storage of 12 months); but certain trend tags have an additional “short” scan interval. This interval is set by default at 5 seconds, with FIFO storage of two weeks.

The default tags are: Current A, Current B, Current C, Voltage A-B, Voltage B-C, Voltage C-A, Power Factor Total, Apparent Power Total, Reactive Power Total, Real Power Total, and Frequency. When you choose one of these tags for trending, you will get both long and short interval trending. The long interval trend will use the trend tag name from the Profile Editor. The short interval trend tag will have the same name as the long tag with an “s” appended to it.

You can edit the *Profile Editor.exe.config* file to add or delete tags that will have short scan intervals, and to change the short scan interval for all of the tags that are listed.

To edit short scan interval settings, follow these steps:

1. Locate and open *Profile Editor.exe.config* in Notepad:

   [Project Drive]\Documents and Settings\ All Users\Application Data\Schneider Electric\Profile Editor\v1.0

   Vista operating systems: [Project Drive]\ProgramData\Schneider Electric\PowerLogic\Profile Editor\v1.0)
2. To change the short scan interval, scroll the "TrendShortIntervalSamplePeriod" setting. The default value is 00:00:05, or five seconds (HH:MM:SS). Changing this rate will change the interval for all of the tags that are listed in the setting in step 3.
3. To change the tags that are included in the short scan interval, scroll to the "TrendShortIntervalTags" setting. The numbers listed (defaults: 1003,1004,1005,1050,1046,1042,1014,1015,1016,1001,1034) are the tag IDs. You can add or delete tags. Tag IDs are listed on the Define Device Type Tags tab (when the Advanced Properties option checked).

   NOTE: If you choose a device that includes the tags in this list, you will always have these short scan interval tags included.

For example, if you wanted to change the scan interval to ten seconds and add Overcurrent A for a CM4000, you would edit these two lines in this way:

"TrendShortIntervalSamplePeriod" value="00:00:10"

"TrendShortIntervalTags"
value="1003,1004,1005,1050,1046,1042,1014,1015,1016,1001,1034,19"

Calculating Disk Storage for Trends

There are two methods of calculating disk space usage for trends: scaled and floating point. The Profile Editor uses floating point by default. For more information on these calculations, see Calculating Disk Storage in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help).

Waveform Management

This chapter discusses how waveforms are stored and associated with alarms. In this section, you will find these topics:

Waveform Storage

Waveform to Alarm Linkage and Special Waveform Tags

Waveform Storage

Waveform records are organized within devices into files. These files are periodically checked for and downloaded as they appear on the device. When downloaded, on the PowerLogic SCADA I/O server the files are converted into a Comtrade format and then stored in a hierarchical fashion.

A single waveform will be stored as follows:

Citect Data *directory*> \ <ClusterName> \ <ProtocolName> \ <PortName> \ <IODeviceName> \ Waveforms \ <UTCTimestamp>.CFG

Citect Data *directory*> \ <ClusterName> \ <ProtocolName> \ <PortName> \ <IODeviceName> \ Waveforms \ <UTCTimestamp>.DAT

For example,

C:\Data\Cluster1\SEPAM\Sepam_Port\Sepam_IODev\Waveforms\

DST_00000000001203566197_0000000511.CFG

DST_00000000001203566197_0000000511.DAT

The CFG file is a **Comtrade configuration file**, and the DAT file is the **Comtrade data file**. Within the CFG file is a timestamp that reflects the device time start time of the waveform (this time is not adjusted to the I/O Server time zone or daylight saving, but it is stored per the device configuration). The file name has the UTC time in seconds since 1970 of the waveform.

The prefix of waveform file name reflects the type of the waveform. Currently, waveforms of the following types are supported:

DST_	Disturbance waveform
ADT_	Adaptive waveform
SST_	Steady state waveform

Waveforms of all mentioned types are currently supported by the PowerLogic driver for the CM4000, but only disturbance waveforms are linked to alarms see *Waveform to Alarm Linkage and Special Waveform Tags* on page 106.

If it is detected that the waveform data file has changed while it is being downloaded, the file gets discarded and is not stored on the IOServer.

Waveform to Alarm Linkage and Special Waveform Tags

PowerLogic SCADA associates a waveform file with an alarm based on the correlation between the alarm timestamp and the waveform timestamp. A waveform belongs to a given alarm if its timestamp falls within 10 seconds on either side of the alarm's timestamp. If more than one waveform falls within this range, the closest waveform time to alarm time will be chosen.

NOTE: If adaptive, steady-state, and disturbance alarms exist for the timeframe, disturbance alarms are given priority.

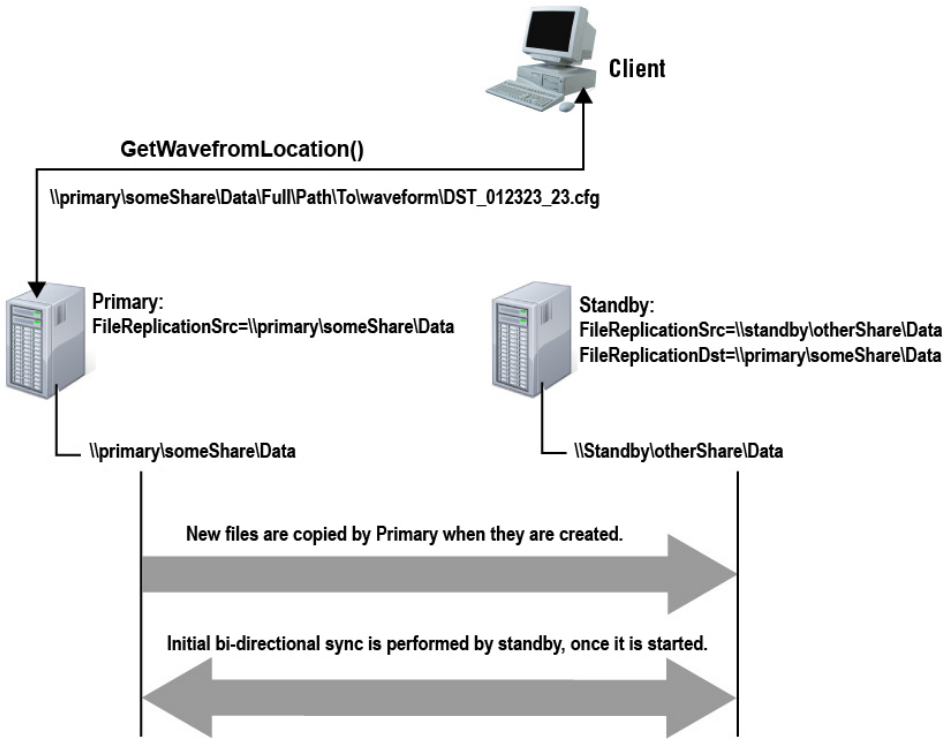
In addition, there are two special digital waveform tags defined (0 = FALSE, 1 = TRUE):

- **WaveformDownloading**: indicates whether a waveform file is currently being downloaded
- **WaveformCollectionEnabled**: indicates whether the waveform collection is enabled at all

File Replication parameters (see *Performance Tuning Parameters* on page 146) are vital for linkage to work. FileReplicationSrc not only defines the root directory where all waveform are stored, but it is also used when returning fully-qualified network waveform file names. The following requirements must be met in order to ensure replication and waveform linkage works:

- A network share must be created on both primary and secondary servers. This share can be read only for all users if network configuration allows servers to write into each other shares. A good idea may be to share the default PowerLogic SCADA data directory.
- FileReplicationSrc must define a path on local share where server will store all its files
- FileReplicaitonDst must be set to what FileReplicaitonSrc is set to on other server.

The figure below illustrates a configuration example and replication and linkage processes.



Graphics: Using the Runtime Environment

This chapter discusses the Graphics page in the runtime environment. In this section, you will find these topics:

Launching and Closing the Runtime Environment

Viewing the Graphics Page in the Runtime Environment

- *Logging On to the Runtime Environment*
- *Viewing the Interface*
- *Viewing the Alarms/Events Page*
- *Using the Analysis Page*
- *Using the Genie Status Page*

Multi-Monitor Support

Using the Graphics Test Mode

Using the Demo Mode

Launching and Closing the Runtime Environment

Launching Runtime: To launch the runtime environment, click Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Runtime. Or, from the Project Explorer, Project Editor, or Graphics Builder design-time page, click File > Run.

Closing Runtime: There are two ways to shut down the runtime environment:

- You can click the “X” in the upper right corner of the screen. You must enter a user name and password. (It is important to define at least one user in your project.)
- From the taskbar, click the Runtime icon:

  This displays the Runtime Manager screen. Click Shutdown All. There is no required user name/password. Unless you disable this feature on the taskbar, users can shut down the runtime environment without having to enter any information.

Viewing the Graphics Page in the Runtime Environment

In the runtime environment, the Graphics Page displays with the appearance that is determined by the menu.dbf file for this project. If you need to change the appearance of tabs and menus, see *Using Menu.dbf to Change the Graphics Page Appearance* on page 89.

In this section, you will find these topics:

Logging On to the Runtime Environment

Viewing the Interface

Viewing the Alarms/Events Page

Using the Analysis Page

Using the Genie Status Page

Logging On to the Runtime Environment

In the upper left-hand corner, click the Login link. Then enter your User ID and password. The features that are available will vary, depending on the user level.

Viewing the Interface

The runtime environment page includes a variety of tabs. Using the Normal template from the pls_include_1 library, you will see the following:

In the upper left corner of the screen, the project is listed. Your logon name is below that. At the top of the page, there are right and left navigation arrows. When there are multiple pages in a project, click them to scroll between pages.

There are two rows of tabs:

- The upper row is typically used for single lines, alarms/events, analysis (for trends), and system supervision (allows you to view the network connection topics).
- The lower row has links to options for the topic chosen from the upper row; for example, if you choose single lines on the upper row, the lower row shows all of the links to the individual single line pages.
- The small grey triangle on the right-hand side of the upper row of links (), is the reload button. When you make changes to the menu.dbf file, click this button to refresh the view in the runtime environment.

NOTE: When viewing the single lines: if the busbars and circuit breakers do not display with the expected colors, it could be that a custom genie is not set up correctly. If you are using a custom genie, verify that <PwrLineColor> has not been left out. See *Adding a Custom Genie for Use in the Color Change Tool* on page 97 for details on customizing genies and displaying colors.

When there is a communication loss for a device, the genie or any part of the genie on the one-line page will have cross-hatches (gray dots) over the affected area, and a communication loss (CL) indication displays on the genie. An alarm will also annunciate. The color state before communication was lost will remain unchanged.

Viewing the Alarms/Events Page

To view the alarms or events, click the Alarms/Events tab; then select the tab for the Alarm Log or Event Log. The Event Log displays all alarms and events that have occurred. The Alarms Log displays active enabled alarms. To sort a log, displaying information according to one of its columns, click the column title.

To filter the information that displays, click Filter, just above the Date column (see *Filtering and Freezing Log Views* on page 114 for more information). From the

Advanced Alarm Filter window, you can filter by date range, by text matches for various attributes, or by alarm type.

To freeze the view of the current log, click Freeze, in the upper right corner of the screen. No additional alarms will display, and none will drop out, until you click Unfreeze.

See *Changing the Display Name in an Alarm or Event Log* on page 115 for more information.

Viewing, Printing, and Export Options

Each of the logs includes these navigation buttons:

Button	Description
	Acknowledge Current Page of Alarms: Click to acknowledge all of the alarms that display on the current page.
	Silence Alarms: Click to silence all active alarms. This does not clear unacknowledged alarms or make alarms inactive; it only stops the audible portion of the alarm.
	Print/Export Alarms: Click to begin printing or exporting part or all of the log. Select All or the number of pages, then choose whether to print or export (to .txt file). When printing, the default location is: Windows XP: Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\Data Vista operating systems: ProgramData\Schneider Electric\PowerLogic SCADA 7.10\Data **Note**: When printing: To avoid truncation of data, choose the Landscape orientation.
	Save Columns: Click to save the configuration of columns that you have selected after you close the Alarm Log. You must be an administrator to do this.
	Add Columns: Click to add additional columns. Choose from the list. Columns will be placed on the right side of the table. See *Event/Alarm Log Columns Table* for a list of columns and names.
	Auto Fit Columns: Click to correct the alignment of columns, to ensure that they all fit on the page.
	Page Number: Display only; this is the page number that you are viewing.
	Page To Top and Page Up: Click to either go to the top of the log or to move up one page.
	Page Down: Click to move down one page.

Event/Alarm Log Columns Table

This table lists the column headings that are available for use when viewing the event log and alarm logs. The field name is the name that you select when you add a column. The display name is the default column heading in the log. To change the display name, see *Changing the Display Name in an Alarm or Event Log*.

Field Name	Default Display Name	Description
Tag	Tag	alarm tag
Name	Description	description of alarm e.g., Sag Vcn or Under Voltage B-C
Cluster	Cluster	cluster name to which the alarm belongs

Field Name	Default Display Name	Description
Date (OnDate: event log)	Date	date (MMDDYYYY) that alarm annunciated
Time (OnTime: event log)	Time	time (HH:MM:SS) that alarm annunciated
TimeMS (OnTimeMS: event log)	Time	time (HH:MM:SS:MS) that alarm annunciated
OffDate	OffDate	event log only: date (MMDDYYYY) that alarm dropped out
OffTIme	OffTime	event log only: time (HH:MM:SS) that alarm dropped out
AckDate	AckDate	event log only: date (MMDDYYYY) that the alarm was acknowledged
AckTime	AckTime	event log only: time (HH:MM:SS) that the alarm was acknowledged
AckTimeMS	AckTimeMS	event log only: time (HH:MM:SS:MS) that the alarm was acknowledged, with milliseconds
State	State	state of the alarm: disappearance, appearance defaults, or customized from the alarm description field
LogState	LogState	alarm logs only: active or inactive
SumState	State	event log only: state of the entry in the event log: disappearance, appearance
AlmComment	AlmComment	event log only: these entries come from the time-stamped digital alarm window “Comment” fields
Comment	Comment	event log only: displays comments from the alarm
UserName	User	user name associated with the event
FullName	FullName	full user name associated with the event
custom1	Equipment	default equipment name displays; used for alarm filtering and viewing
custom2	n/a: waveform information	used for waveform acquisition
custom3	n/a: metadata, categorization	These are all alarm/event attributes.
custom4	n/a: metadata, alarm type	
custom5	n/a: metadata, alarm group	
custom6	n/a: metadata, subcategorization	
custom7	n/a: metadata, alarm level	
custom8	n/a: user filter	used for customized filtering

Changing the Display Name in an Alarm or Event Log

Although you cannot delete column headings, you can edit them via the AlarmFormat Summary parameter. To do this:

1. Open your project.
2. Open the Project Editor. Click System > Parameters, From the Parameters window, scroll (do not use the dropdown list) to the Section Name *AlarmFormat*.
3. **For the Event Log**, scroll to display *Summary* in the Name field. The Value field lists all of the columns in the format X|Y, where X is the field name (the name you

select when adding columns to a log) and Y is the display name (as it appears in the column heading). Edit Y to change the heading that displays.

4. **For the Alarm Log, Unacknowledged Alarms Logs, and Disabled Alarms**, repeat this process, with *Active* chosen in the Name field.
5. After making any changes, compile and run the project. The headings will be changed when you view the logs from the Graphics Page.

The Event Log

The Event Log displays all alarms and events that have occurred.

The Alarm Log

The Alarm Log displays active enabled alarms. To sort on any of the columns, click the column title.

To filter the alarms that display, click Filter, just above the Date column (see instructions on using the filter option in *Filtering and Freezing Log Views* on page 114). From the Advanced Alarm Filter window, you can filter by date range, by text matches for various attributes, or by alarm type.

To freeze the view of the current log (this is also true for unacknowledged or disabled alarms), click Freeze in upper right corner of the screen. (Note that the button name changes to Unfreeze.) No additional alarms will display and none will drop out until you click Unfreeze.

Each alarm provides additional options. To view these options, right-click the alarm. Then you can:

- acknowledge or disable the alarm
- view alarm detail (similar to the genie status page in the single lines of the runtime environment)
- view waveforms: If this option does not display, there are no waveforms for this alarm.) Waveforms can display only if the device is set to “acquire on event,” and the waveform option is checked in the Profile Editor (see *Enabling Waveforms* on page 44).

  When the waveform is available for viewing, it displays when you click this link. Waveforms must be correctly set up before they will display. See *Enabling Waveforms for Onboard Alarms* on page 93 for more information. See *Viewing Waveforms* on page 119 for instructions on viewing waveforms.

  When you select the waveform option, you may see a message telling you “please try again after waveform has been acquired.” This means one of two things:

  — the alarm has been acquired at the device, but it has not yet been passed to PowerLogic SCADA

  — the device was not set to acquire a waveform, and the waveform option was checked in the Profile Editor

  NOTE: If there are multiple waveforms captures for this alarm, and if there is a disturbance waveform, it is the only one that is available here. If there are both an adaptive and transient, but no disturbance, the one with the earliest time stamp displays.

Unacknowledged Alarms and Disabled Alarms

As with the Alarm Log, these logs display either unacknowledged alarms or disabled alarms. The sort and filter options operate as they do in the Alarm Log.

Alarm and Events Logging

Alarms from the Event Log can be saved to a file on the alarm server, thus protecting them from being lost when the FIFO size is passed. This feature is disabled by default, but it can be enabled by setting the FileFormat INI parameter.

[PLSEventLog] FileFormat: Determines the file format to be used for logging alarm/event data to disk files.

Allowable Values:

0 - (Disable)

1 - (CSV)

2 – (XML)

Default value: 0

Filtering and Freezing Log Views

Filtering Information

To filter the information that displays in any of the alarm logs or the event log, click Filter (in the upper left corner of the screen). The Advanced Alarm Filter screen displays. The table below describes its settings.

Filter Option	Description: Display all alarms for:
Basic Filter box:	
Start Date/End Date	a date range. Choosing only a start date displays alarms from that date to the current date. Choosing only an end date displays alarms for the past year up to that date. For example, to display alarms only for today's date, enter only a start date.
Start Time/End Time	a time range. Choosing only a beginning time displays alarms from that time through the end of the day (23:59:59 or 11:59:59 p.m.). Choosing only an ending time from the start of the day (00:00:00 or 12:00:00 a.m.) through the time selected.
Tag	a single tag; use tag name only, do not include equipment name. For example, enter *MMXU1\A\phsA*, not *MainCM4\MMXU1\A\phsA*. To filter on tag and equipment, enter the tag here and the equipment in the Equipment Name field.
Equipment Name	a device (entered when using the Profile Wizard or Automation Interface; (listed in Citect Explorer > System > Equipment)
Cluster	a single cluster, which was added when setting up the project (listed in Project Editor > Servers > Clusters)
Alarm Name	the descriptive information in the Alarm Name field of the Time Stamped Digital Alarms page
Alarm Description	Alarm Desc from Time Stamped Digital Alarms: a customized on and off text description, such as "active" and "inactive"
Custom Filter 8	There are eight custom filters, which can be assigned by the customer in each alarm. A group of alarms in a specific location could have the same name in CUSTOM8 so that custom filtering can be easily applied. Custom8 has a default assignment of "Equipment." To change custom filter assignments, use the AlarmFormat parameter (Project Editor > System > Parameters).
Group Filter box:	

Filter Option	Description: Display all alarms for:
Categorization	These “alarm filters” are created in the Profile Editor when alarms are created.
Alarm Type	
Alarm Group	
Subcategorization	
Alarm Level	
Type Filter box: These are advanced PowerLogic SCADA topics; see PowerLogic SCADA help for more information.	
State	the state of the alarm
Type	the data type established for the alarm tag
Area	the area associated with the alarm
Category	the alarm category; there are four predefined categories (event, high, medium, low); additional categories can be added.
Priority	the priority of the alarm category; not used in the default PLS_Include project

Freezing the View

To freeze the view in an alarm window, click Freeze in the upper left corner of the screen. This “locks” the alarm page so that only the alarms that are currently displayed will remain in view. The button name changes to “Unfreeze.” Alarms may continue to occur, but they will not display until you click “Unfreeze.”

Changing the Display Name in an Alarm or Event Log

Although you cannot delete column headings, you can edit them via the AlarmFormat Summary parameter. To do this:

1. Open your project.
2. Open the Project Editor. Click System > Parameters, From the Parameters window, scroll (do not use the dropdown list) to the Section Name *AlarmFormat*.
3. **For the Event Log**, scroll to display *Summary* in the Name field. The Value field lists all of the columns in the format X|Y, where X is the field name (the name you select when adding columns to a log) and Y is the display name (as it appears in the column heading). Edit Y to change the heading that displays.
4. **For the Alarm Log, Unacknowledged Alarms Logs, and Disabled Alarms**, repeat this process, with *Active* chosen in the Name field.
5. After making any changes, compile and run the project. The headings will be changed when you view the logs from the Graphics Page.

The Event Log

The Event Log displays all alarms and events that have occurred.

The Alarm Log

The Alarm Log displays active enabled alarms. To sort on any of the columns, click the column title.

To filter the alarms that display, click Filter, just above the Date column (see instructions on using the filter option in *Filtering and Freezing Log Views* on page 114). From the Advanced Alarm Filter window, you can filter by date range, by text matches for various attributes, or by alarm type.

To freeze the view of the current log (this is also true for unacknowledged or disabled alarms), click Freeze in upper right corner of the screen. (Note that the button name changes to Unfreeze.) No additional alarms will display and none will drop out until you click Unfreeze.

Each alarm provides additional options. To view these options, right-click the alarm. Then you can:

- acknowledge or disable the alarm
- view alarm detail (similar to the genie status page in the single lines of the runtime environment)
- view waveforms: If this option does not display, there are no waveforms for this alarm.) Waveforms can display only if the device is set to “acquire on event,” and the waveform option is checked in the Profile Editor (see *Enabling Waveforms* on page 44).

  When the waveform is available for viewing, it displays when you click this link. Waveforms must be correctly set up before they will display. See *Enabling Waveforms for Onboard Alarms* on page 93 for more information. See *Viewing Waveforms* on page 119 for instructions on viewing waveforms.

  When you select the waveform option, you may see a message telling you “please try again after waveform has been acquired.” This means one of two things:

  — the alarm has been acquired at the device, but it has not yet been passed to PowerLogic SCADA

  — the device was not set to acquire a waveform, and the waveform option was checked in the Profile Editor

  NOTE: If there are multiple waveforms captures for this alarm, and if there is a disturbance waveform, it is the only one that is available here. If there are both an adaptive and transient, but no disturbance, the one with the earliest time stamp displays.

Unacknowledged Alarms and Disabled Alarms

As with the Alarm Log, these logs display either unacknowledged alarms or disabled alarms. The sort and filter options operate as they do in the Alarm Log.

Alarm and Events Logging

Alarms from the Event Log can be saved to a file on the alarm server, thus protecting them from being lost when the FIFO size is passed. This feature is disabled by default, but it can be enabled by setting the FileFormat INI parameter.

[PLSEventLog] FileFormat: Determines the file format to be used for logging alarm/event data to disk files.

Allowable Values:

0 - (Disable)

1 - (CSV)

2 – (XML)

Default value: 0

Using the Analysis Page

The Analysis Page offers two options for viewing data trends. In either option, you must select the tag(s) that are to be included. To be available for viewing in trends, a tag must be included in a device profile, and it must have the “Trend Tag” box checked (see *Adding, Editing, or Deleting a Device Profile* on page 45 for more information).

NOTE: Trend data is automatically logged when you check “Trend Tag” for tag and then add it to the PowerLogic SCADA project. If too many tags are chosen as trend tags, it could cause the hard drive to fill up. There are two methods of calculating disk space usage: scaled and floating point. For more information on these calculations, see Calculating Disk Storage in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help).

- Trending: Use this option to view historical trends. To select tags, click the Add Pen button on the toolbar:

  Then associate the pen with a tag. By default, most trend data is polled every 15 minutes, and it is stored for one year (Citect Project Editor > Tags > Trend Tags) or until it is FIFO’d out. Some tags are polled every 5 seconds and are stored for two weeks. These tags are:

  Current A
  Current B
  Current C
  Apparent Power Total
  Reactive Power Total
  Real Power Total
  Voltage A-B
  Voltage B-C
  Voltage C-A
  Frequency
  Power Factor Total

- Instant Trend: Use this option to view real-time trends. This allows viewing of data that is not set up for storage. To select tags for this trend, click the Instant Trend Selector on the toolbar:

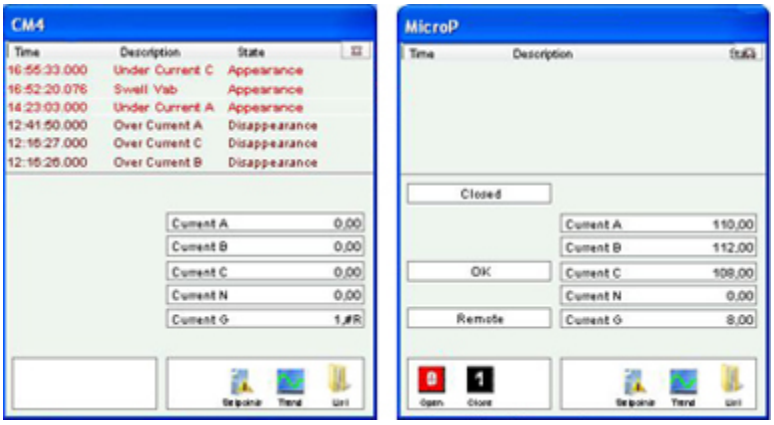
  

In either trending option, click Help for help using the tool:

Using the Genie Status Page

You must be logged in to the runtime environment to be able to use this page. Some features are only available to certain user levels (see *Adding and Modifying User Accounts* on page 127 for user access levels). To view this page, click on a genie.

One of two status pages displays. The page on the left illustrates the status page for a meter genie. The page on the right illustrates the status page for a circuit breaker genie.



At the top of the page, the most recent alarms and events are listed (racked in/out, Comms Loss, etc.). To view details about an individual alarm or event, right-click the alarm. You can view:

- a waveform. (If you do not see “Waveform” in the list when you right-click the alarm, there are no waveforms for this alarm.) Waveforms can display only if the device is set to “acquire on event,” and the waveform option is checked in the Profile Editor (see *Enabling Waveforms* on page 44).

  When the waveform is available for viewing, it displays when you click this link. Waveforms must be correctly set up before they will display. See *Enabling Waveforms for Onboard Alarms* on page 93 for more information. See *Viewing Waveforms* on page 119 for instructions on viewing waveforms.

  When you select the waveform option, you may see a message telling you “please try again after waveform has been acquired.” This means one of two things:

  — the alarm has been acquired at the device, but it has not yet been passed to PowerLogic SCADA

  — the device was not set to acquire a waveform, and the waveform option was checked in the Profile Editor

  NOTE: If there are multiple waveforms captures for this alarm, and if there is a disturbance waveform, it is the only one that is available here. If there are both an adaptive and transient, but no disturbance, the one with the earliest time stamp displays.
- details about the device (currents, voltages, powers, resets, others.),
- Finally, you can acknowledge or disable the alarm. Acknowledged and disabled alarms are moved to their own sub-tabs.

On the left side of the of the status page, status messages display, based on the tags defined for equipment referenced in this genie. The list varies, depending on the device. Possible tags are:

XCBR1\Pos Position (circuit breakers only)

XCBR1\CCBRkdPos Racked Out (circuit breakers only)

XCBR1\CBRkdPos Matching Fault/Trip Circuit Supervision (circuit breakers only)

XCBR1\Loc Local/Remote (circuit breakers only)

XCBR1\ESwPos Earth Switch (circuit breakers only)

PTRC1\Op Tripped

LPHD1\PhyHealth Communication Failure

NOTE: For MicroLogic Type P devices, circuit breaker status fields will display #COM if the device does not have a CCM. Thus, you should not add any tags that refer to the CCM, such as Racked In/Racked Out.

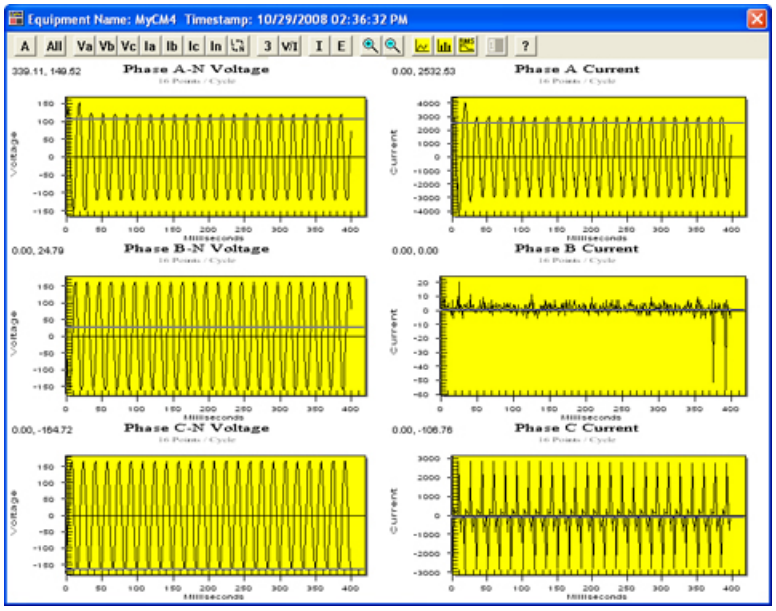
On the right side of the page, real-time values will display for the tag type that you chose in the Value field when you added the genie in the design-time mode. For example, if you enter MMXU1\A\phsA as the value, you will see real-time currents here, as illustrated above. If you did not enter anything in the Value field when adding the genie, this area will be blank.

At the bottom left corner of the circuit breaker status page, Open, Close, Reset (for circuit breakers).

At the bottom right corner, are the Setpoints, Trend, and List options. See the following sections for descriptions.

Viewing Waveforms

After you select a waveform for viewing from the genie status page (and after the waveform has been sent to PowerLogic SCADA), the waveform displays, as shown below. If multiple waveform types are captured for an alarm from a device, the disturbance waveform type is given preference.



The toolbar at the top of the screen includes these buttons:

Button	Description
A	Not currently used.
All	Click to display all graphs (all three voltages and all three currents of the waveform capture.
Va Vb Vc	Click to display individual voltage phases (Va = A-N, Vb = B-N, Vc = C-N).
Ia Ib Ic In Ig	Click to display individual current phases (Ia = phase A, Ib = phase B, Ic = phase C, In = neutral, Ig = calculated ground current).
	Click to toggle between line-to-line and line-to-neutral voltage phases.
3	Click to view all three voltage or all three current phases within the same graph. Phase A, B, and C voltages display together. Phase A defaults to black, B to blue, C to red. In the same way, phase A, B, and C currents display together (A is black, B is blue, and C is red).

Button	Description
	To change colours, right-click the graph title. Select Control Properties. Go to the Colours tab and make changes.
V/I	Click to view each phase, both voltage and current, within a single graph. You see phase A voltage and current in one graph, phase B voltage and current in another, and phase C voltage and current in a third.
I	Click to display the Import Waveform dialog. If you have SMS installed on the computer, it defaults to the Comtrade folder. If not, it goes to the root of the C drive, for you to locate the waveform you want to import. Double-click the waveform to view it.
E	Click to begin exporting a waveform. Select the type of file you will save (bitmap, text, or metafile). Then select the location where you want to store the waveform you are viewing.
	Click to zoom in or out. When zoomed in, there is a scroll bar at the bottom of each graph, used to move to the left or right.
	Click to view the display as a waveform plot.
	Click to view the display as a spectral plot. This feature does not work for the "3" or "V/I" views.
RMS	Click to view the display as an RMS plot.
	Select any phase; then click here to view the digital status of any of the digital channel(s) that have been set up for this circuit monitor.
?	Not currently used.

Entering Setpoints for Alarms

NOTE: You are only able to enter setpoints here if the Publish Alarm Properties field is set to TRUE when you set up the alarm server. See *Before You Use the Profile Wizard* on page 68 for instructions.

To add setpoints for alarms, follow these steps:

1. View the page, then click the genie for which you want to enter setpoints. A status window displays with the name of the genie.
2. Click Setpoints, then choose Analog, Digital, or All. When the Alarm Setpoints screen displays, select the first value you want to change. At the “keypad” screen (see below), enter the new value. Click OK to save it. Do this for each setpoint that you want to change.

Based on these setpoints, alarms can begin to display both in the alarms window at the top of the runtime screen and on the Alarms/Events tab (assuming you have set one up for this installation).

When there is a comms loss for a device: the last state before the loss happened is displayed on the screen.

Viewing Real-Time Trends

This option displays an historical trend. The data that displays is determined by the value that was selected in the Value Type field when this genie was added to the one-line page.

To view a trend: From the one-line page in the runtime environment, click a genie to view its status window. Click Trend, in the lower right corner. The Analyst screen displays for that trend.

You can select the timeframe for the trend. You can also uncheck phases to remove them from the trend, or highlight a phase to bring it to the front of the trend. For detailed information about the buttons on the screen, click “?” at the top of the page.

Viewing Lists of Real-Time Information for the Genie

To view lists of real-time currents, voltages; powers; resets and controls; and miscellaneous readings, click List, in the lower right corner.

Depending on the list item you choose, you can view all of that single category of settings for the genie.

For resets and controls, which are interactive, you should assign users a high level of security. For a list of the default user levels, see *Adding and Modifying User Accounts* on page 127. That link also includes information on creating unique users.

Multi-Monitor Support

PowerLogic SCADA provides the means for viewing the runtime environment from multiple monitors. This setup allows you to view unrelated information at separate monitors that are placed beside each other. In this example, there are four monitors, configured left to right in a single row:

Main Monitor Monitor 2 Monitor 3 Monitor 4

(displays Windows Start button)

In the Citect.ini file for the project (Windows XP: Documents and Settings\All Users\Application Data\Schneider Electric\PowerLogic SCADA 7.10\User\[projectname] or Vista: ProgramData\Schneider Electric\PowerLogic SCADA 7.10\User\[projectname]), add the following instructions:

[MultiMonitors]

Monitors = 4

DesktopWidth = 5120

DesktopHeight = 1024

ScreenWidth = 1280

ScreenHeight = 1024

StartupPage0 =

StartupPage1=

StartupPage2=

StartupPage3=

where

Monitors = x, y; x = number monitors in a row, y = number of rows (option)

DesktopWidth = width of the desktop (in the above example the number 5120 = 1280 * 4

DesktopHeight = Height of the desktop (in the above example it is 1024; in scenario 3 it will be 1024 *2 = 2048)

DesktopPosition = (the position of the main monitor. In the above example it is 0,0 i.e 0 row and 0 col. In scenario 2 it will be 0,1). Not typically needed, not used above.

ScreenWidth, ScreenHeight = screen width of each monitor (typically all monitors should be of same width and height)

StartupPage0 = startup page that needs to be displayed on monitor 1

StartupPage1 = startup page that needs to be displayed on monitor 2

StartupPage2 = startup page that needs to be displayed on monitor 3

StartupPage3 = startup page that needs to be displayed on monitor 4

Using the Graphics Test Mode

The Graphics Test Mode is only available in the pls_example project. This mode can be enabled by setting the following parameter in the project parameters:

Section Name: PowerLogicSCADA

Name: GraphicsTest

Value: Enable Graphics Test Mode

To disable, remove this parameter information.

The Graphics Test Mode is designed to enable off-line testing of animation in single-line drawings. This feature is useful for project development, prior to commissioning a system. Do not use this mode with physical devices that are connected to a live environment.

NOTE: Although it may appear that what you enter here will change settings at a device, these changes are all internal to the Graphics Test Mode. Nothing that you enter here will change settings at any device.

⚠CAUTION

UNINTENDED OPERATION

Do not use the Graphics Test Mode with physical devices in a live environment.

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

There are 55 generic devices available for Graphics Test Mode. These devices have this naming convention:

Breaker_1 through **Breaker_50**

and

Meter_1 through **Meter_5**.

Using the Demo Mode

You can run PowerLogic SCADA without the hardware key in demonstration (Demo) mode. In demonstration mode, you can use all PowerLogic SCADA features normally, but with restricted runtime and I/O.

If you configure PowerLogic SCADA to run as multiple processes on one CPU or multiple CPUs, you cannot use PowerLogic SCADA in demo mode; but if you run PowerLogic SCADA as one process, you can use a demo mode. The following demonstration modes are available:

- 15 minutes with a maximum of 50,000 real I/O.
- 10 hours with no static points and a maximum of one dynamic real I/O. This is useful for demonstrations using memory and disk I/O. PowerLogic SCADA starts in this mode if no static points are configured.

  If you want to demonstrate DDE, CTAPI, or ODBC writes to PowerLogic SCADA in this mode, you can only write one point. To write to more than one point, you must force PowerLogic SCADA to start in 15 minute-50,000 I/O demo mode by creating at least one static I/O point.

  For this to work, you must configure a real variable tag, with an accompanying PLC or I/O device. The tag must be used by a page or in Cicode. If you do not have a real I/O device connected, PowerLogic SCADA displays a hardware error, which you can disable using the IODeviceControl function.
- 8 hours with a maximum of 42,000 real I/O. This is only available through special PowerLogic SCADA Integration Partners (CIP) keys.

Assigning and Controlling User Privileges

NOTE: You need to give users appropriate levels of access, depending on the work they will do. For safety reasons, only advanced users should be given access to such features as controls and resets.

This chapter discusses how to establish user security for the system. This section includes the following topics:

Default User Access Settings

Adding and Modifying User Accounts

Changing Access Rights

Default User Access Settings

The following table describes the access rights to the runtime environment for each default access level. These access levels are included in PLSSecurity.ci file (in the PLS_Include project). Any changes made to this file will be overwritten when PLS_Include is updated (every new release). You need to document every change you make, so that you can update PLSSecurity.ci when PLS_Include is upgraded.

Access Right		User Levels / Global Privileges				
Description	ID	Operator1	Operator2	Engineer	Administrator	Kernel
		2	4	6	7	8
Circuit Breaker, Switch control	PL_Sec_CBControl		X	X	X	X
Alarms acknowledgment	PL_Sec_AlmAck	X	X	X	X	X
Alarm configuration	PL_Sec_AlmConfig			X	X	X
Add/remove log-in users	PL_Sec_ UserConfig				X	X
Shutdown Runtime	PL_Sec_Shutdown			X	X	X

NOTE: Levels 1, 3, and 5 are currently unused. You can use them as you wish. See *Changing Access Rights* on page 128 for instructions on changing access level privileges.

A method of limiting access to the design time/configuration environment is to remove the user rights to certain PowerLogic SCADA files. From the server, use Windows security to remove user accounts from individual features. For example, access to the following list of .exe files should be restricted to users who have design time/configuration privileges.

- CtDraw32.exe - Graphics builder
- CtEdit32.exe - Project Editor
- CtExplor.exe - Project Explorer
- ProfileWizard.exe - Profile Wizard
- ColorSwap.exe - Color Swap Tool
- ProfileEditor.exe - Profile Editor

All of these files need to have the same level of security, as they are interrelated.

Additionally, to prevent users from accessing and changing the code responsible for enforcing user security in the runtime environment, you need to lock down the user rights for the Cicode files (.ci extension) in the PLS_Includes project.

NOTE: Another way to lock users out from changing an existing project is to implement read-only projects.

Adding and Modifying User Accounts

You must add at least one user to any project before you can run and view it.

The Users screen controls all of the user access levels for each project. To assign PowerLogic SCADA user access, follow these steps:

From the PowerLogic SCADA Explorer, choose the project for which you want to assign user access.

1. From the Project Editor, click System > Users.
2. In the Users window, add the following for each user:
   a. User Name: to be used at login
   b. Full Long: displays in the runtime environment, in the login area.
   c. Password and Confirm Password
   d. Global Privilege (see the table above for access levels)
3. After you add the first user, click Add.
4. To add additional users, do the following: with a user displayed, click Add (which creates a copy of the user); type the new user information in place of the old information; then click Replace to overwrite the information.

   Note that the record count increases by one with each addition. To view the users one at a time, scroll through the list.

Adding Windows Integrated Users

Instead of using the above method to define users in the project, you can incorporate PowerLogic SCADA users and security options with the standard Windows security system. Using the integrated Windows security feature, the Windows user can log on to PowerLogic SCADA runtime with runtime privileges and areas configured within the project. For a Windows user to be able to log on to runtime, it must be linked to a PowerLogic SCADA "role," which is defined in the project with associated privileges.

To assign Windows integrated user access,

To link a Windows user to a PowerLogic SCADA role, you should add the "role" that specifies the Windows group of which the Windows user is a member.

The pre-existing AutoLogin capability is extended to include the client, when the user is a Windows user, having an associated PowerLogic SCADA role. In order to invoke this functionality for a Windows user, you need to set the [Client]AutoLoginMode parameter in the Citect.ini file.

When a PowerLogic SCADA is also included as a Windows user, the PowerLogic SCADA user takes priority at runtime. However, if a valid PowerLogic SCADA user login fails for some reason, the Windows user credentials will not be checked and an alert will be generated to advise that the login was not effective.

For more information, see Using PowerLogic SCADA with Windows Security in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help).

Changing Access Rights

You can edit the default access rights to each of the eight levels, thus changing the privileges that are available at each level. This is done in the PLSSecurity.ci file (in the PLS_Include project).

To change access rights:

1. Open PLSSecurity.ci.
2. Locate the AccessRights section.
3. For each right that you want to add (for example, to add access privileges for working with switches), add a new CASE.
4. Save and close PLSSecurity.ci.

NOTE: PLSSecurity.ci is overwritten any time that the PLS_Include project is overwritten. This happens with every release of PowerLogic SCADA, including service packs. To ensure that you do not lose changes that you enter, you need to:

- Note the changes that you make. Keep a copy of these changes
- Re-merge the changes every time that the PLS_Include project is overwritten.

Appendix 1: Default Genie Library

The genie library includes a number of general genies for objects such as motors and pumps. There are also genies that are specific to PowerLogic. These genies use a particular naming convention, which is described in the table below. In the PowerLogic library, each genie name begins with “pls,” and is followed by a description of the type of genie according to this table:

first	second	third	fourth
pls indicates PowerLogic library	alarm = alarm	base = primitive genies	1 = small
	ansi = ANSI style	cb = circuit breaker	2 = large
	display = equipment details	sw = switchgear	
	gen = generic	cmd = control genies	
	iec = IEC style	eq = equipment (devices)	
	style = navigation		

Additionally, the actual genies have abbreviated names. When you highlight a genie name, the abbreviation displays above the top row of genie icons.

The following tables list and define the individual genies in each of the PowerLogic libraries.

PLS_ALARM

This library includes genies that provide functionality to alarm displays.

Genie Abbreviation	Description
Field	data portion of an alarm row
Row	a genie with a collection of fields
Selector	for column resizing
Setpoint	a setpoint row

PLS_ANSI_BASE_1 / PLS_ANSI_BASE_2

These libraries include a variety of base symbols used to created genies for ANSI equipment.

1 = small size

2 = large size

Genie Abbreviation	Description
sl_battery_gen	single-cell battery
sl_battery_multi	multiple-cell battery
sl_capacitor	capacitor
sl_capacitor_vari	variable capacitor
SL_Closed_HV	closed circuit breaker position
sl_closed_knife	closed knife switch
sl_closed_lv	closed low-voltage circuit breaker
sl_conductive_path_1	conductive path 1

Genie Abbreviation	Description
sl_conductive_path_2	conductive path 2
sl_conductive_path_3	conductive path 3
sl_conductor_junction	conductor junction
sl_contact_nc_closed	contact break, closed
sl_contact_nc_open	contact break, open
sl_contact_no_closed	contact make, closed
sl_contact_no_open	contact make, open
sl_contact_term	contact terminal
sl_ct	current transformer
sl_in_cb_rack	incoming, circuit breaker racked out, plug
SL_In_Rack	incoming, circuit breaker racked out, socket
sl_in_sw_head	incoming switch head
sl_inductor	inductor
sl_inductor_adjust	adjustable inductor
sl_inductor_gen	general inductor
sl_inductor_magcore	magnetic core inductor
sl_inductor_vari	variable inductor
sl_open	open symbol
sl_open_fuse_sw	open isolating fuse-switch
SL_Open_HV	open high-voltage circuit breaker
sl_open_knife	open knife-type switch
SL_Open_LV	open low-voltage circuit breaker
sl_out_cb_rack	outgoing, circuit breaker racked out, plug
SL_Out_Rack	outgoing, circuit breaker racked out, socket
sl_pb_break	push-button, break
sl_pb_make	push-button, make
sl_pb_term	push-button, terminal
sl_pt	potential transformer
sl_relay	relay
sl_resistor	resistor
sl_resistor_adjust	adjustable resistor
sl_resistor_vari	variable resistor
sl_separable_con_closed	separable connector, closed
sl_separable_con_open	separable connector, open
sl_separable_con_plug	separable connector plug
sl_separable_con_socket	separable connector socket

PLS_ANSI_CB_1 / PLS_ANSI_CB_2

These libraries include genies for ANSI-type high-voltage and low-voltage drawout circuit breakers.

1 = small size

2 = large size

Genie Abbreviation	Description
hv_cb_bus_dr_c	high-voltage drawout circuit breaker, remote control
hv_cb_bus_dr_nc	high-voltage drawout circuit breaker, not remote
hv_cb_bus_nd_c	high-voltage non-drawout circuit breaker, not remote
hv_cb_bus_nd_nc	high-voltage non-drawout circuit breaker, not remote
hv_cb_fd_dr_c	high-voltage, drawout circuit breaker, remote control, with earth at bottom
hv_cb_fd_dr_nc	high-voltage, drawout circuit breaker, no remote control, with earth at bottom
hv_cb_fd_nd_c	high-voltage, non-drawout circuit breaker, remote control, with earth at bottom
hv_cb_fd_nd_nc	high-voltage, non-drawout circuit breaker, no remote control, with earth at bottom
hv_cb_inc_dr_c	high-voltage, drawout circuit breaker, remote control, with earth at top
hv_cb_inc_dr_nc	high-voltage, drawout circuit breaker, no remote control, with earth at top
hv_cb_inc_nd_c	high-voltage, non-drawout circuit breaker, remote control, with earth at top
hv_cb_inc_nd_nc	high-voltage, non-drawout circuit breaker, no remote control, with earth at top
hv_cb_nes_dr_c	high-voltage drawout circuit breaker, remote control, no earth
hv_cb_nes_dr_nc	high-voltage drawout circuit breaker, no remote control, no earth
hv_cb_nes_nd_c	high-voltage non-drawout circuit breaker, remote control, no earth
hv_cb_nes_nd_nc	high-voltage non-drawout circuit breaker, no remote control, no earth
lv_cb_bus_dr_c	low-voltage drawout circuit breaker, remote control, busbar-type with earth at bottom
lv_cb_bus_dr_nc	low-voltage drawout circuit breaker, no remote control, busbar-type with earth at bottom
lv_cb_bus_nd_c	low-voltage non-drawout circuit breaker, remote control, busbar-type with earth at bottom
lv_cb_bus_nd_nc	low-voltage non-drawout circuit breaker, no remote control, busbar-type with earth at bottom
lv_cb_fd_dr_c	low-voltage drawout circuit breaker, remote control, earth on load side (bottom of drawing)
lv_cb_fd_dr_nc	low-voltage drawout circuit breaker, no remote control, earth on load side (bottom of drawing)
lv_cb_fd_nd_c	low-voltage non-drawout circuit breaker, remote control, earth on load side (bottom of drawing)
lv_cb_fd_nd_nc	low-voltage non-drawout circuit breaker, no remote control, earth on load side (bottom of drawing)
lv_cb_inc_dr_c	low-voltage drawout circuit breaker, remote control, earth on feeder (top of drawing)
lv_cb_inc_dr_nc	low-voltage drawout circuit breaker, no remote control, earth on feeder (top of drawing)
lv_cb_inc_nd_c	low-voltage non-drawout circuit breaker, remote control, earth on feeder (top of drawing)
lv_cb_inc_nd_nc	low-voltage non-drawout circuit breaker, no remote control, earth on feeder (top of drawing)
lv_cb_nes_dr_c	low voltage drawout circuit breaker, no earth, remote control
lv_cb_nes_dr_nc	low voltage drawout circuit breaker, no earth, no remote control
lv_cb_nes_nd_c	low voltage non-drawout circuit breaker, no earth, remote control
lb_cb_nes_nd_nc	low voltage non-drawout circuit breaker, no earth, no remote control

PLS_ANSI_SW_1 / PLS_ANSI_SW_2

These libraries include ANSI-style switches:

1 = small size

2 = large size

Genie Abbreviation	Description
sw_fused	switch: feeder, fused
sw_fused_isolated	switch: feeder, fused, isolated
sw_general	switch: feeder, general
sw_knife	switch: knife type

PLS_DISPLAY

This library includes two genies that provide data row items for equipment.

Genie Abbreviation	Description
equiplistitem	data row for the equipment tag list
EquipValueItem	data row for the equipment popup

PLS_GEN_BASE_1 / PLS_GEN_BASE_2

These libraries include a variety of "parts" related to generators, motors, and transformers.

Genie Abbreviation	Description
chassis_ground	chassis ground
Dev_Base	device base
es_inc	earth switch, incomer
es_out	earth switch, feeder
Gen_1	generator, option 1
Gen_2	generator, option 2
gen_AC	generator: AC
gen_DC	generator: DC
genset	engine-generator
ground	ground
Motor_1	motor, option 1
Motor_2	motor, option 2
motor_ac	motor: AC
motor_dc	motor: DC
motor_synch	motor: synchronous
SL_Base	circuit breaker base symbol
sl_br_in	circuit breaker line in, non-drawout
sl_br_out	circuit breaker line out, non-drawout
SL_Bustie	bus tie
SL_CommLoss	comms loss
SL_Discrepancy	position discrepancy

Genie Abbreviation	Description
sl_harmonic_filter_1	harmonic filter 1
sl_harmonic_filter_2	harmonic filter 2
SL_In	incoming bus
SL_Local	local, rather than remote control
SL_Out	feeder
SL_Tripped	tripped
Test_CB_Control	health test for the circuit breaker control
transformer_1_in	transformer 1: general, on-line
transformer_1_in_y	transformer 1: star (wye), on-line
transformer_1_out	transformer 1: general, off-line
transformer_1_out_d	transformer 1: delta, off-line
transformer_1_out_y	transformer 1: star (wye), off-line
transformer_2_in	transformer 2: general, on-line
transformer_2_in_Y	transformer 2: star (wye), on-line
transformer_2_out	transformer 2: general, off-line
transformer_2_out_D	transformer 2: delta, off-line
transformer_2_out_Y	transformer 2: star (wye), off-line (no 2 IN D? or 1 IN D?)

PLS_GEN_CMD_1 / PLS_GEN_CMD_2

These libraries include genies that control display of popups and values:

1 = small size

2 = large size

Genie Abbreviation	Description
CmdDetail	provides access to the equipment detail popup
cmddetail_meter	provides access to the meter detail popup
Control	control in a circuit breaker
value	value section of a circuit breaker
value_meter	value section of a meter

PLS_GEN_EQ_1 / PLS_GEN_EQ_2

These libraries include the general equipment used to make up generators, motors, and transformers:

1 = small size

2 = large size

Genie Abbreviation	Description
busbar_horz	horizontal busbar
busbar_vert	vertical busbar
gen_ac	generator: AC
gen_dc	generator: DC
gen_nd_1	generator 1: no current designation

Genie Abbreviation	Description
gen_nd_2	generator 2: no current designation
mot_ac	motor: AC
mot_dc	motor: DC
mot_nd_1	motor 1: no current designation
mot_nd_2	motor 2: no current designation
mot_syn	motor, synchronous
trans_nd_1	transformer 1: no connection designation
trans_nd_2	transformer 2: no connection designation
trans_sd_1	transformer 1: star-delta (wye-delta)
trans_sd_2	transformer 2: star delta (wye-delta)
trans_ss_1	transformer 1: star-star (wye-wye)
trans_ss_2	transformer 2: star-star (wye-wye)

PLS_IEC_BASE_1 / PLS_IEC_BASE_2

These libraries include a variety of symbols for IEC equipment:

1 = small size

2 = large size

Genie Abbreviation	Description
sl_cap_bank_tuned_3	capacitor bank 3: tuned
sl_cap_bank_tuned_4	capacitor bank 4: tuned
sl_capacitor	capacitor
sl_capacitor_vari	capacitor, variable
sl_closed	closed switch
sl_contact_nc	contact break
sl_ct	contact
sl_fuse_1	fuse, option 1
sl_fuse_2	fuse, option 2
SL_Head	head
sl_head_2	head
sl_in_cb_rack	incoming, circuit breaker when racked out, plug
SL_In_Rack	incoming, circuit breaker when racked out, socket
sl_in_sw_hd_isol	incoming, switch head, isolated
sl_in_sw_head	incoming, switch head
sl_inductor	inductor
sl_inductor_adjust	inductor, adjustable
SL_Open	open
sl_out_cb_rack	feeder, circuit breaker when racked out, plug
SL_Out_Rack	feeder, circuit breaker when racked out, socket
sl_resistor	resistor
sl_resistor_adjust	resistor with adjustable contact
sl_resistor_vari	resistor, variable

Genie Abbreviation	Description
sl_sw_static_1	static switch 1
sl_sw_static_2	static switch 2

PLS_IEC_CB_1 / PLS_IEC_CB_2

These libraries include high-voltage drawout circuit breakers:
1 = small size
2 = large size

Genie Abbreviation	Description
hv_cb_bus_dr_c	high-voltage drawout circuit breaker, remote control
hv_cb_bus_dr_nc	high-voltage drawout circuit breaker, no remote control
hv_cb_bus_nd_c	high-voltage non-drawout circuit breaker, remote control
hv_cb_bus_nd_nc	high-voltage non-drawout circuit breaker, no remote control
hv_cb_fd_dr_c	high-voltage, drawout circuit breaker, remote control, earth at bottom
hv_cb_fd_dr_nc	high-voltage, drawout circuit breaker, no remote control, earth at bottom
hv_cb_fd_nd_c	high-voltage, non-drawout circuit breaker, remote control, earth at bottom
hv_cb_fd_nd_nc	high-voltage, non-drawout circuit breaker, no remote control, earth at bottom
hv_cb_inc_dr_c	high-voltage, drawout circuit breaker, remote control, earth at top
hv_cb_inc_dr_nc	high-voltage, drawout circuit breaker, no remote control, earth at top
hv_cb_inc_nd_c	high-voltage, non-drawout circuit breaker, remote control, earth at top
hv_cb_inc_nd_nc	high-voltage, non-drawout circuit breaker, no remote control, earth at top
hv_cb_nes_dr_c	high-voltage drawout circuit breaker, remote control, no earth
hv_cb_nes_dr_nc	high-voltage drawout circuit breaker, no remote control, no earth
hv_cb_nes_nd_c	high-voltage non-drawout circuit breaker, remote control, no earth
hv_cb_nes_nd_nc	high-voltage non-drawout circuit breaker, no remote control, no earth

PLS_IEC_SW_1 / PLS_IEC_SW_2

These libraries include IEC-style switches:
1 = small size
2 = large size

Genie Abbreviation	Description
sw_general	general switch
sw_isolated	isolated switch

PLS_METER

This library includes meter symbols.

Genie Abbreviation	Description
meter_general	general meter

ITEM1

This library includes miscellaneous symbols.

Genie Abbreviation	Description
Item1	value type and units block for a circuit breaker
Item2	value type and units block for a circuit breaker

Genie Abbreviation	Description
tab1	menu tab
Tab2	menu tab

Appendix 2: Using Citect INI Parameters

There are a number of Citect INI parameters that you may use to configure driver parameters. These settings may be configured at the protocol level, cluster level, port level, or device level. More specific settings will override a general one. The order of precedence is:

Protocol Name > Cluster Name > Port Name > I/O Device Name

The level at which you want the INI settings to be in effect determines the name you define. For example:

To set the default timeout for all devices using the Micrologic protocol, use:

[MICROLOGIC]

Timeout = 2000

To override this default for cluster ‘Cluster_1,’ use:

[MICROLOGIC.Cluster_1]

Timeout = 1000

To override the default value for port ‘Port_1’ on cluster ‘Cluster_1,’ use:

[MICROLOGIC.Cluster_1.Port_1]

Timeout = 3000

To override the default value for I/O device ‘CircuitBreaker_1’ on port ‘Port_1’ on cluster ‘Cluster_1,’ use:

[MICROLOGIC.Cluster_1.Port_1.CircuitBreaker_1]

Timeout = 4000

Most settings can be configured to be specific to a particular I/O device. Exceptions are noted in the description for the individual parameter.

Parameters Database

All INI parameters described in the sections below can be set in the Parameters database. Using special syntax, you can access the parameters in the Project Editor (System < Parameters):

- The section name generally corresponds to the INI section name, although it includes the protocol name, cluster name, and primary device name only.
- The name is the INI value name.

If the parameter is set in the Parameters database, it becomes a new default for either protocol, cluster, or a concrete device (depending on the section name hierarchy).

Examples:

Section Name: [MICROLOGIC.Cluster_1.Breaker_1]

Name: Timeout

Value: 2000

This defines a new default timeout value for a redundant pair of MicroLogic devices (primary device is named Breaker_1 in Cluster_1).

Section Name: [PWRMODBUS.Cluster1]

Name: UseWriteMultiRegistersOnly

Value: 0

This sets UseWriteMultiRegistersOnly to 0 for all PWRMODBUS devices in Cluster 1.

The INI file is read after the parameter database is processed; thus the override options are set in the Parameters database.

In this section, you will find parameters organized into these categories:

General PowerLogic Driver Parameters

Performance Tuning Parameters

Waveform Parameters

Sepam Event Reading Parameters

MicroLogic Modules Configuration Parameters

Data Replication Parameters

General PowerLogic Driver Parameters

The following parameters are common to all PowerLogic SCADA devices.

watchtime

Controls how often PowerLogic SCADA will interrogate the driver to determine whether it is still online. This parameter can only be configured for an entire driver, and hence will have the driver dll name as its section name. Where another setting may be [PM870], to set this setting it must be [PLOGIC], as PLOGIC is the name of the dll. This is the only parameter whose section name is defined in this fashion.

Parameter type: seconds

Default Value: 2

Example: [SEPAM] watchtime = 5

kernelStatisticUpdateRate

Controls how frequently the statistics displayed in the driver kernel window are updated. This time period can be increased in order to decrease CPU load. This parameter can only be configured for the entire protocol (as with the watchtime parameter); it will have the driver dll name as its section name.

Parameter type: milliseconds

Default value: 5000

Examples:

[SEPAM40]

kernelStatisticUpdateRate = 20000

[SEPAM80]

kernelStatisticUpdateRate = 10000

UseWriteMultiRegistersOnly

Controls PWRMODBUS driver behaviour when a single register is to be written. This parameter is set to 1 by default, enabling all writes to be made using "write multiple registers" MODBUS function. Setting this parameter to 0 allows driver to perform write using "write single register" function if (and only if) one MODBUS register is about to be written in current operation.

Parameter type: integer

Default value: 1

Examples:

[PWRMODBUS]

UseWriteMultiRegistersOnly = 1

[PWRMODBUS.MYCLUSTER.PORT_1.BCM1]

UseWriteMultiRegistersOnly = 0

timeout

Controls how long the driver waits for a response from a device before setting that device as offline. This value should be greater than the device/gateway timeout period. A timed out request will not be retried. The reason for this is that TCP is a guaranteed transport mechanism, and the lack of a response indicates device failure or communications failure to that device. A device connected via a gateway should use the gateway's retry mechanism.

Parameter type: milliseconds

Default value: 5000

Examples:

[SEPAM40]
Timeout = 2000

[SEPAM40.MYCLUSTER.PORT_1.SLOW_SEPAM]
Timeout = 15000

retry

Defines the number of retry attempts for specific MODBUS requests. Retries may occur either when the request is timed out or certain MODBUS exception reply messages are received. The exact behaviour is controlled by the RetryTimeout and RetryException parameters.

Parameter type: number of attempts

Default value: 3

Examples:

[SEPAM40]
retry = 1

[SEPAM40.MYCLUSTER.PORT_1.SEPAM_DEVICE]
retry = 5

RetryTimeout

When enabled (by default), the driver will re-try a timed-out MODBUS request.

Parameter type: long (boolean)

Default value: 1

Examples:

[SEPAM40]
RetryTimeout = 1

[SEPAM40.MYCLUSTER.PORT_2.SEPAM_DEVICE]
RetryTimeout = 0

RetryException

When enabled (disabled by default), the driver will re-try a MODBUS request that has received MODBUS Exception messages. The number of retries is defined by the Retry parameter.

When Retry Exception is enabled, retry occurs when any of the following MODBUS exception messages is received:

SLAVE_DEVICE_FAILURE_EXCEPTION = 0x5

GATEWAY_PATH_UNAVAILABLE_EXCEPTION = 0xA

GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND_EXCEPTION = 0xB

SLAVE_DEVICE_BUSY_EXCEPTION = 0x6

MEMORY_PARITY_ERROR_EXCEPTION = 0x8

NEGATIVE_ACKNOWLEDGE_EXCPETION = 0x7

Parameter type: long (boolean)

Default value: 0

Examples:

[SEPAM40]

RetryTimeout = 1

[SEPAM40.MYCLUSTER.PORT_2.SEPAM_DEVICE]

RetryTimeout = 0

standbyRefreshRate

Controls how often a standby IO server attempts to poll a device to update its cache. This time period determines the maximum age that values may be when switching from a primary IO server to a standby. Decreasing this value degrades communications to the device.

Parameter type: seconds

Default value: 60

Examples:

[SEPAM40]

standbyRefreshRate = 30

[SEPAM40.MYCLUSTER.PORT_1.SLOW_SEPAM]

standbyRefreshRate = 120

standbyCheckTime

Controls how often the driver will enquire of PowerLogic SCADA as to whether it is in standby or primary mode. This value can be increased to reduce CPU load.

Parameter type: milliseconds

Default value: 500

Examples:

[SEPAM40]

standbyCheckTime = 500

[SEPAM40.MYCLUSTER.PORT_1.SLOW_SEPAM]
standbyCheckTime = 1000

statusUnitCheckTime

This parameter defines how frequently the driver will try to re-establish the connection with a device that has gone offline on a port that is not disconnected. It sets the maximum rate at which the driver enquires of the device, to determine if it is still operational. If the "watchtime" parameter is set to a longer time, that value will be used instead.

NOTE: If a network gateway has multiple devices connected to it, and one device is disconnected, the driver takes it offline and does not try to reconnect it according to this parameter's schedule. If the port is taken offline and then is reconnected, the driver will reconnect the devices immediately.

Parameter type: seconds

Default value: 5 (20 for MicroLogic)

Example:

[SEPAM40]
statusUnitCheckTime = 5

[SEPAM40.MYCLUSTER.PORT_1.SLOW_SEPAM]
standbyCheckTime = 10

initUnitCheckTime

Controls how long the driver waits before attempting to bring a device online after it has gone offline. This value can be decreased to bring offline devices back into service in a shorter period of time. In a multi-drop scenario, this time should be relatively long, to prevent init unit requests from stalling communications to the rest of the devices on that port.

Parameter type: seconds

Default value: 120

Examples:

[SEPAM40]
initUnitCheckTime = 5

[SEPAM40.MYCLUSTER.PORT_1]
initUnitCheckTime = 120

initCacheTimeout

Controls how long the driver will spend attempting to populate the cache before bringing a device online. When a tag has been incorrectly configured, the device will come online after this period of time.

Parameter type: seconds

Default value: 60

Examples:

[SEPAM40]

initCacheTimeout = 60

[SEPAM40.MYCLUSTER.PORT_1.SLOW_SEPAM]

initCacheTimeout = 30

cacheRefreshTime

Controls the maximum rate at which the driver will attempt to repopulate its cache. If the driver cannot refresh its cache within the time period specified, it will collect data as fast as the network allows.

Parameter type: milliseconds

Default value: 500

Examples:

[SEPAM40]

cacheRefreshTime = 1000

[SEPAM40.MYCLUSTER.PORT_1.FAST_SEPAM]

cacheRefreshTime = 200

[SEPAM40.MYCLUSTER.PORT_1.UNIMPORTANT_DEVICE]

cacheRefreshTime = 5000

StatusRegister

Defines a holding register that the driver reads to determine whether a device is responding to communication requests. The result of this read is not important, however it must be a valid register address within the device.

Parameter type: register address

Default value: 1100 or 2 (Sepam)

Examples:

[PWRMODBUS]

statusRegister = 1000

[PWRMODBUS.MYCLUSTER.PORT_1.DEVICE_A]

statusRegister = 16000

StatusRegisterType

Used together with StatusRegister; defines the type of the status register. Can only be configured for the PWRMODBUS driver. This parameter can have one of the following values:

0 - HOLDING register (default)

1 - INPUT register

2 - COIL register

3 - DIGITAL input (input coil) register

Any other value equals the default.

Parameter type: register type

Default value: 0

Example:

[PWRMODBUS]

statusRegister = 1000

[PWRMODBUS.MYCLUSTER.PORT_1.DEVICE_A]

statusRegister = 16000

statusRegisterType = 2

ModbusBase

Defines the base address for a device. Some MODBUS device registers are defined using a base address of 1. in this case, reading register 100 would actually require reading register 99. In other devices (such as the Sepam) the base address is 0. This parameter allows the base address to be configured according to the device.

Parameter type: integer

Default value: 0 for Sepam; 1 for all other drivers

Examples:

[PWRMODBUS]

ModbusBase = 1

[PWRMODBUS.MYCLUSTER.PORT_1.DEVICE_A]

ModbusBase = 0

RegMode

Specifies the order of bytes in a device register. It can only bet set for PWRMODBUS driver, and is supposed to be unit-specific. Value values are:

	RegMode	Order of bytes
Big endian (default)	0	1 0
Little endian	1	0 1

Any other value reverts to big endian.

Parameter type: integer

Default value: 0

Examples:

[PWRMODBUS]

RrMode = 0 # Default

[PWRMODBUS.MYCLUSTER.PORT_1.DEVICE_A]

RegMode = 1 # This device has little endian registers

timeZone

Time zone names are taken directly from the Windows registry database (case-insensitive), and will otherwise default to using the I/O server's local time zone. The Windows time zone database is located in the Windows registry in:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones\. The examples of time zone names are:

AUS Central Standard Time

China Standard Time

Pacific Standard Time (Mexico)

Use the general section [POWERLOGICCORE] to specify the time zone for all devices, e.g.

[POWERLOGICCORE]

Timezone = Mountain Standard Time

This sets the default time zone for all devices (Sepam, PLogic, Micrologic, PWRMODBUS). Otherwise the time zone can be specified for each device with precedence taken as described in the start of this section.

Examples:

[PLOGIC870.Cluster1.Singapore_Port]

Timezone = Singapore Standard Time

[PLOGIC870.Aus_Cluster]

Timezone = Aus Central Standard Time

Not having a time zone specification means that the device is in the same time zone as the machine where the I/O Server is running. No time conversion will be done.

Performance Tuning Parameters

Several parameters are provided to allow tuning of the performance. These parameters fall into three broad categories; bandwidth allocation, packet blocking optimisation, and tag scan rates.

Bandwidth Allocation Parameters

Bandwidth can be allocated for the different types of data as desired. The parameters to perform this are as follows:

[Parameter]	[Default Value]	[Parameter Type]
EventBandwidth	25	integer
WaveformsBandwidth	12	integer
CommandsBandwidth	13	integer
RealTimeBandwidth	50	integer

The percentage bandwidth allocated to each queue will be the ratio of an individual queue's value when compared to the total sum of defined bandwidths. The default values have a sum of 100 for ease of reference. Any unused bandwidth will be shared amongst the other categories.

Bandwidth can be configured at the port level, but not the device level.

Example:

[SEPAM40]

EventsBandwidth 30

WaveformsBandwidth 5

CommandsBandwidth 15

RealTimeBandwidth 50

[SEPAM40.MYCLUSTER.PORT_1]

EventsBandwidth 50

WaveformsBandwidth 30

CommandsBandwidth 10

RealTimeBandwidth 10

BandwidthAllocation

This parameter allows the ratio of bandwidth assigned to each device sharing a port to be configured. This parameter can only be configured at the device level.

Parameter type: integer

Default value: <equal split>

Example:

[SEPAM40.MYCLUSTER.PORT_1.DEVICE_A]

BandwidthAllocation 70

[SEPAM40.MYCLUSTER.PORT_1.DEVICE_B]

BandwidthAllocation 30

Packet Blocking Optimisation Parameters

For all devices except the Sepam, parameters can be configured to optimise the MODBUS packets that are created for collection of data from the device. Sepam devices have pre-configured blocks that are already optimised.

The parameters that control the blocking are as follows:

enableScatteredReads

This causes the driver to use the 'scattered read' extension that can help improve blocking. This option should be enabled for devices that support this extension.

Parameter type: Boolean flag

Default value: 0 for generic Power MODBUS driver, 1 for PowerLogic driver

Example:

[PWRMODBUS.MYCLUSTER.PORT_1.DEVICE_A]

enableScatteredReads 1

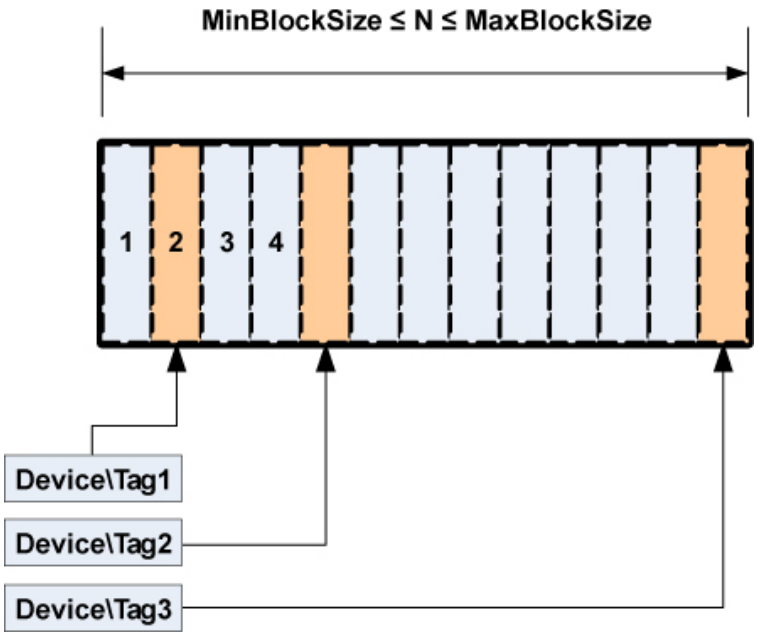
[PWRMODBUS.MYCLUSTER.PORT_1.DEVICE_B]

enableScatteredReads 0

percentBlockFill

This parameter defines the maximum percentage of configured registers contained in a block before the drivers creates fixed blocks instead of scattered blocks. The following figure illustrates now a block of N registers can be constructed:

- If M<N registers are configured, the block builder can either create a scattered block or a multi-register block.
- If M/N*100% is less than PercentBlockFill, the block builder creates a scattered registers block.
- If the percentage of configured registers ≥ PercentBlockFill, the block builder creates a multi-register block.



Parameter type: percentage

Default value: 50

Example:

[PM870.MYCLUSTER.PORT_1.PM_DEVICE]

percentBlockFill 50

[CM4000.MYCLUSTER.PORT_1.CM_DEVICE]

percentBlockFill 80

maxBlockSize

This parameter defines the maximum number of registers that can be read in a single request. By default, this is 124, but some devices can read more than this.

Parameter type: integer

Default value: 124

Example:

[PWRMODBUS.MYCLUSTER.PORT_1.DEVICE_A]

maxBlockSize 1024

minBlockSize

This parameter defines the minimum number of registers to read as a fixed block before the block builder will instead add those registers to a scattered block. If latency is low, and scattered reads are expensive, this value should be lower. If latency is high, or scattered reads are inexpensive, it is better to set this value higher. Only applicable when scattered reads are enabled.

Parameter type: integer

Default value: 20

Example:

[PM870.MYCLUSTER.PORT_1.LOW_LATENCY_DEVICE]

minBlockSize 10

[CM4000.MYCLUSTER.PORT_1.HIGH_LATENCY_DEVICE]

minBlockSize 100

Tag Scan Rate Parameters

Each tag can be configured at a priority level from 1-3 where 1 is the highest. Parameters exist to adjust the relative scan rates of the high and low priority tags in comparison to the nominal tag scan rate.

HighScanRate

Parameter type: percent relative to nominal

Default value: 50

LowScanRate

Parameter type: percent relative to nominal

Default value: 200

Using the default parameters, the high priority tags will be refreshed twice as fast as the normal priority tags, and the low priority tags will be refreshed at half the rate of the normal priority tags. These parameters can be configured at the port level and higher.

Using the default settings and a nominal tag refresh rate of 1 second:

Low Priority Tag Refresh: 2000 ms

Normal Priority Tag Refresh: 1000 ms

High Priority Tag Refresh: 500 ms

Example:

[PM870.MYCLUSTER.PORT_1]]

HighScanRate 25

LowScanRate 500T

Advanced Tag Block Capabilities (Invalid Memory Access Blocks defined)

Some devices may restrict access to certain memory registers. Such registers may be available for read only, write only or may not be available at all, resulting in a MODBUS exception when the registers are addressed.

Definition: Blocks of registers that cannot be read or written to are referred as “invalid memory access blocks.”

These devices create a challenge for the PWRMODBUS driver. If the device has invalid blocks that do not support scattered reads (or they are disabled for this device), the driver may try to read registers in blocks that intersect with the registers that cannot be read. This can result in the whole block being invalidated and, in certain cases, may also result in device being taken offline. Figure 1 (below) illustrates an invalid block in the middle of an address space.

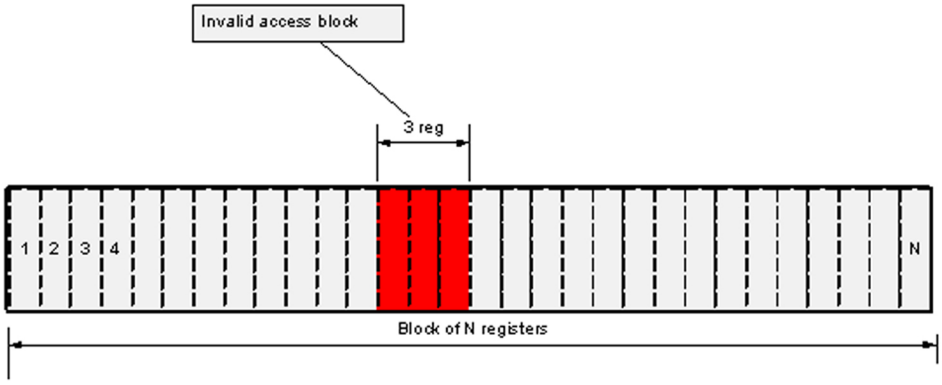
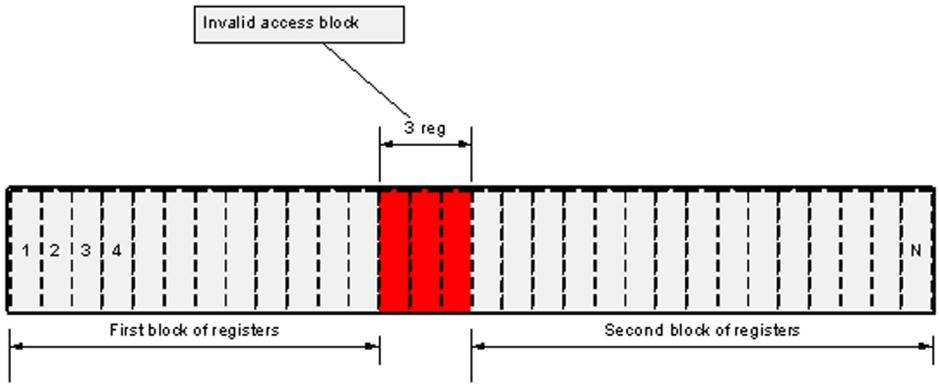


Figure 1 shows how the "invalid memory access block" affects MODBUS register blocking. In this situation, if the driver does not know that the block of 3 inaccessible registers exists, it will try to block all registers from 1 to N (depending on data that was requested by the real-time data collector). This block, however, will never be read successfully, as the device will respond with an exception to all attempts to read invalid registers.

If the configuration includes information about invalid memory access blocks, the driver will create two blocks instead of one, as shown in Figure 2:



In Figure 2, invalid registers were taken into account when the block was constructed. When configuring device that has invalid memory areas, it is especially important to define all blocks that may interfere with any of the tags.

Invalid Block Tag Definition Syntax

Invalid access memory areas are defined as variable tags, using the following format for the address:

```
T:IB;{m|i|c|s}:<start_register>:u<count>;E:1;L:P:0
```

where

- `m, i, c` and `s` define the type of MODBUS register
- `<start_register>` is the first register address of the invalid access block
- `<count>` defines the number of registers in the invalid access block

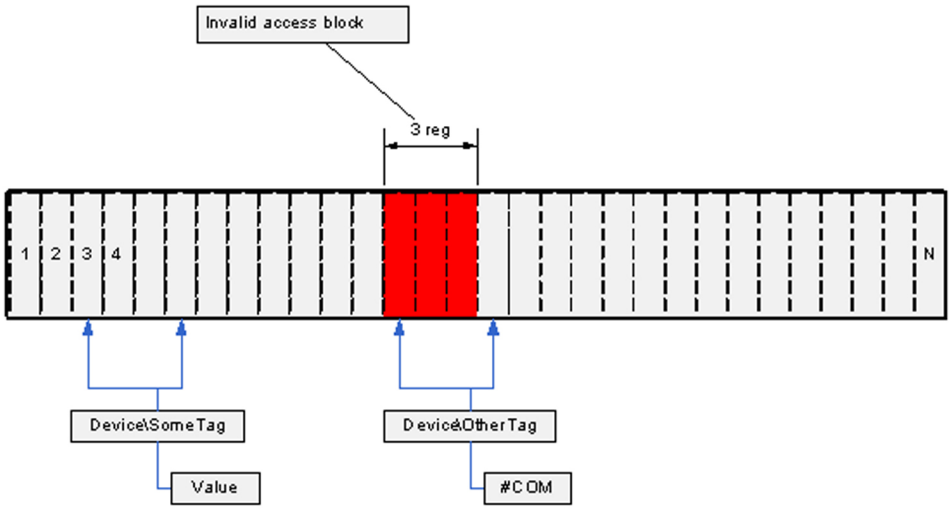
Example:

```
T:IB;m:300:u10;E:1;L:P:0
```

This defines an invalid access block of ten holding registers starting from register 300.

Configuration Notes

When one or more invalid access blocks is defined according to the syntax above, tags configured to read any of invalid registers will be affected by it. If any of the tag registers fall into an invalid memory access block, this tag will not be readable; any attempt to read its value will result in #COM, as shown on Figure 3:



However, such tags do not affect other tags, because the PWRMODBUS driver implements algorithms that prevent tags from being invalidated by invalid memory block logic.

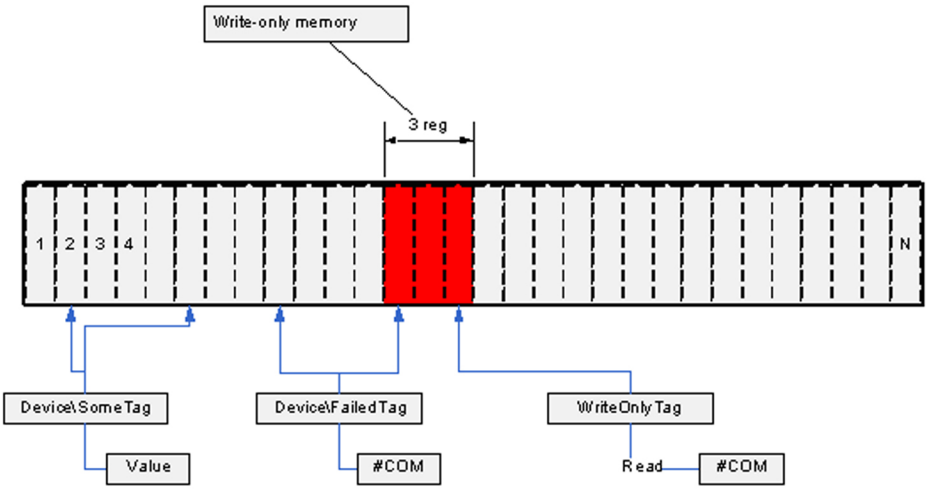
Tags that try to use invalid registers are detected on startup and can be found by analysing the log file. This is an example trace:

```
[DEBUG]
[REAL][GeneralDriver::BaseDatapointBuilder::BuildDataPoints()]
Adding datapoint. Tag - BCM1\H_QIVR34\SwlStr Address -
T:SS;m:283:2;E:1;L:P:26 Datapoint: class Datapoints::Status_SS

[DEBUG] [REAL][RealTimeData::DeviceCache::Subscribe()] Init
Registers: Polled Registers: Address:283 Type:3

[ERROR] [MISC] [RealTimeData::BlockBuilder::AddDataPoint()] Cannot
add datapoint, one or more invalid memory addresses fall into non-
splittable block

[ERROR] [MISC]
[GeneralDriver::BaseDatapointBuilder::BuildDataPoints()] Could not
init datapoint. Tag BCM1\H_QIVR34\SwlStr Address
T:SS;m:283:2;E:1;L:P:26. Analyze other messages, this tag address
may contain invalid registers
```

Such output is expected when a holding register with address 283 is declared invalid. This trace helps figure out any configuration issues.

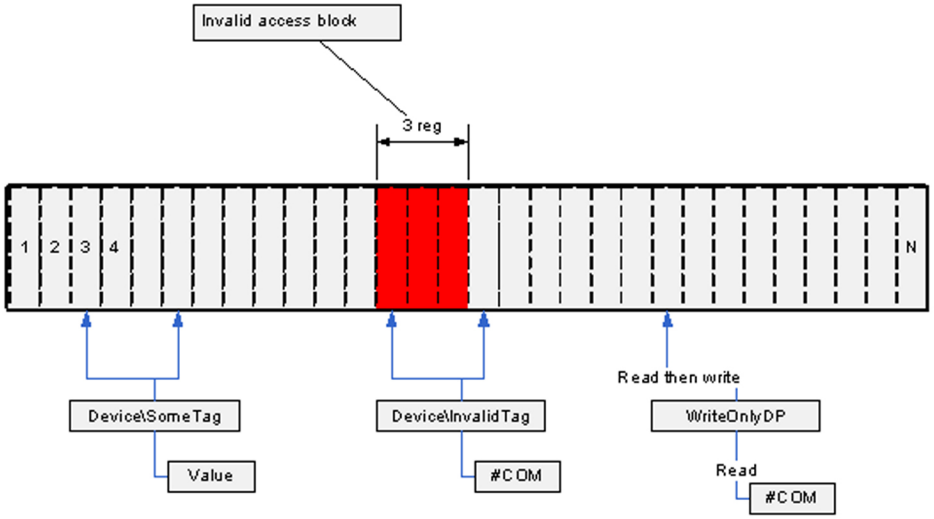
Write-only Memory

If a certain register range is accessible for write only, no additional configuration may be needed. However, to prevent the driver block optimizer from including these registers in a poll, they also must be configured by adding “invalid block” tags as described in the *Invalid Block Tag Definition Syntax*, described above. Declaring these registers invalid will not prevent drivers from trying to write to them. Figure 4 illustrates a write-only memory configuration:



Write-only registers should not be confused with write-only datapoints that internally read a register before attempting to write. Declaring the register they read invalid will result in a datapoint not working; such mistake should be avoided. Figure 4 shows “WriteOnlyDP” as an example; this tag cannot be read (it will result in #COM), but internally it needs to get the register value before writing into it. If this register was declared invalid, tag writes would also fail.

Figure 5 illustrates a write-only datapoint:



Tag Blocking Notes

PowerLogic SCADA drivers support an advanced blocking mechanism for tags. That is, real-time tags are no longer blocked together with write-only tags.

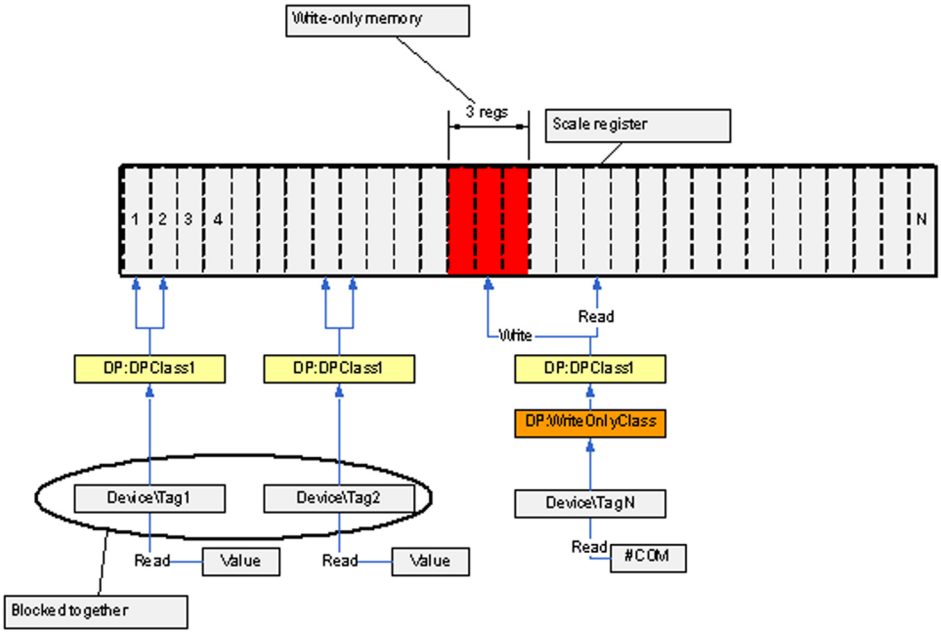
Tags found invalid, due to intersection with invalid memory areas, are not blocked with “good” real-time tags and will not therefore interfere with them.

Write-only Tags

Beginning with driver version 2.0.1.1, the write-only tags feature is fully supported.

There are no special logic codes or address formats for write-only tags. If a tag references memory that was declared invalid (see *Invalid block tag definition syntax*, above), and its datapoint has writing capabilities, the tag becomes write only. No preliminary checks are performed to verify that the memory can be written to, and no additional configuration is needed. It is assumed that, if the tag is configured to write into memory that has been declared “invalid,” the memory can actually be written to.

It is important to understand that scaled write tags (code 110) will become write-only tags, if that scale register can be read. Tag Device\TagN on Figure 6 explains this case: the datapoint needs to read the scale value from the scale register in order to write scaled value to write-only register. However, as long as the actual register belongs to the memory that can only be written to (and it is configured using T:IB tag syntax as explained in *Invalid Block Tag Definition Syntax* above), this tag cannot be read.



The fact that the tag mentioned before cannot be read will not affect other tags reads (see *Tag Blocking Notes*, above).

Note: The write-through feature of the device cache is disabled for write-only tags.

Waveform Parameters

The following parameters configure the waveform downloading behavior. These parameters are only applicable for Sepam devices and PowerLogic devices that support waveforms.

[Parameter] [Default Value] [Parameter Type]

WaveformsDisable 0 Boolean value

WaveformMatchMargin 10 seconds

WaveformCheckTime 30 seconds (PM/CM)

WaveformZone 1 integer (Sepam)

WaveformsDisable

This parameter enables or disables waveform downloading for a particular device.

Parameter type: Boolean value

Default value: 0

Example:

[SEPAM40.MYCLUSTER.PORT_1.DEVICE_A]

WaveformsDisable 1 //Disable waveform downloading

WaveformMatchMargin

Alarms are matched to waveforms by the timestamp of each. This parameter is the maximum difference between alarm timestamp and waveform timestamp for PowerLogic SCADA to consider it a match.

Parameter type: seconds

Default value: 10

Example:

[SEPAM40]

WaveformMatchMargin 2

WaveformCheckTime (PM, CM, and Sepam)

This parameter defines the time the driver will wait between checking for new waveforms.

Parameter type: seconds

Default value: 30

Example:

[SEPAM40.MYCLUSTER.PORT_1.DEVICE_A]

WaveformCheckTime 60 (checks every 60 seconds)

WaveformZone (Sepam)

This parameter defines the Sepam waveform zone that the Sepam driver will use to collect waveforms from the device. This allows two masters to extract waveforms from the same device. Valid values are 1 or 2.

Parameter type: integer

Default value: 1

Example:

[SEPAM40.MYCLUSTER.PORT_1.DEVICE_A]

WaveformZone 2

Sepam Event Reading Parameters

These two Sepam parameters are described below.

EventTable

This parameter defines the Sepam event table that the Sepam driver uses to collect alarms from a device. This allows two masters to extract alarms from the same device. Valid values are 1 or 2.

Parameter type: integer

Default value: 1

Example:

[SEPAM40.MYCLUSTER.PORTO_1.DEVICE_A]

EventTable 2

EventIdle

This parameter defines the time that the driver will wait before requesting the next event from a Sepam device. It may be possible to reduce this value to increase the rate at which alarms can be retrieved from the device.

Parameter type: milliseconds

Default value: 500

Example:

[SEPAM40.MYCLUSTER.PORTO_1.DEVICE_A]

EventIdle 200

MicroLogic Modules Configuration Parameters

A MicroLogic unit consists of three or four modules, each acting as a separate MODBUS device; however the I/O server views MicroLogic as one I/O device. The communication control module (CCM) is optional for MicroLogic; its presence may be detected by the driver or specified in the INI file.

CCM

The CCM parameter specifies whether a CCM is present on the device or if the driver should try to detect its presence ("auto mode"). Valid values are:

CCM not present - 0

CCM present - 1

Auto mode - 2 (default)

Any other value reverts to auto mode.

Parameter type: integer

Default value: 2

Example:

[Micrologic.MYCLUSTER.PORT_1.DEVICE_A]

CCM=1

Module-Specific Packet Blocking Optimisation Settings

Due to different firmware versions, MicroLogic modules may require different blocking settings. This is especially true when MicroLogic contains a BCM that supports MODBUS "read multiple registers" requests for up to 124 registers, and an MM or a PM module that supports 21 register reads at max. The MicroLogic driver allows blocking optimisation parameters to be overridden for each of the device's modules, as in the following example:

[Micrologic.MYCLUSTER.PORT_1.DEVICE_A]

maxBlockSize = 124

[Micrologic.MYCLUSTER.PORT_1.DEVICE_A.BCM]

maxBlockSize = 21

The parameter set for the device applies to all of its modules unless overridden in a module-specific section (e.g., [Micrologic.MYCLUSTER.PORT_1.DEVICE_A.BCM])

These parameters can be overridden:

- enableScatteredReads
- minBlockSize
- maxBlockSize
- PercentBlockFill

This applies to the BCM, CCM, MM, and PM modules.

Data Replication Parameters

These parameters are used to configure the root data directory path of the local server and a partnered standby server. If FileReplicationSrc is not defined, a UNC path to PowerLogic SCADA default data directory is used. These settings are server wide, and must be added to the 'PowerLogicCore' area of the ini file.

FileReplicationDst

Parameter type: string

Default value: \\servername\share\path

FileReplicationSrc

Parameter type: string

Default value: \\servername\share\path

Example:

[PowerLogicCore]

FileReplicationDst = \\standby\Power\Data

FileReplicationSrc = \\primary\Power\Data

Graphics Library Parameters

Maximum number of entries that can be held in Event Log

The Alarm Summary length parameter in Citect.ini defines the maximum number of entries that can be held in the Event Log (default = 5000 entries). You can view all events in the Event Log and alarms in the alarm logs (Alarm Log, Unacknowledged Alarms, Disabled Alarms).

Each event requires 256 bytes of memory, plus the length of the comment. 32,000 entries will require at least 8 MB of memory. If you have many events, you should ensure that there is enough memory to store them in RAM.

After the parameter number is reached, older events are FIFO'd out to storage in [Installed Project Directory]\Schneider Electric\PowerLogic SCADA 7.10\Logs

Parameters for Alarm and Event States

[Alarm]

UseConfigLimits = 1

CacheLength = 2500

!Sound1 = <wave file name>

!Sound1Interval = <repeating interval in milliseconds>

!Sound2 = <wave file name>

!Sound2Interval = <repeating interval in milliseconds>

!Sound3 = <wave file name>

!Sound3Interval = <repeating interval in milliseconds>

[AlarmFormat]

EventLog=OnDate | Date, OnTimeMS | Time, Custom1 | Equipment, Name | Description,
SumState | State | Custom2 | Location, UserName | User

[AlarmStateText]

ON=<default text for ACTIVE state>

OFF=<default text for INACTIVE state>

ACK=<default text for ACKNOWLEDGED state>

ENA=<default text for ENABLE state>

DIS=<default text for DISABLE state>

CLE=<default text for CLEAR state>

These parameters are read only when the system starts up. The user must restart PowerLogic SCADA if they change these parameters.

If you do not specify any value for these parameters, these default values will be used, in this order:

Appearance

Disappearance

Acknowledge

Enable

Disable

Clear

[PLSEventLog]

```
FileFormat=0
EntryCount=250
ScanRate=10

[Path]
[PLSEventLog]=c:\SharedData\Alarm
```

Appendix 3: Glossary

alarm categorization

Added when setting up custom tags, this is one of the alarm filters. which will be used for filtering and sorting alarms in the Alarm Log. Categories are: normal, over, over hs, rate of change, reversal, sag, swell, transient, under, and under hs.

alarm text (On/Off)

For onboard alarms, this is the text (added while adding a custom tag) that displays when the alarm is on or off. This text will display in the Alarm Log.

alarm filters

Setup in the Profile Editor, these filters help you filter and sort data that displays in the Alarm Log.

alarm groups

Added when setting up custom tags, this is one of the alarm filters. which will be used for filtering and sorting alarms. Groups are: frequencies, motors, power factors, powers, temperatures, time, and voltages.

alarm levels

Added when setting up custom tags, this is one of the alarm filters. which will be used for filtering and sorting alarms. Levels are: event, high, medium, and low.

alarm types

Added when setting up custom tags, this is one of the alarm filters. which will be used for filtering and sorting alarms. Types are: diagnostic, power quality, protection, and system.

Automation Interface

Used instead of the Profile Wizard, this tool allows you to add multiple devices at a time to a PowerLogic SCADA project.

bandwidth

The amount of space or processor resource being used by a part of the system. You can use the bandwidth allocation parameters to allocate bandwidth for different types of data.

baud rate

The speed of transmission of electrical signals on a line. This is often described in bits per second (bps), although the baud rate and bps are not truly interchangeable. The baud is actually the measurement of how frequently the sound changes on the line.

bitmask

A mask is defined as data that is used with an operation to extract information that is stored in another location of the code. A bitmask is the most common mask used. It

extracts the status of certain bits in a binary string or number (a bit field or bit array).

Cicode

This programming language, which is similar to Visual Basic or "C," allows you to access and edit real-time data in the PowerLogic SCADA project. Although not difficult to use, the person working in Cicode must have received Cicode training from PowerLogic SCADA.

cluster

A discrete group of alarms servers, trends servers, reports servers, and I/O servers. It would usually also possess local control clients. For a plant comprising several individual sections or systems, multiple clusters can be used, one cluster for each section.

Color Change Tool

(also Project Color Change Tool) This tool allows you to assign colours for busbars and circuit breakers in genies, for use in graphics pages. You can assign separate colours for up to seven different states (for example, error, intermediate). When the busbar or circuit breaker is in the specified state, the system will change its colour to match what is set up here.

CommsMethod (communications method)

This is the communication protocol, such as MODBUS/RTU via Gateway, that is being used by a device. When adding devices in the automation interface, you will need to specify the CommsMethod.

ComPort

(also COM port) The computer's communications port used to connect to devices, for sending and receiving serial data.

control

This is a command written to a device register that then causes an action within some equipment. There are a series of default control tags in PowerLogic SCADA to achieve these actions. For example, in the Sepam 40, there are control tags to operate a circuit breaker and enable a recloser.

custom device type

This is a "new" device type that is added to a system. Although the Profile Editor includes many standard device types, it may be necessary to add a new device type that includes custom tags, or one that includes a different set of tags than the standard device types.

custom tag

This is a "new" tag that is added to the system. Although the Profile Editor includes many standard tags, you may need to add a tag for a third party device, or to edit an existing tag to change its attributes. In these cases, you need to add a custom tag.

DataBits

This is the number of data bits used in data transmission. The I/O device and the ComPort must have the same value.

demo mode

This demonstration mode allows you to run PowerLogic SCADA without a hardware key. You can use all of the PowerLogic SCADA features, but with limited runtime and I/O options. See *Using the Demo Mode* on page 124 for details.

design time environment

To be used only by the person who is creating and installing the project for the end user, this is the environment in which you add devices, profiles, and projects, as well as create genies and one-lines.

device category

Used in the Profile Wizard to logically group device profiles, to make them easier to locate. The default category is "Schneider Electric, and the default subcategories are "Monitoring Device," "PLC," and "Protective Device." Do not confuse these terms with:

- categorization and subcategorization (alarm filters, used during run-time, to filter and sort alarm data)
- category type: real-time filters that provide metadata for future reporting

device type drivers

These are programs that allow PowerLogic SCADA to interact with a device or series of devices. PowerLogic SCADA includes several generic drivers (generic MODBUS, Sepam 40 Range, MicroLogic 5P and 6P, CM4000 series, and PM800 series) that interact with "standard" device types.

device profile

A subset of the device type: where the device type includes all of a device type's attributes, the device profile includes only the specific tags that are used by an individual customer.

device type

A device type includes everything that we “know” about a device and its capabilities. Each device type is made up of real-time tags, alarm tags , trend/logging tags , resets, and controls. In short, it includes everything that is required to communicate and display predefined data for a customer.

engineering unit templates

Used for conversions between base units and their conversions (for example, inches to centimeters or amperes to kiloamps).

enumeration (used for the circuit breaker status)

This is a single value (0-5) that defines a condition that is determined by multiple bits. They allow for dynamic contingencies, such as when you need to use multiple bits to describe the position of a circuit breaker.

format code

These codes define the attributes of the address field of a tag. See *Format Code Definitions* on page 56 for a list of format codes.

genie

A genie is a multi-layer graphic that is used on the Graphics page to indicate an object, such as a motor, generator, circuit breaker, or switch. Using genies, you only have to configure common behaviours of that object once. The default genie library includes a large number of pre-defined genies.

graphics test mode

Only available in the pls_example project, use this mode for off-line testing of animation in one-line drawings. Do not use this mode with physical devices that are connected to a live environment.

IEC tag name

he IEC61850-compatible name that is created when a tag is created. When you add a tag to PowerLogic SCADA, you must follow the IEC61850 format.

logic code

Logic codes tell the PowerLogic SCADA program how to mathematically certain values in device registers, thus providing values that the user needs. Examples of logic codes are date and time for a circuit monitor or a Sepam device, digital inputs/outputs, and IEEE power factor.

metadata

Metadata provides data about other data. In PowerLogic SCADA, metadata might include additional information about a custom tag: its category type, utility type, statistical type, or quantity. It is often used for reporting purposes.

multi-monitor support

This option allows you to view the runtime environment from multiple computer monitors. In PowerLogic SCADA, this allows you to view a different startup page on each monitor.

onboard alarm

Onboard alarms are alarms that are detected and stored in a device's data logs. If an onboard alarm is configured within a device, you can map it, via the Profile Editor, to a digital time-stamped alarm in PowerLogic SCADA. These alarms and associated waveforms can be read and displayed in the PowerLogic SCADA Alarm Log.

parity

Parity is used as a simple means of detecting error by verifying that the result is odd or even. In PowerLogic SCADA, parity is required for the generic serial or MODBUS/RTU comms methods, when adding a device to PowerLogic SCADA.

PC-based alarms

PC-based alarms are alarms that are detected from a device and are stored in the software. You can add them to the Profile Editor when you create the device profile. All PC-based alarms are analog by default.

PMCU

The Meter Configuration Help Utility. Use this application to set up the features within PowerLogic devices, and enabling such features as onboard alarms and waveforms. The information that is generated from PMCU is then available for use within PowerLogic SCADA.

point (see SCADA tag)

polling priority

When adding a custom tag, this field determines the level of priority that PowerLogic SCADA uses when reading data from the related device. Options are low, normal, or high.

PowerLogic SCADA tag name library

This library includes electrical parameters, or measurements or topics. A tag name has three parts:

- an easy to read name (such as Current Phase A)
- a unique identifier
- meta data (attributes used to categorize the data for intelligent display/analysis)

Profile Editor

This tool allows you to create device type tags, device types, and device profiles. This information is then imported into PowerLogic SCADA, for use in creating graphics pages.

Profile Wizard

This tool allows you add device profiles to, or delete them from, a PowerLogic SCADA project. From the Profile Editor, you export profile data into a file that can be used in PowerLogic SCADA. From there, you use the Profile Wizard to add the device profile into a PowerLogic SCADA project.

Project Color Change Tool

This tool, accessed from Citect Explorer, allows you to set unique colours for seven different states in busbars and circuit breakers.

register scaling

This is a conversion that is the result of applying a scaling multiplier to a register value.

resets

This feature allows you to reset data from a device. There are some pre-defined resets, such as device date/time and onboard data logs, You can also add custom resets.

runtime environment

This is the final set of screens that are provided to the end customer. This environment includes the one-line(s), alarm and event pages, and the analysis page (from which users can view trends).

SCADA tag (SCADA point)

A SCADA tag is an extension of the Tag Name. A SCADA tag is made up of five parts: two in addition to those already defined in the PowerScada Tag library:

- an easy to read name (such as Current Phase A)
- a unique identifier
- an address (where on a device to read the raw data from)
- a formatting scheme (what to do with the data after it is read to scale it)
- meta data (attributes used to categorize the data for intelligent display/analysis).

StopBits

The number of bits that signals the end of a character in asynchronous transmission. The number is usually 1 or 2. Stop bits are required in asynchronous transmissions because the irregular time gaps between transmitted characters make it impossible for the server or I/O device to determine when the next character should arrive.

tag address

This "address" includes everything you need to know about a tag (quantity/topic). Included are the data type, priority, and logic code; and how the tag is displayed in registers. You can change address attributes on the Edit Address screen. The full tag address displays on the Define Device Type Tags tab when "Advanced Properties" is selected.

tag group

The basic groups include: real-time, of which there are several sub-groups (for example, currents, energies, frequencies and power factors); onboard; control; and reset.

user privileges (user access, user rights)

This feature allows you to control the amount of access that each user has to the system. User privileges are password-protected. See*Default User Access Settings* on page 126 for more information.

vector math

Vector math and vector math IEEE are two logic codes. They are the result of math that use vectors, which are directed quantities.

voltage levels

Voltage levels are used in the Color Change Tool. There are four levels (0–3). The measurement of each of these levels corresponds to a level used by the standard that applies for each customer.

Appendix 4: Frequently Asked Questions (FAQs)

The following items provide information about topics that generate frequent questions.

How do multiple engineers create device types and profiles at the same time?

It is highly recommend that one PC be nominated as the Master Profile Editor machine. Until the import/export has been incorporated into the Profile Editor, customisation must be completed at one PC. Then you can transfer the data to other machines. Not every change needs to be made on this machine – only changes that will impact your ability to share device types and profiles.

You must ensure that you have maintained synchronization between all the computers for:

- units and unit templates
- categories and sub-categories
- custom tag names

Recommendations on how to do this include the following:

1. Nominate one machine as the master.
2. Have a method in place to create the customisation on the master machine.
3. Nominate one person to make the changes, or allow multiple people to access the master machine and make their own changes.
    a. Ensure that everyone who needs access to the machine has access. It would be ideal to have something similar to Remote Desktop available to your team. Whatever method you use, ensure that only one person can be logged in at a time.
    b. Users must log in to the master machine, make their changes/custom entries, and then synchronize their local machine.
4. Have a daily backup in place for the master machine. Save several weeks of backups before overwriting them.

If I don't use PowerLogic drivers, how do I create device profiles?

The easiest way to do this is to:

1. Create a device type using a non-PowerLogic driver (like MODNET).
2. Using that device type, create a device profile.
3. You need to change the addressing of the new device type. The easiest way to do this is to copy the addressing from a known device type, and then make the necessary changes for the new device type.

Can I install ION E or SMS on the same computer as PowerLogic SCADA?

We do not recommend this practice. Each of those applications uses SQL server, which requires so much of the CPU that PowerLogic SCADA performance would be severely degraded.

When creating custom tags, how do we ensure consistency in their names and attributes?

Until the Profile Editor has been expanded to include an import/export utility, it will take work on part of the integration teams to keep themselves synched.

1. When you name a tag that you believe should be added as a standard tag, you should be able to identify it. We recommend “locking” these tags. Later we can pull the tags that were “locked” and work on creating “real” IEC tag names.
2. Create all tag names on the master computer; then copy the customtags.xml file (located in the OS-specific **data** directory: Data/Profile Editor/ Vx.x) to each computer that will be used to create and edit device-types/profiles.
3. When creating a new tag name, be sure to complete all of the alarm/logging filter (metadata) data fields. Without these fields properly completed, the real-time and alarm filters, and future reports/analysis will not work.

It is often easier to “copy from” a similar tag when creating a new tag. This will pre-populate all of the fields, including the metadata/filter data. Then you can change the tag as needed.

How should we manage categories and subcategories?

We recommend that each integration team decide in advance which categories and subcategories they will use. The automation interface requires the entire Profile name (which uses the category and subcategory as part of its name). Thus, you must be consistent in naming if the profiles are going to be shared and re-used.

1. Category should be used for a vendor.
2. Subcategory should be used to describe a type of device.
3. From the master computer that has the Profile Editor installed, create the categories and subcategories that you plan to use.
4. Copy the DeviceTypeCategories.xml file (located in the OS-specific **data** directory: Data/Profile Editor/ Vx.x) to every computer being used to create profiles.

How should we manage units and unit templates?

Until the Profile Editor has been expanded to include an import/export utility it will take work on part of the integration teams to keep themselves synced.

Create all units and unit templates on the master computer. Then copy the units.xml and unitstemplates.xml (located in the OS specific **data** directory: Data/Profile Editor/ Vx.x) to each computer that will be used to create/edit device-types/profiles.

When should I create a device type rather than device profile?

Create a new device type, instead of a profile, when the addressing for a specific tag needs to change. An example:

The integration team can choose which Input to which they will wire circuit breaker status and position. In this case, the tags for circuit breaker status and position would have different addressing, based on how that particular circuit breaker is wired. We recommend a new device type in this case.

How do we share or transfer device types and device profiles between computers?

Until the Profile editor has been expanded to include an import/export utility, it will take work on part of the integration teams to keep themselves synced.

You must ensure that you have maintained synchronization between all the computers for:

- units and unit templates
- categories and subcategories
- custom tag names

To transfer or share device types:

1. On the source PC: From the OS-specific Data/Profile Editor/ Vx.x directory, locate the device type of interest (<Device Type name>.xml). Copy it to the corresponding directory on the destination PC.
2. On the source PC: From the Data/Profile Editor/ Vx.x /DeviceTypes directory, locate the device type of interest (<Device Type name>.xml). Copy it to the corresponding directory on the destination PC. (You must locate the proper version of the Profile Editor. It will be associated with the version of the Profile Editor that was used to create it.)

   If you do not know the profile name, or if you want to pull over all of the profiles for a given device type):

   a. From the Profile Editor, Add/Edit Device Type screen, click “Edit Existing.”
   b. Choose a Device Type to Create From.
   c. Click “Display Associated Profiles.” Note the list of associated profiles to the chosen device type.
   d. Copy the profile file(s) (<Device Profile name>.xml) to the corresponding directory on the destination machine.

How can we re-use previous Profile Editor projects?

Until the Profile editor has been expanded to include an import/export utility it will take work on part of the integration teams to keep themselves synced. It is recommended that you

You must ensure that you have maintained synchronization between all the computers for:

- units and unit templates
- categories and subcategories
- custom tag names
- device types
- device profiles

To transfer/share projects:

1. On the source PC: From the OS-specific Data/Profile Editor/ Vx.x directory, locate the project of interest. Copy the entire project-specific directory to the corresponding directory on the destination PC.
2. On the source PC: From the Data/Profile Editor/ Vx.x /DeviceTypes directory, locate the project of interest. Copy it to the corresponding directory on the destination PC.

(You must locate the proper version of the Profile Editor. It will be associated with the version of the Profile Editor that was used to create it.)

How do we synchronize a new PC with the master Profile Editor PC?

To synchronize a new machine with the latest device types and profiles from your master Profile Editor PC:

On the source PC: From the OS-specific Data/Profile Editor/ Vx.x directory, copy the entire OS-specific Data/Profile Editor/ Vx.x directory to the corresponding directory on the destination machine.

What do I do before I add or remove devices in the Profile Wizard?

1. You must close all open dbf files in the Project Editor (or in Excel).
2. If you are removing a device, check “pack database after removal” on the last page of the wizard.
3. Any changes that you made inside the Citect Project Editor (such as setpoints or data type modifications) are lost when you delete the device from PowerLogic SCADA.

What are the requirements for device names?

Device Name:

Keep Device name ≤ 16 characters. Use _ as a separator.

If you use a naming convention that incorporates location, you will be able to do filtering on alarm location, etc.

- For example, Site_Building_Panel_device would be named Sx_Bx_Px_Device. (Site1_Building1_Panel1_CM41 — S1_B1_P1_CM41).
- The fewer levels you have, the more characters you can have in each level.

Device Comment:

Use this field for an alias of the device name.

This comment will be placed in the Equipment database, which is accessible from Cicode.

How do I use MODBUS communications methods?

MODBUS TCP/IP via Gateway: Use this for any device that is not speaking TCP/IP natively. These devices connect through a gateway device such as an EGX or ECC.

MODBUS TCPIP: Use this for any device that can speak TCP/IP natively. This includes CM4 or PM8 devices that have an ECC card installed.

How can I add more than one device at a time?

The Automation Interface requires that the profiles have already been exported from the Profile Editor to the PowerLogic SCADA project.

If the automation spreadsheet attempts to add a device that is already present in the project, an error will be thrown.

In the event that an error is thrown (for invalid profiles, communication parameters, etc), the row containing the error will display in Excel. To prevent duplicate device entries from being attempted, you must remove any rows above the row indicated in the error message.

If you need to keep a record of the devices added to the system, then keep each of the spreadsheets that was used to install devices in a known location for that customer.

The Setup Sheet needs to be modified for each project. Specify the entire path for each file.

The Input Sheet requires the following:

The entire path name for each profile. The path name for a profile is based on the category and subcategory from the Profile editor.

Example: Schneider Electric.Monitoring Device.Branch Circuit Monitor Full

In PowerLogic SCADA, what are the naming conventions for servers and clusters?

There is no enforced naming convention for server and cluster names, other than the restriction that each server name and cluster name must be unique. Cluster names must be a maximum of 16 characters, no spaces; they may not begin with a number.

Each team should come up with a naming convention for the servers and clusters. If the team is consistent, it will make it easier to edit/create the automation spreadsheet used for device addition.

How and when do I create users for the Runtime environment?

New projects do not have any users created by default.

The default PowerLogic SCADA graphics objects (such as circuit breakers and alarm pages) are constructed using a pre-defined set of user privileges the PowerLogic SCADA security grid). During development, you must have users of various privilege levels for testing purposes. Create users for each of the various levels according to the PowerLogic SCADA security grid. To make the best use of these privileges, we recommend that you use this security grid when adding users as you create new projects.

See *Default User Access Settings* on page 126. For additional information, see Using_Security.html in the CitectSCADA.chm help file (Start > Programs > Schneider Electric > PowerLogic SCADA 7.10 > PowerLogic SCADA Help).

How do I manage projects in the Citect Explorer environment of PowerLogic SCADA?

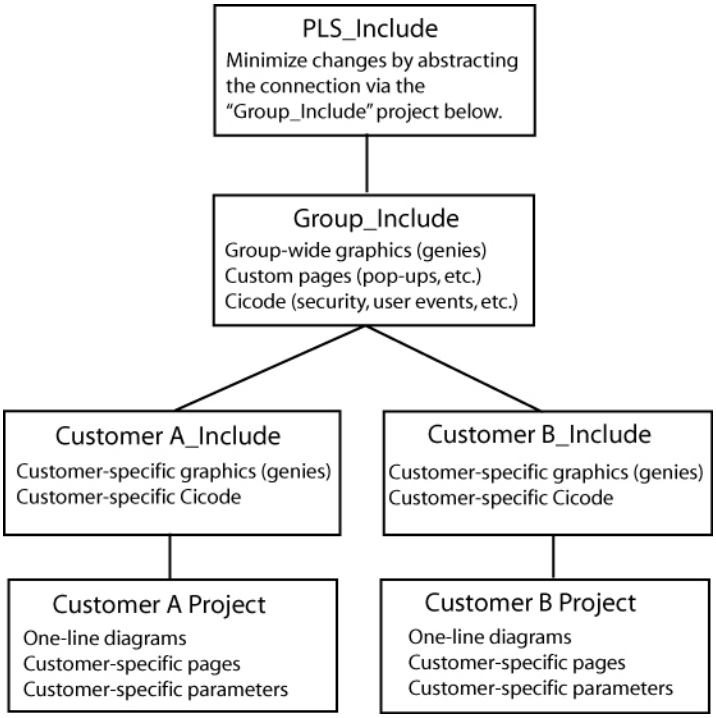
Although the designer may wish to organise each project in a particular way to suit customer's needs, the following is a recommended best practice:

1. Keep original ‘Master’ copies of the PLS_Example and the PLS_Include projects for reference.
2. The Services group may develop a group-wide “include” project that will act as a conduit between the PLS_Include project and all customer projects (e.g., "Group_ Include"). This will make the upgrading of PLS_Include much easier, as it will be the only project that must be modified to be compatible with the new version in the group-wide include project.

   Any changes made to the PLS_Include project should be made at the Group_ Include project level. This would involve removing portions of the code from the PLS_Include project, modifying the code and saving it in the Group_Include project. By removing (or commenting out) the original code and placing the new code in the Group_Include project, a layer of abstraction is preserved, further simplifying the upgrade process. In other words, the only changes to PLS_Include should be code removal.
3. When a new customer project is started, also create a customer-level “include” project.

   Always back up and restore the customer project and its associated include projects together.

   Always restore include project(s) before restoring the customer (or top-level) project.



4. Upgrading PLS_Include:

Document all changes to PLS_Include. This is absolutely necessary when upgrading to a new version of the PLS_Include project.

Minimize changes to the PLS_include project.

Abstract as many changes to the PLS_Include project as possible. This should be accomplished by the use of multiple include projects as shown in the diagram above.

New versions of PLS_Include will include a detailed description of each change, allowing you to merge the old and new versions.

New versions of PLS_Include will maintain backward compatibility where possible.

On the Graphics page, what do I need to know about creating genies?

1. Creating a new genie:

   The easiest way to create a new genie is to use an existing genie from the library. This ensures that the new genie is compatible with the PowerLogic SCADA system, and that it preserves these features:

   — Compatibility with animated busbar logic is maintained. This gives you the ability to use the setting from the Color Change Tool for animated busbar colour.

   — A sizing guide (a dotted rectangle) is included; it displays during graphics edit mode. This guide ensures that new genies can be swapped with existing genies without the need to recreate portions of the drawing.Save the new genie in the appropriate project (do not overwrite the provided genies).

   Save the new genie in the appropriate project (do not overwrite the provided genies).

2. Copying a genie to another project:

   Open the genie in the graphics editor, and do a <save as> into another project/library.

3. Genie Form Files:

   Any new genie (copied or created) will not have a .frm file entry associated with it. While the new genie is functional, it will show a cryptic unformatted properties box in the Graphics Editor. You can create your own .frm file with the needed entries by following the instructions available in the Citect Knowledge base.

   If you want to use the .frm dialog box that belongs to the genie you copied, go to the PLS_Include library; locate the .ctm and .ftm files. Each library has its own .ctm/.ftm files that include the description for every genie in the library. (This is an ascii text file that you can open in any text editor.) Find the genie that you copied (or on which you're basing the new form). Copy the portion that matches the copied genie, and create a .frm file that has the desired library name on it. Copy in the text from the .frm file. Restart Citect Explorer or it will not detect the new .frm.

4. Genie Sizing

   The provided genies come in two sizes: size 1 and size 2. When making a new genie for reuse among multiple projects, it will be beneficial to create a genie for both sizes. Follow the same steps for both sizes (sizing guides are provided for both sizes).

How do we customize existing templates?

Template Editing:

All objects on the page contain one or more Animation Numbers (ANs). Symbols take one AN while genies may take tens to hundreds of ANs. Placeholder ANs allow you to add objects to a template that is used on existing pages.

Some default templates contain ANs that have associated Cicode functions that rely on the animation number to remain a fixed number. For this reason, we have pre-allocated a set of ANs for the default templates. The base normal template uses ANs 1–263, and it has placeholder ANs from 264–500. When customizing this template, you should use the placeholder ANs as required.

You can place an AN (or a placeholder AN) on the page by using the “System 3.x/4.x tools available in the Graphics Builder under Tools< Options.

The default template uses ANs 1–263 and it has placeholder ANs from 264–500.

New objects added to a page or template will take the next available ANs. Any previously used (and now abandoned) ANs will be reused.

To add an object on the template, you must:

1. Open the template.
2. View the page properties and change the page width to 2000. This will reveal the hidden placeholder ANs on the page. You may have to change the width to a wider dimension for widescreen templates.
3. Determine how many ANs the new object requires. (You can place the new object on a blank page and then view the object in the object browser.)
4. Remove exactly the amount of ANs to allow the new object to be placed on the template. Remove ANs beginning with the lowest available placeholder AN (in the default template, this would be 264).
5. Place the object on the template.
6. Save the template.
7. Create a new page based on this template.
8. Drop a numeric object on the page.
    a. This object’s AN should be 502 (501 is reserved for placing the template on the page).
    b. If the object has an AN less than 502 then you have unused AN(s) on the template. This must be resolved. (Place additional ANs on the template to rectify this situation.)
    c. If the object has an AN greater than 502 then you have too many ANs on the template (a AN on the template is going beyond the 500 limit). You must find the culprit (via the object browser) and rectify the situation using the steps above.

How do I change the default pickup/dropout text for alarms?

The default ‘pickup/dropout’ text is shown as Appearance/Disappearance.

To change globally:

This text may be changed by configuring INI parameters in the citect.ini file. For more information, see the System Integrator’s Guide (*Graphics Library Parameters* on page 159)

This is the *global fallback text* that will be used in the event that pickup/dropout text is not specific on a per-alarm basis in the Device Profile. You can specify the per-alarm pickup/drop-out text on the profile tab in the Profile Editor.

To change on an individual basis:

See the System Integrator's Guide (*Viewing Device Profiles* on page 44, "Alarm On Text" and "Alarm Off Text").

What can I modify during runtime?

Set-point Configuration during runtime: In the Alarm Server entry in the project editor, view the advanced properties (F2) and set publish alarm properties = TRUE

Events Accessible at runtime: By default, 5000 events are accessible (view, filter, print) during runtime. Additional events are logged to disk. The number of events accessible at runtime may be changed by the [Alarm] SummaryLength parameter in the INI file.

Why do the browser navigation buttons not work?

If you find that the browser navigation buttons do not work when you are viewing the runtime window, you have probably added a new page, but have done the following:

- add the startup page to the [MultiMonitors]StartupPage parameter
- In the Computer Setup Wizard, General Options Setup screen, do not change the StartupPage field; leave it as <default>.

What can I set up in logging and archiving?

Event Logging and Archiving:

Event fields that are logged to disk may be configured by adjusting the AlarmFormat parameter.

There is no automatic maintenance performed on the log files. It is important that the log/waveform data be cleared out periodically (to prevent the hard drive from filling up; this does not affect performance).

On the Graphics page, how are colour changes displayed for voltage levels?

Definitions:

Voltage Level: This is a number from 0–3 that relates to the selected “ACTIVE” colour in the Color Change Tool. This is a graphics-level number that is used for animation purposes (in expressions and Cicode).

Nominal Voltage: This is the actual voltage present on the line at the customer’s site.

1. Before creating the first graphics page of a new project, determine the nominal voltages that will be assigned to voltage levels 0–3. Examine the customer’s power system to determine nominal voltages required.
2. As a team, decide which voltage level will represent the highest nominal voltage present at a customer’s site (0 or 3). This will provide for easier code re-use and project management.
3. Document the nominal voltages and voltage levels for each project, and keep them with the project (in a text file in the project directory or in a comment in a Cicode file).

4. Run the Color Change tool after any changes to graphics and at the very end of the development process. This ensures that all graphics have the same colour configuration.

   Always update pages after running the Color Change Tool.

   The Color Change Tool will change colours on all projects that are included by the top-level project. Care must be taken to not 'colour change' a project unintentionally.

   Example Project Documentation:

Colour Change Voltage Level	Nominal Voltages	Colour
0	240	blue
1	480	green
2	4160	purple
3	12470	red

How do I create and configure busbars?

When drawing one-line diagrams:

Analyse the drawings at a customer site.

Number the busbars consistently on the one line diagram(s). If busbar 14 spans across multiple pages, it should be numbered busbar 14 on all pages. Label the voltage level (0–3) on each busbar.

Uses for Line Active:

Page Connections: Many one-line diagrams will span multiple pages in PowerLogic SCADA. To connect these pages together, you must use the line active field of the 'incomers' of the second and subsequent pages. Set the line active field of the incoming busbars on these pages to an expression that references the nearest device on the same busbar of the previous page.

Metered Busbar: Many busbars are actually metered. It is more accurate to allow these metering devices to dictate state than to rely solely on the simulation (see Expressions below).

Configuration of Line Active:

Simulation: If the Line-Active field is left blank, the busbar state will be determined by surrounding devices.

Expressions:

A Cicode expression in the form of Device\Tag > Nominal Voltage (I.E., S1_B1_P1_CM41\MMXU1\PhV\zavg > 120).

If the expression is TRUE, the ACTIVE colour will be shown. The active colour is determined by the voltage level assigned.

If the expression is FALSE, the DE-ENERGIZED colour will be shown.

Hard-Coded:

If no upstream devices are available (in the event of an incomer, for example), you may have no other choice than to 'hard code' this field to a '1'. This forces the busbar to always be ACTIVE.

What INI parameters should I use for debugging?

We recommend that you contact Technical Support before performing any debugging.

Parameter: [PowerLogicCore]

DebugCategory = Error

DebugLevel = All

LogArchive = 1

LogFileSize = 200000

Parameter Details:

DebugCategory defines which message categories to log. (See table below).

DebugLevel defines debug levels of messages to be logged. (See table below).

LogFileSize specifies maximum log file size in kilobytes. After reaching the limit, file is saved with suffix ".BAK" and new file is started over.

Debug Levels

The following debug levels are accepted by PowerLogic driver core library:

WARN: log all warning level messages

ERROR: log all error messages

TRACE: log all trace messages

DEBUG: log all debug messages

ALL: include all level messages

Debug Categories

PowerLogic core library and drivers messages are grouped in categories. Each of these categories can be enabled independently from others in any combination.

MISC: miscellaneous messages that do not all into any other category

ALARM: messages related to alarms, regarding collection and detection

WAVE: messages related to waveforms -- waveforms download, processing

REAL: messages related to real-time data collection

STATE: messages related to internal object-state changes

CORE: core events that do not fall into driver-specific logic

MODBUS: TCP/MODBUS messages

ENTRY: trace messages produced when driver API entry points are called by PowerLogic SCADA

PORT: traces related to the port events

UNIT: traces related to specific unit events

WAVETOALARM: not used

DATAPOINT: debug messages related to data points

STATISTICS: enables driver statistics data output

COMMAND: messages related to commands

REPLICATION: messages produced by replication subsystem

ALL: enables all categories

Parameter: [Debug]

Menu = 1

Parameter Details:

The Menu parameter determines whether the Kernel option is displayed on the control menu of the PowerLogic SCADA runtime menu. This can also be enabled using the Computer Setup Editor.

How do I tune my system for best performance?

There are several parameters that you can use to enhance your system's performance:

Driver-tuning parameters:

Parameter (Back Polling Rate): **[SEPAM40]**

CacheRefreshTime = 1000

InitUniCheckTime = 120

Retry = 3

Timeout = 1000

Parameter Details:

The CacheRefreshTime parameter controls the maximum rate at which the driver will attempt to repopulate its cache. If the driver cannot refresh its cache within the time period specified, it will collect data as fast as the network allows.

This back polling rate can be global to all devices or tuned up to a specific I/O device.

The InitUniCheckTime parameter controls how long the driver will wait before attempting to bring a device online after it has gone offline. This value can be decreased to bring offline devices back into service in a shorter period of time. In a multi-drop scenario, this time should be relatively long, to prevent init unit requests from stalling communications to the rest of the devices on that port.

The Retry parameter defines the number of retry attempts for specific MODBUS requests. Retries will only occur in response to the MODBUS errors which are defined below.

The Timeout parameter controls how long the driver will wait for a response from a device before setting that device as offline. This value should be greater than the device/gateway timeout period.

Parameter: [Device]

WatchTime = 5000

Parameter Details:

Device WatchTime is the frequency that PowerLogicSCADA checks devices for history files and flushes logging data to disk.

Default: 5000

Range: 1000–3600000 milliseconds.

Miscellaneous Parameters

Parameter: [Kernel]

Task = 20000

Parameter Details:

Kernel Task is the number of tasks. Increasing the number of kernel tasks is used when “Out of Kernel Task” message is received. The change will be likely for large systems.

Default Value: 256

Range: 50–32767

Parameter: [Page]

ScanTime = 250

Parameter Details:

Page ScanTime determines how often the Animator refreshes a graphics page at runtime.

Default: 250

Range: 1–60000 milliseconds

Parameter: [ALARM]

ScanTime = 500

Parameter Details:

Alarm ScanTime determines the rate at which alarms are scanned and processed.

Default: 500

Range: 0–60000 milliseconds

If a tag is configured in PowerLogic SCADA, how is it polled in the device?

In other words, is a tag only polled on demand when it is requested by a client (for example, when the operator displays a page with the tag on it)? Or are all configured tags polled all of the time, with the relative polling rates/communications bandwidth carefully managed?

The ModNet driver polls real-time tags on a user demand basis (when a user opens a page with the tags on it). Therefore, the time to retrieve data will vary, depending not only on the communications bandwidth, but on the amount of data being requested. This can vary significantly, depending on which pages are displayed by the operators at any particular time.

The PWRMODBUS driver polls all configured tags; however, different types of tags can be polled at different relative rates, and the available communications bandwidth is carefully managed. This approach means that tag update rates are not subject to the scaleability issues associated with operator actions (as is the case for the ModNet driver). It is also advantageous in that performance issues associated with communications bandwidth or IODevice response times can be determined at SAT/time of implementation and are not subject to significant change during operation.

The different tag types can be allocated relative importance in data requests, expressed as a percentage. (See Bandwidth Allocation Parameters in *Performance Tuning Parameters* on page 146.) Keep in mind that any unused bandwidth allocation (from, for example, events retrieval) is made available for other data types to use. If the event does not need the default 25% allocation, it will be made available to the other parameters (real-time tag retrieval, etc). This potentially increases the update rate of real-time tags.

Additionally, the real-time tag relative scan rate based on priority can be set to three different levels. (See Tag Scan Rate Parameters in *Performance Tuning Parameters* on page 146.) This means that, if some real-time tags are more important that others, you can set their relative priorities. For example, configuration tags vs. important real-time tags vs. normal real-time tags.

Device popup from a one-line: Why do the fields overlap?

This is controlled by a parameter entry:

Section: Page

Name: EquipDetailDescLength (the total number of characters in a single row of this popup)

Default = 48. The problem will occur with a larger font or if the window is resized. The default value of 48 can be changed or the window and associated genies can be resized.

Can I change the %CLUSTER% name in the Profile Wizard?

No. If you change the placeholder %CLUSTER% to any other name in the Profile Wizard, the system will be unable to find the actual cluster to which it refers.

When viewing PowerLogic SCADA from a web client, I see a licensing error. How do I fix it?

First, verify that the appropriate licensing is available.

A web client requires the same number of ports open as a standard control or view-only client. You only need to allow communication to those servers from which the client needs information. For example, alarm, trend, reports, and I/O.

When you open PowerLogic SCADA from a client computer, and you see a licensing error, make these port settings. This will open both PowerLogic SCADA and the license key server (7001: USB sentinel key port) in the routers and firewalls between the client and the server. The ports that you need to open are listed in **boldface**:

Default Port	Server Type	Server Role
21	FTP server	page downloads for IDC
80	**Web server**	**project files for Web client**
2073	CTAPI	CTAPI communications
2074	client	Cicode debugging
2075	Reports server	Reports server communications
2076	**Alarm server**	**Alarm server communications**
2077	**Trends server**	**Trends server communications**
2078	**IO server**	**legacy I/O communications**
2079	IDC	Internet display server/client communications
2080	**Alarm server**	**alarm properties connector**
2081	time server	time server communications
2082	**IO server**	**publish subscribe I/O server communications**
20222	ODBC	ODBC server

A device can prevent writes to its registers: how do I ensure that writes are successful?

PowerLogic SCADA cannot provide feedback about whether a write to a device register is successful. If a device is capable is preventing (blocking) writes to its

registers (for example, Sepam), you need to verify that its “block” feature is not enabled. Do this at the device.

NOTE: In Cicode, you can also use the tagwrite function in blocking mode, i.e., bSync parameter = true; Check the return code: 0 = success, anything else = error. For more information, see the Cicode Programming Reference help file.

How do I prevent PowerLogic SCADA from accidentally making invalid areas in memory available to reads and writes?

PowerLogic SCADA normally optimises its packets for greatest performance. This optimisation can sometimes includes invalid areas of memory in devices. These invalid areas can be specifically defined and excluded from optimisation packets created by PowerLogic SCADA. See *Advanced Tag Blocking Capabilities* in *Performance Tuning Parameters* on page 146 for more information.

How do I create an audit in the Event Log for user logins and logouts?

```
//LOGOUT


FUNCTION
PLSLoginUser()


    //INT iPage = PageInfo(1);
    INT iPage = WinNumber();
    IF mbLoginFormShown[iPage] = TRUE THEN
    RETURN; //form already shown
    END
//prevent multiple forms
mbLoginFormShown[iPage] = TRUE;
IF (UserInfo(0) <> "0") THEN
        // Confirm User Action
        IF (0 = Message(StrToLocalText("@(Confirm)"), StrToLocalText("@(Logout)"),
        1+32)) THEN
        PLSAlmDspEventAdd(0, 0, 1, "User Logout", Name(), "Logout", "");
        Logout();
        END
        mbLoginFormShown[iPage] = FALSE;
        RETURN;
    END
    IF (0 = LoginForm())
    PLSAlmDspEventAdd(0, 0, 1, "User Login", Name(), "Login", "");
    END
    mbLoginFormShown[iPage] = FALSE;
```

END

Why do I get an "out of memory" message when I try to run the Automation Interface?

The problem is that you did not have Microsoft Excel installed before you installed PowerLogic SCADA. The only solution is to uninstall both Excel and PowerLogic SCADA; then, reinstall first Excel and then PowerLogic SCADA.

Why am I seeing #COM for circuit breaker status in the genie status page?

If this is a Micrologic P device, and it does not have a CCM, you will not be able to view data referring to circuit breaker status, e.g. racked in/racked out. When there is no CCM, the device profile should not have tags that refer to the CCM.

Index of Terms

A

access rights
  adding for users 44
  changing for users 128
  PowerLogic SCADA 127
  Windows integrated users 127
Adding_base_eng_unit 25
additional sources of information 15
address
  correcting missing or invalid 28
advanced tag block capabilities
  configuration notes 150
  invalid block tag definition syntax 150
  tag blocking notes 152
  write-only memory 151
  write-only tags 152
alarm "off" text 45
alarm "on" text 45
alarm codes
  unique identifier 41
alarm format code definitions 58
alarm log 113, 115
  column headings described 111
  filtering 113-115
  freezing the view 113, 115-116
  logging data to disk 114, 116
alarm states
  startup parameters 101, 159
alarm tags
  adding 92
AlarmFormat Summary parameter
  used to change column headings in alarm logs 112, 115
alarms
  adding an onboard tag 94
  associated with waveforms 106
  changing pickup/dropout text 174
  changing the severity 93
  historical logging parameters 94
  making a PC-based alarm digital 92
  onboard 92
  overriding ON/OFF text for onboard alarms 45
  pc-based 92
  setting up based on an enumeration 93
Alarms/Events page
  filtering alarms 115
  freezing the view 115
  viewing 110
AlarmSummaryLength parameter
  maximum number of entries in the Event Log 94
analysis page 116
animation numbers (ANs)
  used in objects 174
assigning tags to generic I/O points 33
assumptions
  about PowerLogic SCADA system integrators 8
  before system integrator can set up system 8
audible alarms
  setup 95
Automation interface
  "out of memory" message 74
  adding multiiple devices 74
Automation Interface
  required information 76

B

backup
  project file 96
base unit
  deleting 26
base unit and conversion
  adding 26
  adding and editing 25
before you use the Profile Wizard 68
block writes
  using 65
browser navigation buttons do not work 86
busbar connections in genies 87
busbars
  creating and configuring 176

C

CacheRefreshTime parameter 178
calculating disk storage 103, 117
CASE
  changing user access rights 128
category/subcategory
  adding 35
  deleting 35
  editing 35
  managing 168
changing alarm severity 93
circuit breaker status
  using special tags to control 55
Citect.ini
  used to customize a project 99
citect.ini file
  setting up multiple monitors 122
Citect.ini location has changed
  prompt 97
cluster
  adding 68
CLUSTER name (%CLUSTER%)
  changing 180
cluster setup
  redundantI/O servers 70
cluster setup in the Profile Wizard 72
colour change tool
  see Project Color Change Tool 97
colour changes for voltage levels 175
colours
  changing for busbars and circuit breakers 97
  setting flashing colours in the Graphics page 98
comment field
  used in translation 69

compile and run 13
Compile Errors window
viewing PowerLogic SCADA errors 88
computer setup wizard
startup page 86
Computer Setup Wizard
running after first compile 13, 68
configuration tools
described 9
control and reset tag addresses 41
control format codes 58
controls
how they operate 30, 42
conversions
to base units 24-25
creating
CSV file 34
device profile 44
engineering template 23
genie 90
one-line on a graphics page 86
PowerLogic SCADA project 85
project 49
CSV file
creating, viewing, printing 28, 34
CTZ file
backup file 96
custom control format codes 59
custom filters
alarms 114
custom genie
adding for Project Color Change Tool 97
custom genie, creating 90
custom icon 22
custom symbol
creating 91
custom tag
adding 38
consistency in names and attributes 168
deleting 39
deleting when associated with a device type 39
editing 39
customer support contact information 15
customizing a project with Citect.ini 99
customizing tag names 49, 53

D

data replication parameters 158
DBF add-in tool
using to update menu.dbf 89
DBF files
editing 89
debugging
INI parameters to use 177
Define_Dev_Types_Tags 27
demo mode
using 124
design-time environment
described 84
device
adding to the PowerLogic SCADA project 72
device name
requirements 170
device popup
overlapping fields 179
device profile
adding 46, 50
creating 44
defined 12
deleting 47
editing 47
removing from a PowerLogic SCADA project 73
device tags
defining 27
device type
adding 31
defined 12
defining 27
deleting 33
editing 32
generic types used in Graphics Test Mode 123
locking 22
unlocking 22
device type category/subcategory
adding 35
deleting 35
editing 35
device type drivers
INI parameters 100
list of supported 9
device type name
limitations 32
Device Wizard
before you use it 68
devices
adding multiple 171
devices in the PowerLogic SCADA project
editing 74
digital alarms
converting from PC-based analog 92
disabling "tag not exist" prompt. 88
disk storage calculations 103, 117
display projects button
adding a project in the Profile Editor 12, 51
documentation in this release 15
driver
choosing for device types 31
generic power device 9

E

engineering template
creating 23
engineering units
adding 26
adding from the Project tab 51
deleting 26
editing 26
English.dbf
used in translation 73

English.dbf file
used for translation 73
entering copy for translation 71, 73
enumeration
defining 55
setting up alarm based on 93
status enumeration logic code 62
equipment name
adding via Automation Interface 75
adding via Profile Wizard 72
using when creating a one-line 87
Equipment.Profiles
file storage location 52
viewing 50
errors and warnings
correcting in the graphics page 88
event log 113, 115
logging data to disk 114, 116
viewing, printing, and exporting 111
events
historical logging 94
storage limitations 94
Example Automation.xls
using to export multiple devices 75
expanding PowerLogic SCADA screens
F2 key 69
exporting projects
before you export 52
Profile Editor and server on different computers 53
refreshing tags that have been changed 52

F

F2 key
use to expand PowerLogic SCADA screens 69
use to expand screens 69
FAQs 167
"out of memory" message in Automation Interface 182
#COM error for circuit breaker status 182
adding more than one device at a time 171
adding/removing devices in Profile Wizard 170
auditing user logins/logouts in the Event Log 181
browser navigation buttons do not work 175
can I change %CLUSTER% name 180
changing default pickup/dropout text 174
colour changes for voltage levels 175
consistency in custom tags 168
creating and configuring busbars 176
creating device profiles without PowerLogic drivers 167
creating users in Runtime 171
device name requirements 170
device type vs device profile 168
device writing to its registers 180
how to use MODBUS communications methods 170
if tag is defined, is it polled 179
INI parameters used for debugging 177
ION E or SMS installed with PowerLogic SCADA 167
licensing error at web client 180
managing categories and subcategories 168
managing projects in PowerLogic SCADA 172
managing units and unit templates 168
multiple engineers working at same time 167
naming conventions for servers/clusters 171
overlapping fields in device popup 179
performance tuning parameters 178
preventing reads/writes to invalid ares of memory 181
re-using Profile Editor projects 169
sharing information between computers 169
synchronize new PC with master PC 170
what are genies 173
what can be set up in logging and archiving 175
what can I modify during runtime? 175
working with templates 173
FileFormat parameter
historical logging of alarms and events 94, 114, 116
filtering the alarm and event logs 114
flashing colours
setting up in genies 98
format codes
alarm definitions 58
control definitions 58
custom control 59
definitions 56
predefined control 58
real-time definitions 58
reset 58
freezing the view of an alarm window 113, 115-116
frm file
associated with new genie 173
FRM file
formatting genie appearance 90-91

G

generic device types
Graphics Test Mode 123
generic I/O points
assigning tags 33
generic MODBUS drivers
included in system 9
genie
adding custom 97
adding to a drawing 86
creating and editing 90
icon 86
when they do not change color 97
genie in Graphics page 173
genie libraries
ITEM1 135
PLS_ALARM 129
PLS_ANSI_BASE_1 / PLS_ANSI_BASE_2 129
PLS_ANSI_CB_1 / PLS_ANSI_CB_2 130
PLS_ANSI_SW_1 / PLS_ANSI_SW_2 132
PLS_DISPLAY 132
PLS_GEN_BASE_1 / PLS_GEN_BASE_2 132
PLS_GEN_CMD_1 / PLS_GEN_CMD_2 133
PLS_GEN_EQ_1 / PLS_GEN_EQ_2 133
PLS_IEC_BASE_1 / PLS_IEC_BASE_2 134
PLS_IEC_CB_1 / PLS_IEC_CB_2 135
PLS_IEC_SW_1 / PLS_IEC_SW_2 135

PLS_METER 135
genie status page 117
getting more information 15
graphics library parameters 101, 159
graphics page
adding in PowerLogic SCADA 86
setting as startup page 86
Graphics runtime environment
editing appearance 89
graphics test mode
enabling 123
using 123

H

historical parameters of alarms
setup 94
historical trends
viewing from a one-line 121

I

I/O server
adding before using the Equipment Wizard 68
icons
locked and custom 22
IEC61850 tags
categories 55
construction 54
logic codes described 59
INI parameters 137
INI parameters for supported device type drivers 100
INI parameters to use in debugging 177
instant trend (Analysis Page)
described 117
intervals
trend tags 46
invalid or missing tag address
correction 28
IP address
adding for device in the Profile Wizard 73
entering in Example Automation.xls 76, 80

L

licensing error at web client 180
line active
used to link pages in multiple-page drawings 88
locked device type
cannot delete 33
unlocking 22
locked icon 22
locking a device type 22
logic codes
definitions 59
loopback address
used for network addresses 68-69

M

master.dbf
using to update menu.dbf 89
menu.dbf
columns described 90
used to edit Graphics page menu information 89
metadatafields.xml
use to edit alarm filters 115
MicroLogic P
#COM error on genie status page 118, 182
Microsoft Excel
using to add devices to the database 75
MODBUS communications methods 170
multi-state colours 98
Multi-state type
genies 97
MultiMonitors Startup Page parameter
use to make a new graphics page the startup page 86
multiple engineers creating device types and profiles 167
multiple monitor support 122

N

network address
adding 69
network address and servers
adding before using Profile Wizard 68
NetworkTagsDev I/O device
adding 70
normally closed
setting up controls 30

O

ON/OFF text
onboard alarms 45
onboard alarm tag
adding 94
onboard alarms 92
one-line
adding to a graphics page 86
out of memory message
when running Automation Interface 74

P

pack
after deleting a cluster 69
using to delete data 91
parameters
advanced tag block capabilities 149
alarm and event state 101, 159
AlarmFormat 114
AlarmSummaryLength 94
AlarmUseConfigLimits 92
Bandwidth Allocation (port level) 146
BandwidthAllocation (device level) 146

cacheRefreshTime 143
CCM (Micrologic) 157
Citect INI 137
ClientAutoLoginMode
  used to set up Windows integrated user 127
CommandsBandwidth 146
data replication 158
database described 137
enableScatteredReads 147
EventBandwidth 146
EventIdle 156
EventTable 156
FileReplicationDst 158
FileReplicationSrc 158
graphics library 101, 159
HighScanRate 148
initCacheTimeout 142
initUnitCheckTime 142
kernelStatisticUpdateRate 139
LowScanRate 148
maxBlockSize 148
Micrologic 157
minBlockSize 148
ModbusBase 144
module-specific packet blocking optimisation settings 157
MultiMonitors StartupPage
  setting the startup page 86
packet blocking optimization 147
percentBlockFill 147
performance tuning 146, 178
PLSEventLog 94
PLSEventLogFileFormat 114, 116
PowerLogic INI driver 139
PowerLogicSCADAGraphicsTest 123
RegMode 144
retry 140
retryexception 141
retrytimeout 140
Sepam Event Reading 156
standbyCheckTime 141
standbyRefreshRate 141
statusRegister 143
statusregistertype 143
statusRegisterType 143
statusUnitCheckTime 142
Tag Scan Rate 148
time synchronization 100
time zone settings 100
TimeBandwidth 146
timeout 140
timeZone 145
usewritemultiregistersonly 139
watchtime 139
waveform 154
WaveformBandwidth 146
WaveformCheckTime 154
WaveformMatchMargin 154
WaveformsDisable 154
WaveformZone 155
pc-based alarms 92
  setting up 92
performance tuning parameters 146, 178
pickup/dropout text
  changing for alarms 174
PLS_Example project 13, 84
PLSAnimBusbar()
  Cicode function for custom genie animation 91
PLSEventLog parameter
  allowable values 114, 116
  historical logging of events 94
PMCU
  adding onboard alarms and waveform captures 93
  defined 165
polling rules for tags 179
PowerLogic Driver Parameters 139
PowerLogic SCADA
  described 8
PowerLogic SCADA naming conventions 171
PowerLogic SCADA project
  creating 85
predefined control format codes 58
predefined project
  PLS_Example 84
printing
  CSV file 34
Profile Editor.exe.config
  using to change trend tag scan intervals 102
Profile Wizard
  adding a device 68
  before you use it 68
  editing device information in a PowerLogic SCADA project 74
  removing a device 73
profiles
  adding, editing, deleting 45
  enabling waveforms 44
  file storage location 52
  name limitations 47
  viewing 44
project
  adding 50
  creating in PowerLogic SCADA database 52
  deleting 51
  editing 51
  exporting 51
  name limitations 50
  reusing 85
  setting up 49
Project Color Change Tool 97
  adding a custom genie 97
  changing colours of busbars and circuit breakers 97
project XML files
  storage location 52
projects
  managing in PowerLogic SCADA 172
  re-using 169
projects do not display in Color Change Tool 97
Publish Alarm Properties field
  enabling the Setpoints page 68
PwrLineColor
  setting in genies 97

Q

quick start 12

R

re-using projects 169
read logic codes 60
read/write logic codes 65
real-time format codes definitions 58
real-time lists
  viewing from genies 121
redundant I/O servers
  cluster setup 70
refreshing tags in the Project tab 49, 52
reload button
  refreshing view of the runtime environment 90
  runtime environment 110
reset format codes 58
resets
  how they operate 30, 42
restore
  project file 96
reusing a project file 53, 85
runtime environment
  described 9
  reloading 90

S

save_dbf.xls
  using to save edited DBF files 89
SCADA
  defined 8
scan intervals
  trend tags 102
scattered reads
  using enableScatteredReads parameter 147
screen resolution
  Profile Editor 22
selecting 24-25
servers
  adding 69
  adding to PowerLogic SCADA 68
  described 68
servers/clusters
  naming conventions 171
setpoints and delays
  entering 92
  entering from the runtime environment 120
setting up
  alarm based on an enumeration 93
  audible alarms 95
  device type categories 34
  normally closed/normally open controls 30
  pc-based alarms 92
severity
  changing for alarms 93
startup page
  setting 86
storage limitations
  calculating disk space 103, 117
support
  technical 15
symbol
  creating custom for custom genie 91
synchronize new PC with master PC 170
system creation steps 12

T

tag
  customizing the name 49
  IEC61850 described 54
  refreshing in the Project tab 49
tag-polling rules 179
tag addresses
  adding 43
  controls and resets 41
  correcting missing or invalid 28
  editing 40, 43
  onboard alarm 41
  real-time 40
tag definitions
  printing the CSV file 34
tags
  assigning to generic I/O points 33
  scanning intervals for trend tags 46
technical support 15
template
  creating 23
  deleting 25
templates
  create or edit (for units) 24
  working with 173
test mode
  graphics page 123
The_Profile_Wizard 67
time synchronization parameter 100
time zone setting parameters 100
transferring device types/profiles
  between computers 169
translation
  using description fields 71
trend tag intervals 117
trend tag limits 46-47
trend tag scan intervals 46, 102
trending (Analysis Page)
  described 117
trends
  setting up in Profile Editor 46
  viewing from within a one-line 121
typical workflow
  creating a graphics page 83
  creating a project in Profile Editor 49
  creating a system 12
  Profile Editor 18

U

uninstalling and reinstalling PowerLogic SCADA 11
unique ID
  determining for onboard alarms 41, 94
units (engineering) and conversions
  adding 26
  creating or editing a units template 24
  deleting 26
  deleting a unit template 25
  editing 26
units/unit templates
  managing 168
update pages
  use to update after colours changed. 98
user access
  adding and modifying 125
  changing rights 128
  limiting privileges 126
  PowerLogic SCADA 127
  Windows integrated 127
users
  creating in Runtime environment 171
using block writes 65

V

value type
  controlling real-time display 87, 119
Variable tags not found
  disabling the prompt 88
voltage levels
  colour changes 175

W

waveform parameters 154
waveforms
  enabling in Create Device Profiles tab 44, 50
  enabling onboard 93
  linking to alarms 106
  management 104
  special digital tags 106
  storage rules 105
  viewing from the Alarm Log 113, 116, 118
  viewing in the runtime environment 119
Windows integrated user accounts 127
Windows security
  using to remove user access rights 126
write logic codes 65
writes (logic codes) 65

X

XML files
  storage for project files 52

PowerLogic™ SCADA System Integrator's Guide
User Guide

Schneider Electric
295 Tech Park Drive, Suite 100
LaVergne, TN 37086 USA

For technical support:
Global-PMC-Tech-support@schneider-electric.com
(00) + 1 250 544 3010
Contact your local Schneider Electric sales representative for assistance or go to www.schneider-electric.com

63220-100-200D1 05/2011
Replaces 63220-100-200C1, dated 01/2010