Orchestra User Manual and Help

USER MANUAL

HED, Inc.

12. Application Notes 267

Introduction

1 Introduction

Orchestra® is a suite of software that lets users build, implement, and manage their vehicle automation control strategy.

Orchestra® is an integrated development environment that contains the following tools:

Arranger-Composer™ 138
Conductor™ 168
Downloader 216
Application Configurator 216

The manual's purpose is to assist the user to create an application using off the shelf CANLink® Modules, regardless of software experience.

As a companion to the user manuals, there are downloader guides to help users download and install the software.

Related:

Writing Applications 12
System Requirements 13
Term Definitions 13
License Levels 14

1.1 Writing Applications

Applications can be written exclusively by Hydro Electronic Devices (HED®), the original equipment manufacturer (OEM), or a combination of both. The applications also have some flexibility in terms of how they are written. Based on the requirements of the application and the customer, Orchestra® allows the software to be written with rungs using ladder logic or coded using C or C++.

Related:

Rungs 106
Coding Within Orchestra 2 324
Using Presto 334

1.2 System Requirements

The minimum system requirements to install and run Orchestra® include:

- Minimum 250 MB free disk space
- Minimum 1024 x 768 screen resolution monitor
- RS-232 or USB communications port

In addition, depending on the application, the following optional equipment may be required:

- GridConnect CAN Tool
- Additional USB port to install a hardware lock key (a dongle)
- USB to RS-232 converter

Related:
License Levels 14

1.3 Term Definitions

This section outlines words used in this manual that users will need to know before reading further.

- **Application** – Software created by the user to control module specific functions. This software is downloaded to the master module.
- **CAN** – This is how the modules communicate with each other. Please reference Bosch 2.0 A and B Controller Area Network Specification.
- **CANLink® Module** – HED® product utilizing the CANLink® protocol. A combination of hardware and software. This includes inputs, outputs, displays, modem, etc.
- **CANLink® Protocol** – HED® proprietary compatible CAN protocol.
- **Display** – Programmable piece of hardware that can give a visual representation of the application
- **HED®** – Hydro Electronic Devices
- **I/O** – A module's means of interface to the physical world – short hand for inputs and outputs.
- **Master Module** – The module to which the application software resides
- **I/O Module** - Client module that has no application software and interfaces its I/O to the master module
- **OEM** – Original Equipment Manufacturer
- **User** – Person operating Orchestra®

Related:
Properties List 248
Glossary 263

1.4 License Levels

1=highest, access to everything, 2, 3, 0=access only to the downloaders

	Access Levels	**3**	**2**	**1**	**0**
Software	**Application Configurator**	x	x	x	
	Conductor™	x	x	x	
	Downloaders	x	x	x	x
	Arranger			x	

Dongles are physical licenses. The user can order a software license from HED®. This license will run with Orchestra® web server, regardless of dongle level.

Read/Write Levels

For data items or modules, there is a read level and a write level. In Conductor, the read level lets the user see parameters of the level and lower. The write level lets the user debug and set data items of their level and lower.

Related:

System Requirements 13

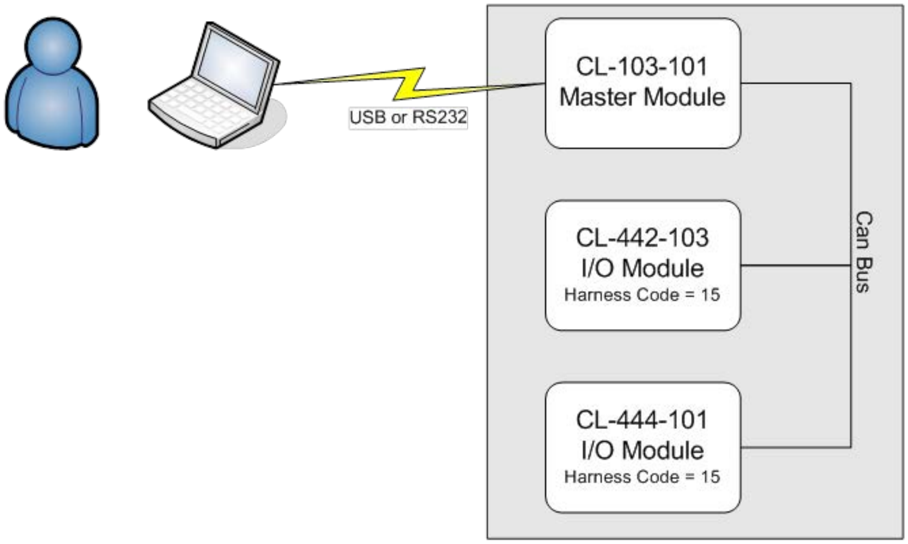
Typical System Overview

2 Typical System Overview

2.1 What a Typical System Looks Like

A typical system consists of a number of CANLink® Modules that work together to control some aspect(s) of a vehicle. HED® Control Module to Client modules use proprietary CAN communication protocol called CANLink®. Control modules can also communicate with other CAN protocols such CANopen. Other modes of communication to the Control module are J1708 and RS232. Not every module supports all of the protocols, and it is highly advised to consult data sheets to determine what is and is not supported with each module.

The CANLink® Modules work together in a way that there is a single master module that will control the other modules that are designated as I/O modules, up to 40 total modules, to accomplish the goal of the system. Each system must have a master module, but not every system needs to have I/O modules; the master module can act as both. What is needed will be based on the requirements of the application.



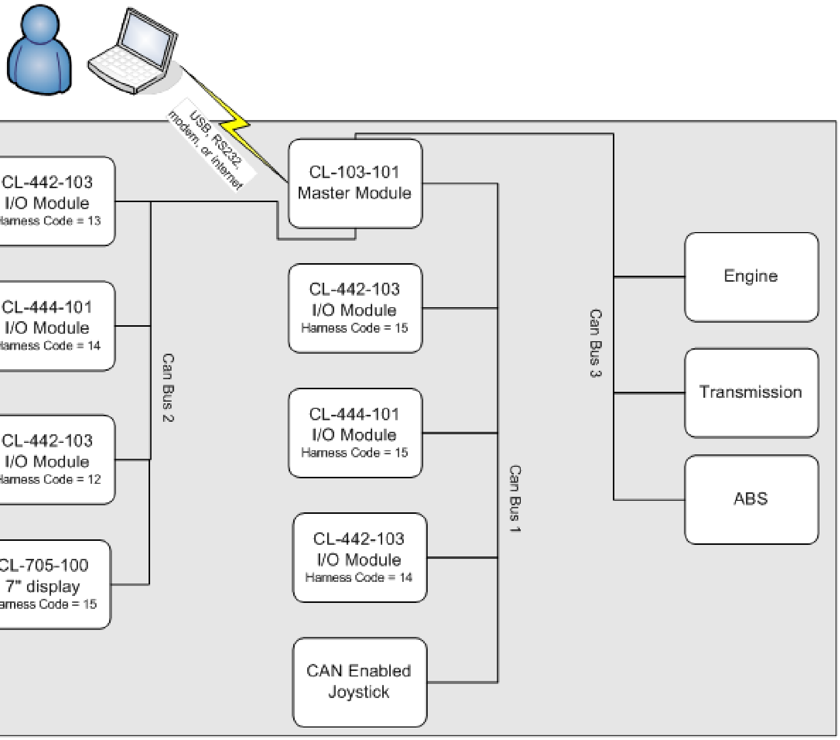
Simple System Example

Related:

Identifying Master Modules and I/O Modules 17

2.2 Identifying Master Modules and I/O Modules

Each I/O Module has a module type and harness code, where the type is a unique identifier based on the hardware itself, and the harness code allows for multiple modules with the same type to exist within the same system. The combination of the two identifies the I/O Module to the master module. Within a system, two I/O Modules of the same type cannot have the same harness code, but two modules can have the same harness code as long as the type is different.


USB, RS232, modem, or internet
CL-442-103 I/O Module Harness Code = 13
CL-444-101 I/O Module Harness Code = 14
CL-442-103 I/O Module Harness Code = 12
CL-705-100 7" display Harness Code = 15
Can Bus 2
CL-103-101 Master Module
CL-442-103 I/O Module Harness Code = 15
CL-444-101 I/O Module Harness Code = 15
CL-442-103 I/O Module Harness Code = 14
CAN Enabled Joystick
Can Bus 1
Can Bus 3
Engine
Transmission
ABS

Advanced System Example

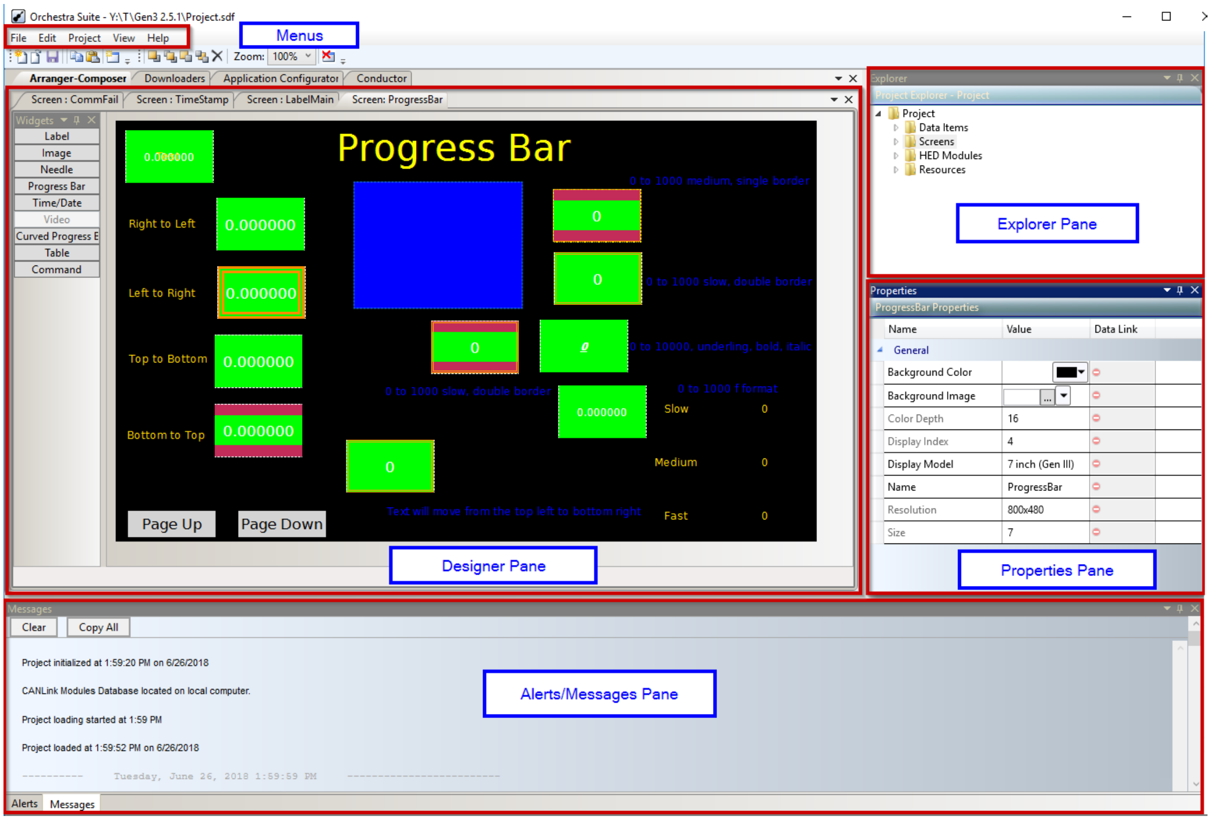
Related:

What a Typical System Looks Like 16

Screen Layout

3 Screen Layout

The Composer™ screen has four separate panes: the Designer pane, Explorer pane 26, Properties pane 32, and Alerts pane 32. Along with the four panes, there are the File, Edit, Project, View, and Help drop-down menus. Each pane can be docked and resized independently.



Name	Value	Data Link
General		
Background Color	■	⊖
Background Image		⊖
Color Depth	16	⊖
Display Index	4	⊖
Display Model	7 inch (Gen III)	⊖
Name	ProgressBar	⊖
Resolution	800x480	⊖
Size	7	⊖

```
Project initialized at 1:59:20 PM on 6/26/2018

CANLink Modules Database located on local computer.

Project loading started at 1:59 PM

Project loaded at 1:59:52 PM on 6/26/2018

----------     Tuesday, June 26, 2018 1:59:59 PM     ------------------------
```

Orchestra Screen Layout

Each of the panels in Orchestra® Suite may be docked to an inside edge of the main window, or undocked and left to float outside of the main window. Panels that are docked can be configured to be hidden (collapsed) or shown. Docking options may be accessed by choosing the docking menu in each panel.

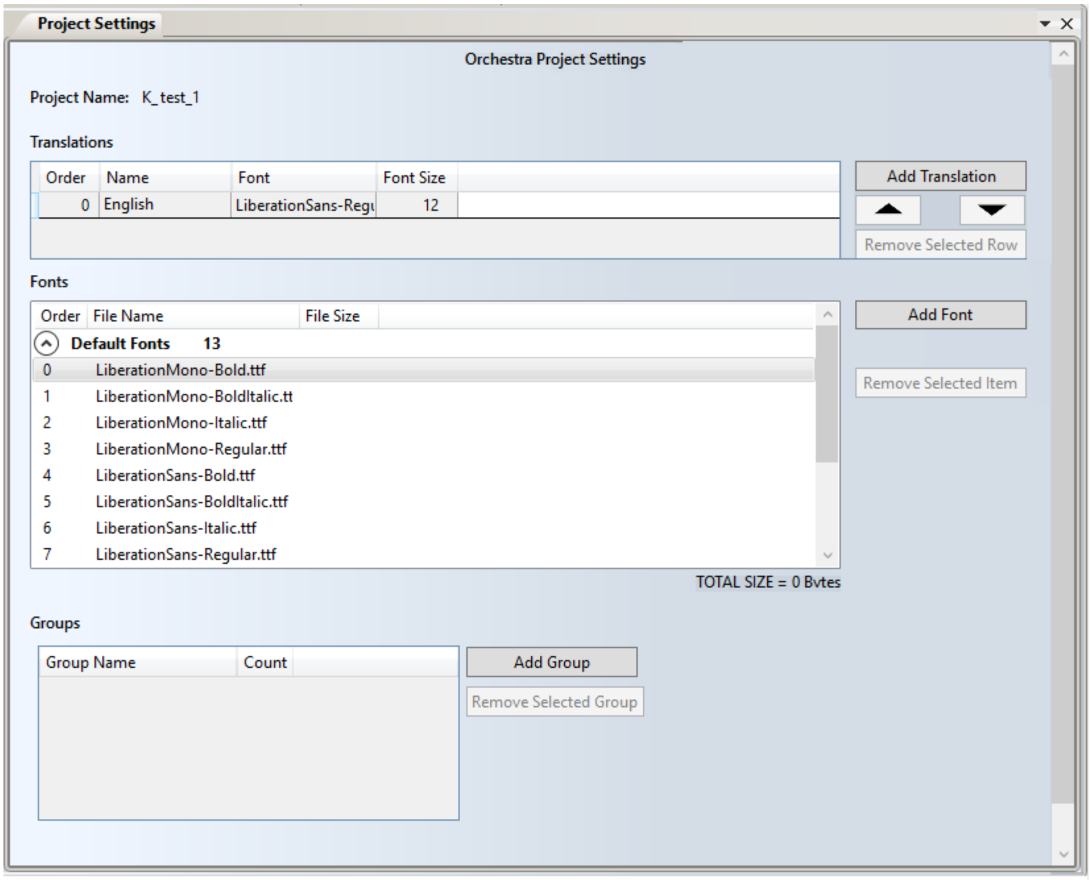
Note: The Alerts Pane 32, Properties Pane 33, and Explorer Pane 26 can be undocked, redocked, and resized as the user sees fit. If one of those panes was closed, it can be accessed again using the View pull down menu.

Related:

Project Menu 22
Explorer Pane 26
Alerts Pane 32
Messages Pane 32
Properties Pane 33
Designer Pane 37

3.1 Project Settings

The Project Settings screen lets the user edit translations, fonts, and groups. Each section of the screen lets users add and remove items.



Orchestra Project Settings

Project Name: K_test_1

Translations

Order	Name	Font	Font Size
0	English	LiberationSans-Regu	12

Fonts

Order	File Name	File Size
Default Fonts	**13**	
0	LiberationMono-Bold.ttf	
1	LiberationMono-BoldItalic.tt	
2	LiberationMono-Italic.ttf	
3	LiberationMono-Regular.ttf	
4	LiberationSans-Bold.ttf	
5	LiberationSans-BoldItalic.ttf	
6	LiberationSans-Italic.ttf	
7	LiberationSans-Regular.ttf	

TOTAL SIZE = 0 Bytes

Groups

Group Name	Count

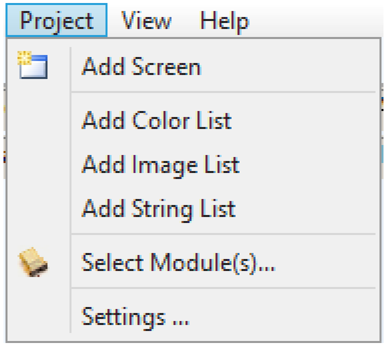
Project Settings Page

Related:

Edit Project Settings 308
Add String List 299

3.2 Project Menu

The Project menu has several options that users can use to build their project. Users can add screens 22, color lists, image lists, and string lists. They can also select modules from the list, and adjust project settings.



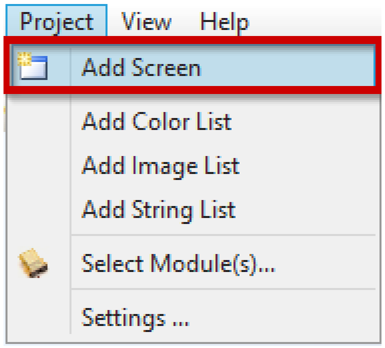
Project Menu

Related:

Add Screen 22
Add Color List 23
Add Image List 23
Add String List 24
Select Module(s) 24

3.2.1 Add Screen

Click this to add a screen to the project. The new screen is added to the Screens folder in the Explorer pane.



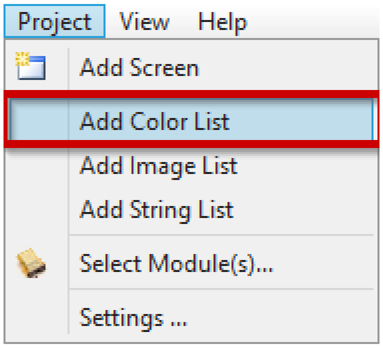
Project Menu: Add Screen

Related:

Add Screen 284
Project Menu 22
Add Color List 23
Add Image List 23
Add String List 24
Select Module(s) 24
Settings 25

3.2.2 Add Color List

Click this to add a color list to the project. The new color list is added to the Color Lists subfolder of the Resources folder in the Explorer Pane.



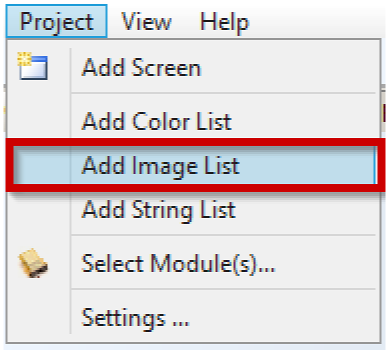
Project Menu-Add Color List

Related:

Add Color List 292
Project Menu 22
Add Screen 22
Add Image List 23
Add String List 24
Select Module(s) 24
Settings 25

3.2.3 Add Image List

Click this to add an image list to the project. The new image list is added to the Image Lists subfolder of the Resources folder in the Explorer Pane.



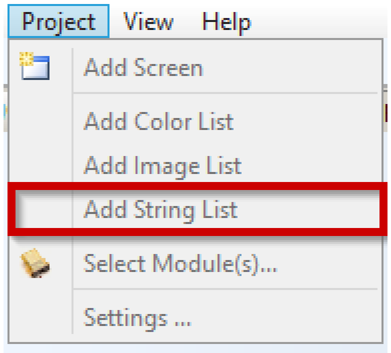
Project-Add Image List

Related:

Add Image List 297
Project Menu 22
Add Screen 22
Add Color List 23
Add String List 24
Select Module(s) 24
Settings 25

3.2.4 Add String List

Click this to add a string list to the project. The new string list is added to the String List subfolder of the Resources folder in the Explorer Pane.



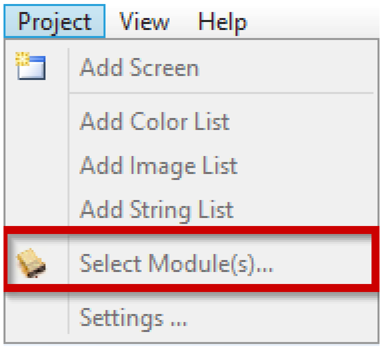
Project-Add String List

Related:

Add String List 299
Project Menu 22
Add Screen 22
Add Color List 23
Add Image List 23
Select Module(s) 24
Settings 25

3.2.5 Select Module(s)

Click this to add master and IO modules to the project, configure CAN Lines, and edit pin assignments. When clicked, this goes to the Module tab 40.



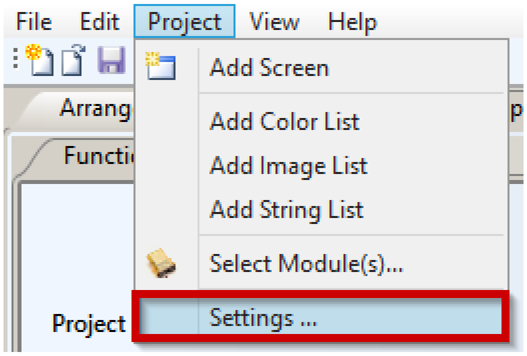
Project-Select Modules

Related:

Module Tab 40
Add Modules 272
Project Menu 22
Add Screen 22
Add Color List 23
Add Image List 23
Add String List 24
Settings 25

3.2.6 Settings

Click this to edit project settings using Orchestra® Project Settings page 21.



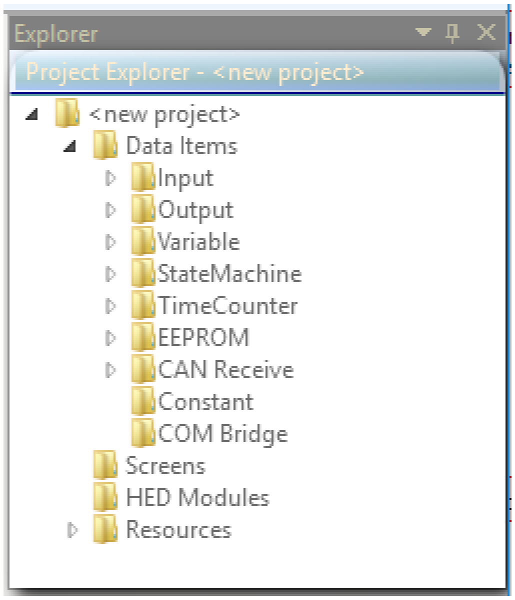
Project-Settings

Related:

Project Settings 21
Edit Project Settings 308
Project Menu 22
Add Screen 22
Add Color List 23
Add Image List 23
Add String List 24
Select Module(s) 24

3.3 Explorer Pane

The Explorer pane consists of folders and subfolders that contain everything the application will contain in an accessible manner. Within the Explorer pane, all of the folders are contained within the project folder. The subsequent subfolders are broken down into Data Items, Screens, HED® Modules, Functions, and Resources.

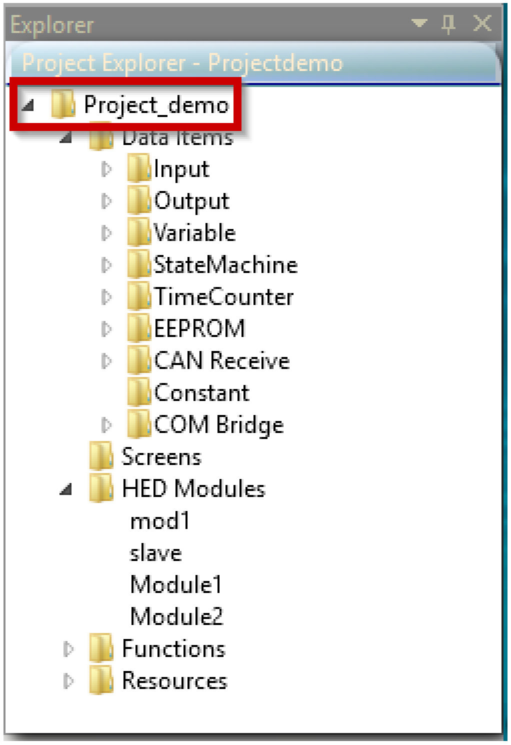


Explorer Pane

The explorer panel allows you to navigate through the elements in your project. Right-clicking an item in the explorer provides a context menu of actions that can be performed on the item.

Project Folder

The project folder is located at the top of the Explorer Pane folder list.



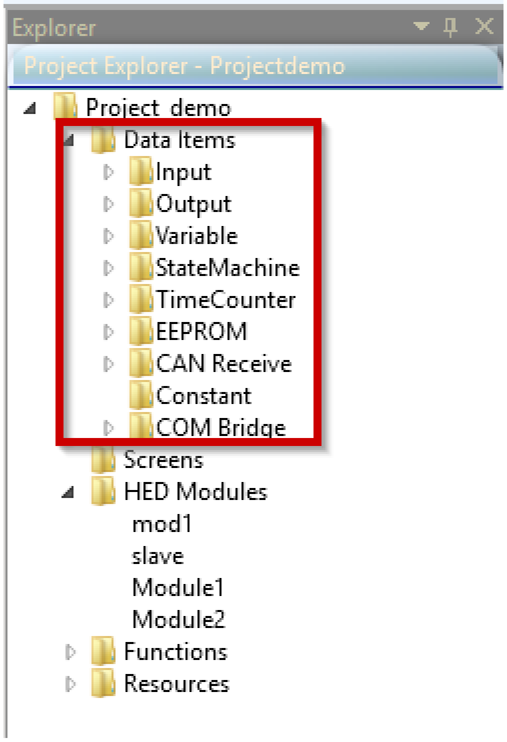
Project Folder

Related:

Data Items 28
Screens 29
HED Modules 29
Functions 30
Resources 31
Screen Layout 20

3.3.1 Data Items Folder

The Data Items folder contains all of the data item types supported by the application, separated by each type for quick access by the user.

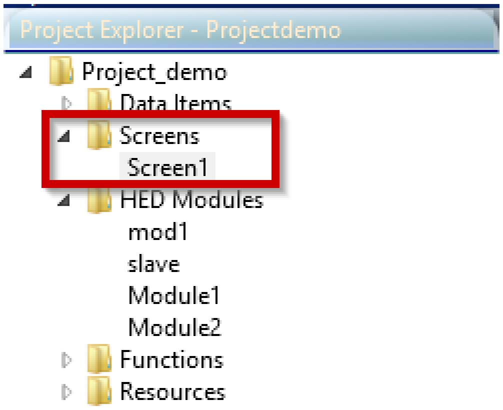


Data Items Folder

Related:
Data Items 54

3.3.2 Screens Folder

The Screen folder contains all the screens created using the Arranger™ tool.

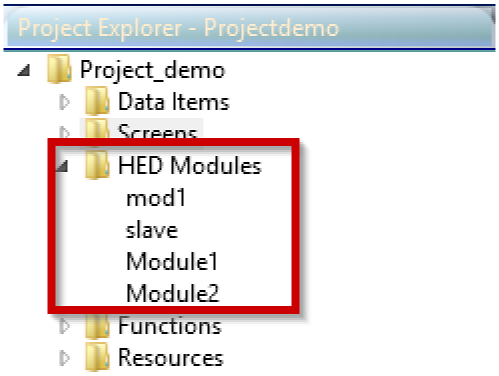


Screens Folder

Related:
Arranger-Composer 138
Add Screen 284

3.3.3 HED Modules Folder

The HED® Modules folder is where the modules selected for the project are stored for reference, editing, and mapping.

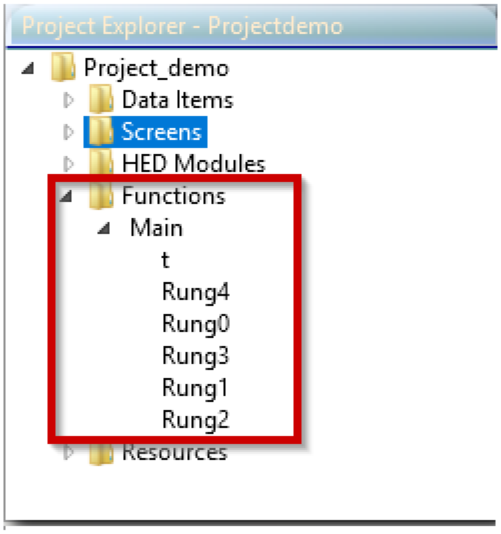


HED® Modules Folder

Related:
Explorer Pane 26
Module Tab 40

3.3.4 Functions Folder

The Functions folder is where the application rung logic is created and edited.



Project Explorer - Projectdemo

- Project_demo
  - Data Items
  - Screens
  - HED Modules
  - Functions
    - Main
      - t
      - Rung4
      - Rung0
      - Rung3
      - Rung1
      - Rung2
  - Resources

Functions Folder

Related:

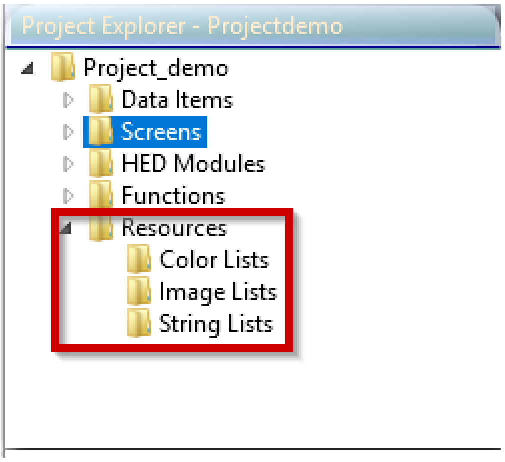
[Explorer Pane 26]

[Function: Main 107]

3.3.5 Resources Folder

The Resources folder stores all of the Image, Color, and String lists the user creates for the application.

Resource Lists allow you to define reusable sets of color, image, and string resources. For example, if you want to apply standard colors throughout all of your screens, you can define those colors once in a color list that you reference when setting color properties for screens and widgets.



Resources Folder

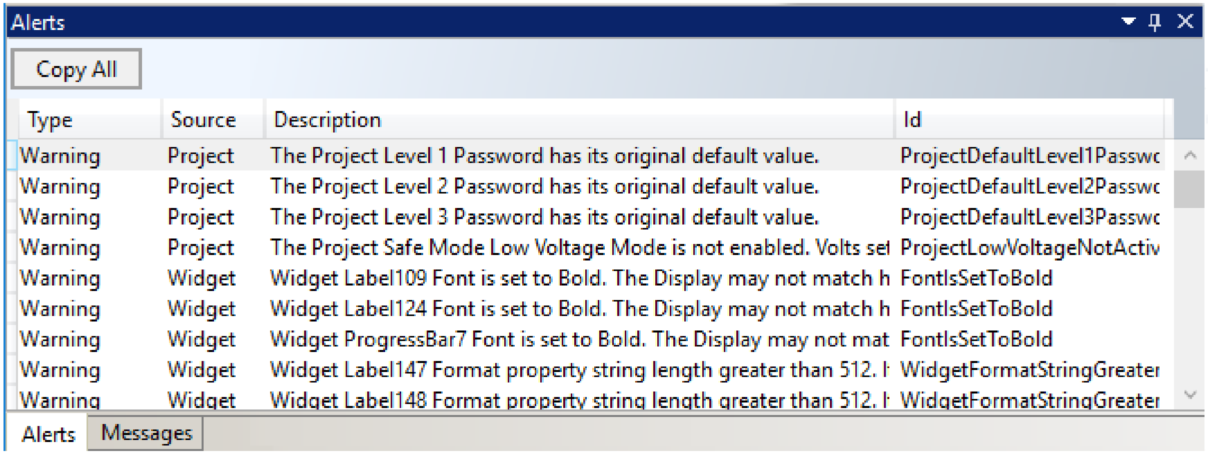
Related:

[Add Color List](292)
[Add Image List](297)
[Add String List](299)

3.4 Alerts Pane

This pane is where Orchestra® signals to the user any errors and warnings that Orchestra® finds with the current application automatically, without having to compile first.

When a new project is created, the Alerts pane shows a number of errors and warnings because the settings within the new project have not yet been set by the user. As the settings are filled in and logic is added, the warning and errors will go away.



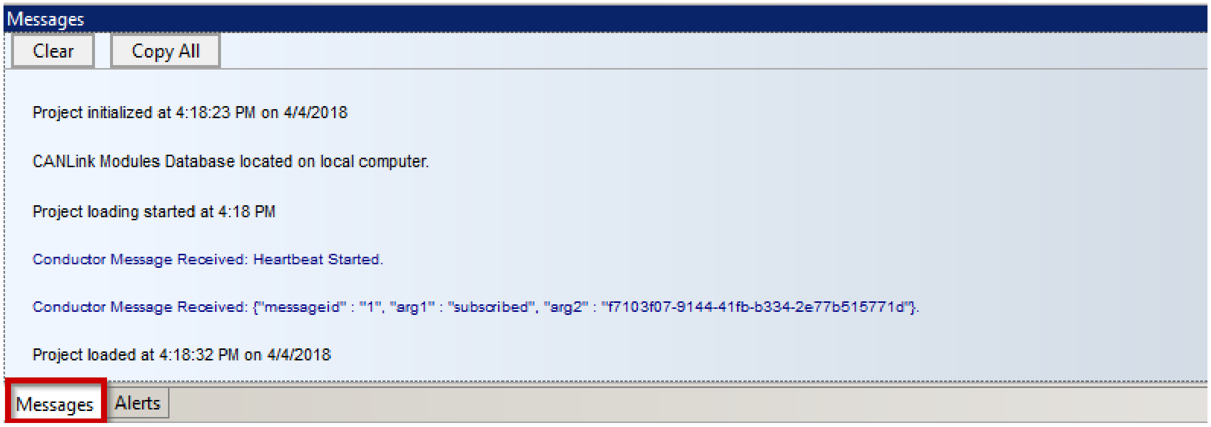
Type	Source	Description	Id
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc
Warning	Project	The Project Level 2 Password has its original default value.	ProjectDefaultLevel2Passwc
Warning	Project	The Project Level 3 Password has its original default value.	ProjectDefaultLevel3Passwc
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts set	ProjectLowVoltageNotActiv
Warning	Widget	Widget Label109 Font is set to Bold. The Display may not match h	FontIsSetToBold
Warning	Widget	Widget Label124 Font is set to Bold. The Display may not match h	FontIsSetToBold
Warning	Widget	Widget ProgressBar7 Font is set to Bold. The Display may not mat	FontIsSetToBold
Warning	Widget	Widget Label147 Format property string length greater than 512. I	WidgetFormatStringGreater
Warning	Widget	Widget Label148 Format property string length greater than 512. I	WidgetFormatStringGreater

Alerts Pane

Related:
[Messages Pane 32]

3.5 Messages Pane

The Messages pane shows updates on what the program is doing, what the user did to the project, and version information.



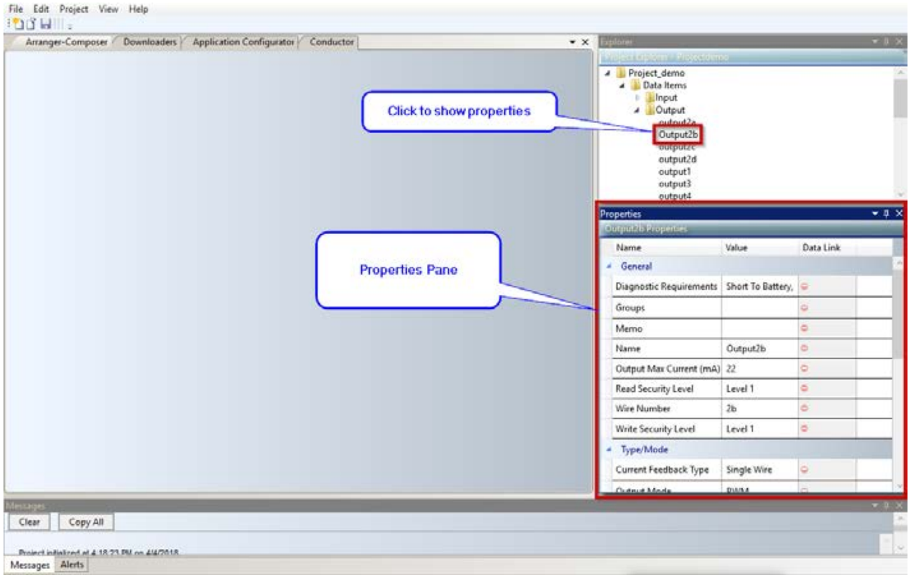
```
Project initialized at 4:18:23 PM on 4/4/2018

CANLink Modules Database located on local computer.

Project loading started at 4:18 PM

Conductor Message Received: Heartbeat Started.

Conductor Message Received: {"messageid" : "1", "arg1" : "subscribed", "arg2" : "f7103f07-9144-41fb-b334-2e77b515771d"}.

Project loaded at 4:18:32 PM on 4/4/2018
```

Messages Pane

Note: The Messages and Alerts share a pane. To switch between the two, click one of the tabs at the bottom of the pane.

3.6 Properties Pane

The Properties pane auto-populates with the available properties of whatever folder is currently highlighted within the Explorer pane. There are different property options for Data Items, Screens, Modules, Functions, and Resources. For each property, you can change its value and assign a data link to it.



Name	Value	Data Link
General		
Diagnostic Requirements	Short To Battery,	⊖
Groups		⊖
Memo		⊖
Name	Output2b	⊖
Output Max Current (mA)	22	⊖
Read Security Level	Level 1	⊖
Wire Number	2b	⊖
Write Security Level	Level 1	⊖
Type/Mode		
Current Feedback Type	Single Wire	⊖
Output Mode	PWM	⊖

Properties Pane

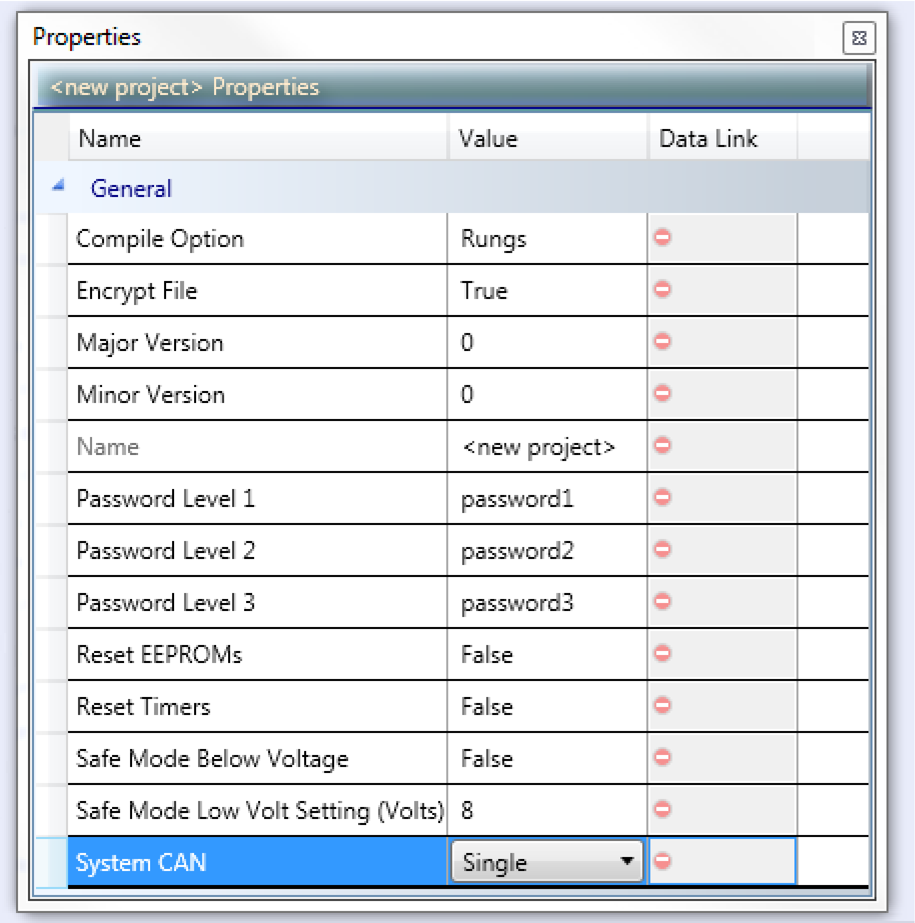
Related:

[Project Properties](34)

3.6.1 Project Properties

By highlighting the Project folder in the Explorer pane, all of the associated properties show up for the user to edit as needed. Each property can be changed by the user.

Using the Properties pane in conjunction with the Explorer pane, the user has a quick reference of all the properties associated with anything contained within the application.



Properties

<new project> Properties

Name	Value	Data Link
◢ General		
Compile Option	Rungs	⊖
Encrypt File	True	⊖
Major Version	0	⊖
Minor Version	0	⊖
Name	<new project>	⊖
Password Level 1	password1	⊖
Password Level 2	password2	⊖
Password Level 3	password3	⊖
Reset EEPROMs	False	⊖
Reset Timers	False	⊖
Safe Mode Below Voltage	False	⊖
Safe Mode Low Volt Setting (Volts)	8	⊖
System CAN	Single ▾	⊖

Properties Pane

Properties

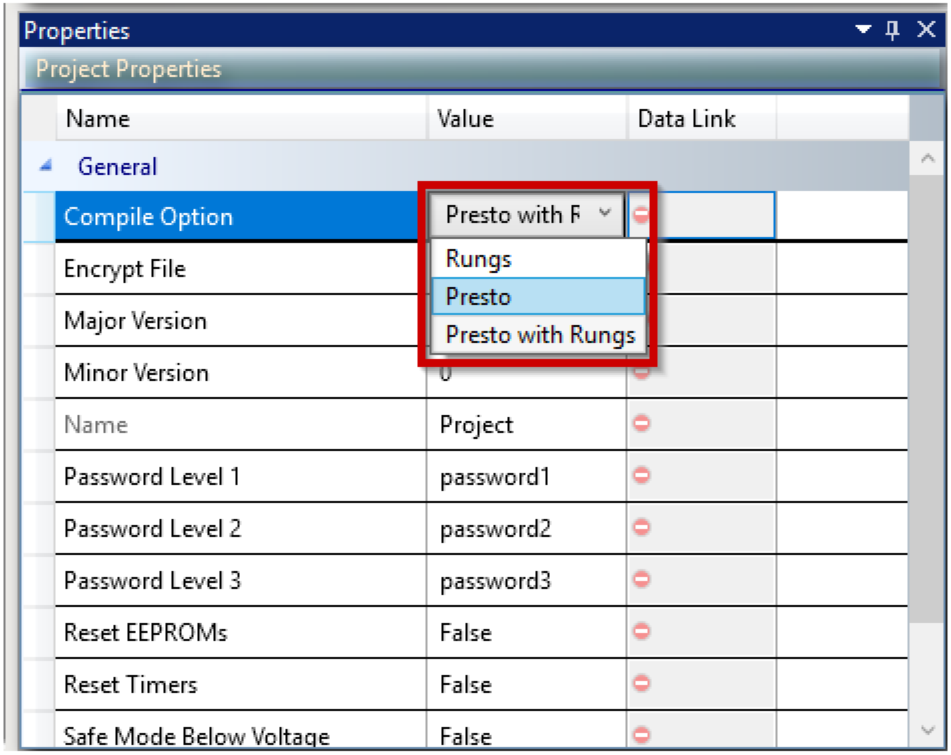
1. Compile Option 249
2. Encrypt File 251
3. Major Version 254
4. Minor Version 255
5. Name 255
6. Password Level 1 257
7. Password Level 2 257
8. Password Level 3 257
9. Reset EEPROMS 257

10. Reset Timers 257
11. Safe Mode Below Voltage 257
12. Safe Mode Low Volt Settings 257
13. System CAN 259

Related:
Compile Option 35
Properties Panel 33

3.6.1.1 Compile Option

The compile option is what determines how the program will be written and has three options; Rungs 106, Presto™, and Presto™ with Rungs. Set the project compile option in the project properties 34.

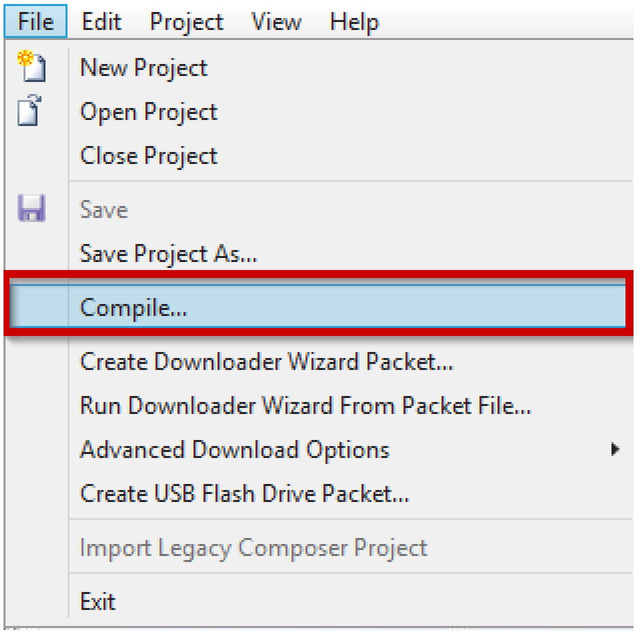


Name	Value	Data Link
Compile Option	Presto with R	
Encrypt File		
Major Version		
Minor Version	0	
Name	Project	
Password Level 1	password1	
Password Level 2	password2	
Password Level 3	password3	
Reset EEPROMs	False	
Reset Timers	False	
Safe Mode Below Voltage	False	

Project Properties: Compile Options

Rungs: Rungs is the graphic layout of the logic inside of Orchestra®
Presto™: Presto™ creates a CodeWarrior™ project that the user writes their logic in C

To activate the Compile option, the project must be configured and all errors cleared.


File
Edit
Project
View
Help
New Project
Open Project
Close Project
Save
Save Project As...
Compile...
Create Downloader Wizard Packet...
Run Downloader Wizard From Packet File...
Advanced Download Options
Create USB Flash Drive Packet...
Import Legacy Composer Project
Exit

Compile Option

Related:

Using Presto 334

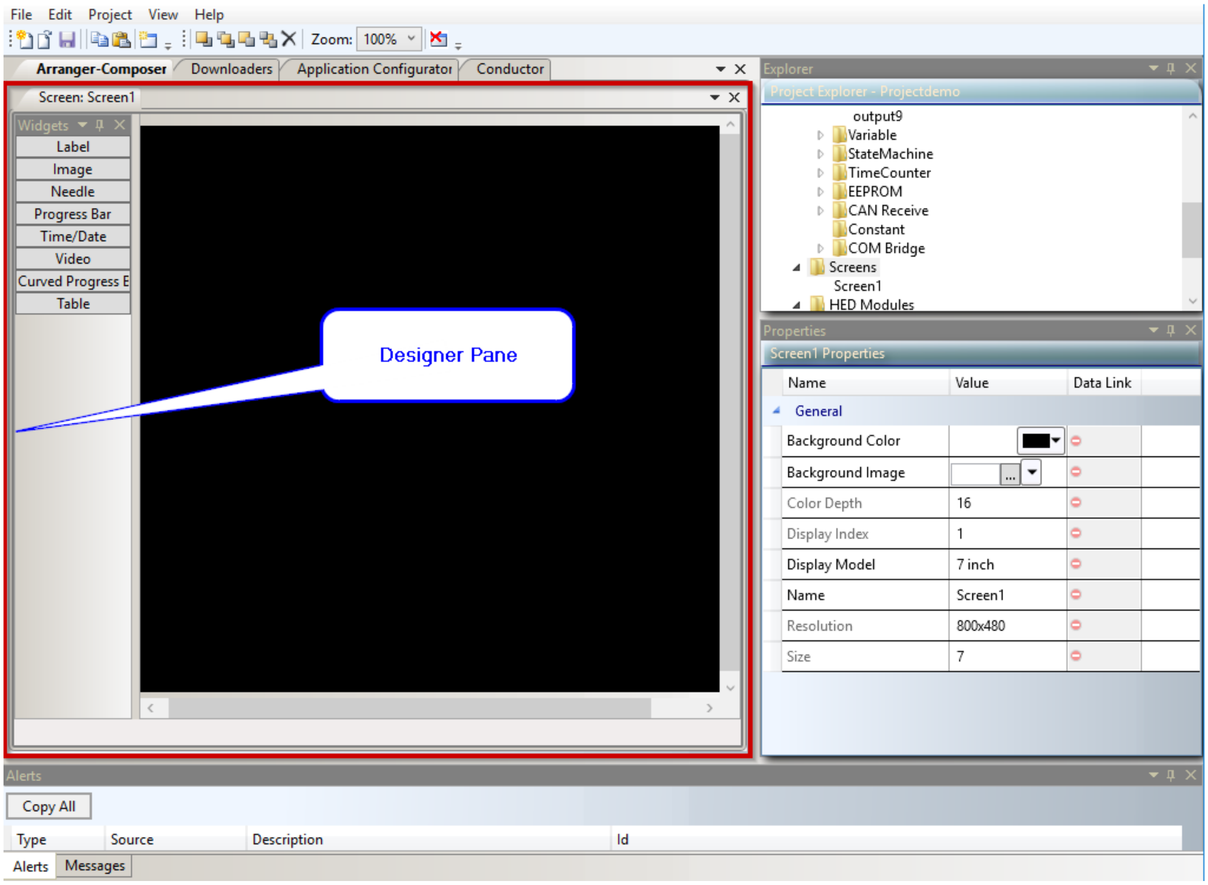
Rungs 106

Compile a Project 330

3.7 Designer Pane

The designer pane presents a variety of editors with more advanced features than are available directly in the properties panel. For example, the Screen Designer is used to lay out widgets on a screen that will be downloaded to a display module in your vehicle. More details about each of the designers is presented in later chapters.

Whenever the Screen Designer is open, a new toolbar appears with options that are available while configuring a screen.



Name	Value	Data Link
General		
Background Color		
Background Image		
Color Depth	16	
Display Index	1	
Display Model	7 inch	
Name	Screen1	
Resolution	800x480	
Size	7	

Designer Pane

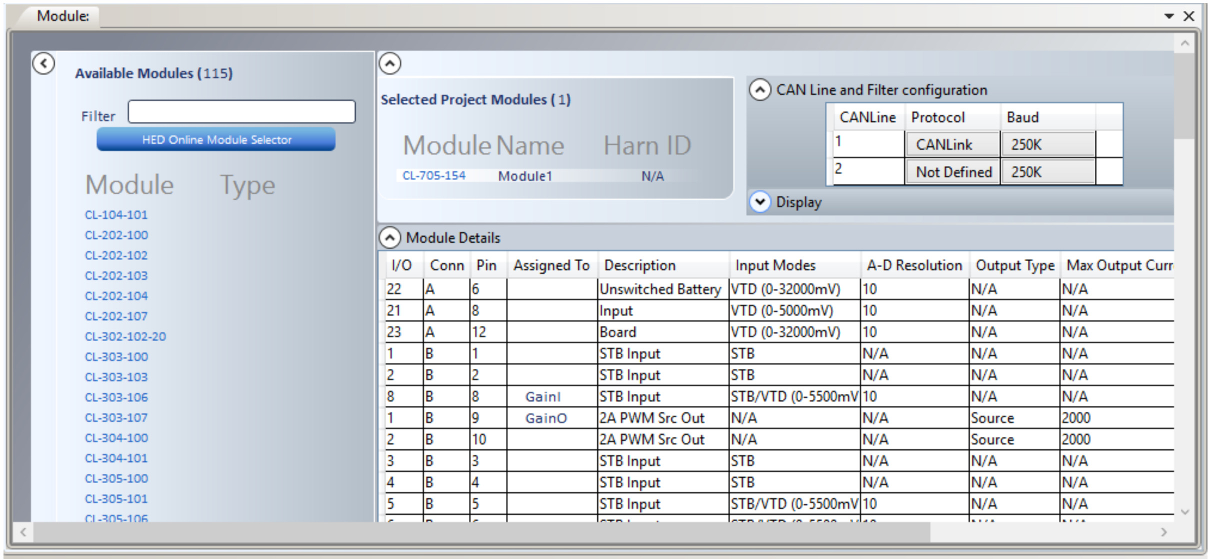
Related:

Screen Layout 20

Module Tab

4 Module Tab

The Module tab is used to add modules to the project.


Module:
Available Modules (115)
Filter
HED Online Module Selector
Selected Project Modules (1)
CAN Line and Filter configuration
Display
Module Details

Module	Type
CL-104-101	
CL-202-100	
CL-202-102	
CL-202-103	
CL-202-104	
CL-202-107	
CL-302-102-20	
CL-303-100	
CL-303-103	
CL-303-106	
CL-303-107	
CL-304-100	
CL-304-101	
CL-305-100	
CL-305-101	
CL-305-106	

Module	Name	Harn ID
CL-705-154	Module1	N/A

CANLine	Protocol	Baud
1	CANLink	250K
2	Not Defined	250K

I/O	Conn	Pin	Assigned To	Description	Input Modes	A-D Resolution	Output Type	Max Output Curr
22	A	6		Unswitched Battery	VTD (0-32000mV)	10	N/A	N/A
21	A	8		Input	VTD (0-5000mV)	10	N/A	N/A
23	A	12		Board	VTD (0-32000mV)	10	N/A	N/A
1	B	1		STB Input	STB	N/A	N/A	N/A
2	B	2		STB Input	STB	N/A	N/A	N/A
8	B	8	GainI	STB Input	STB/VTD (0-5500mV	10	N/A	N/A
1	B	9	GainO	2A PWM Src Out	N/A	N/A	Source	2000
2	B	10		2A PWM Src Out	N/A	N/A	Source	2000
3	B	3		STB Input	STB	N/A	N/A	N/A
4	B	4		STB Input	STB	N/A	N/A	N/A
5	B	5		STB Input	STB/VTD (0-5500mV	10	N/A	N/A

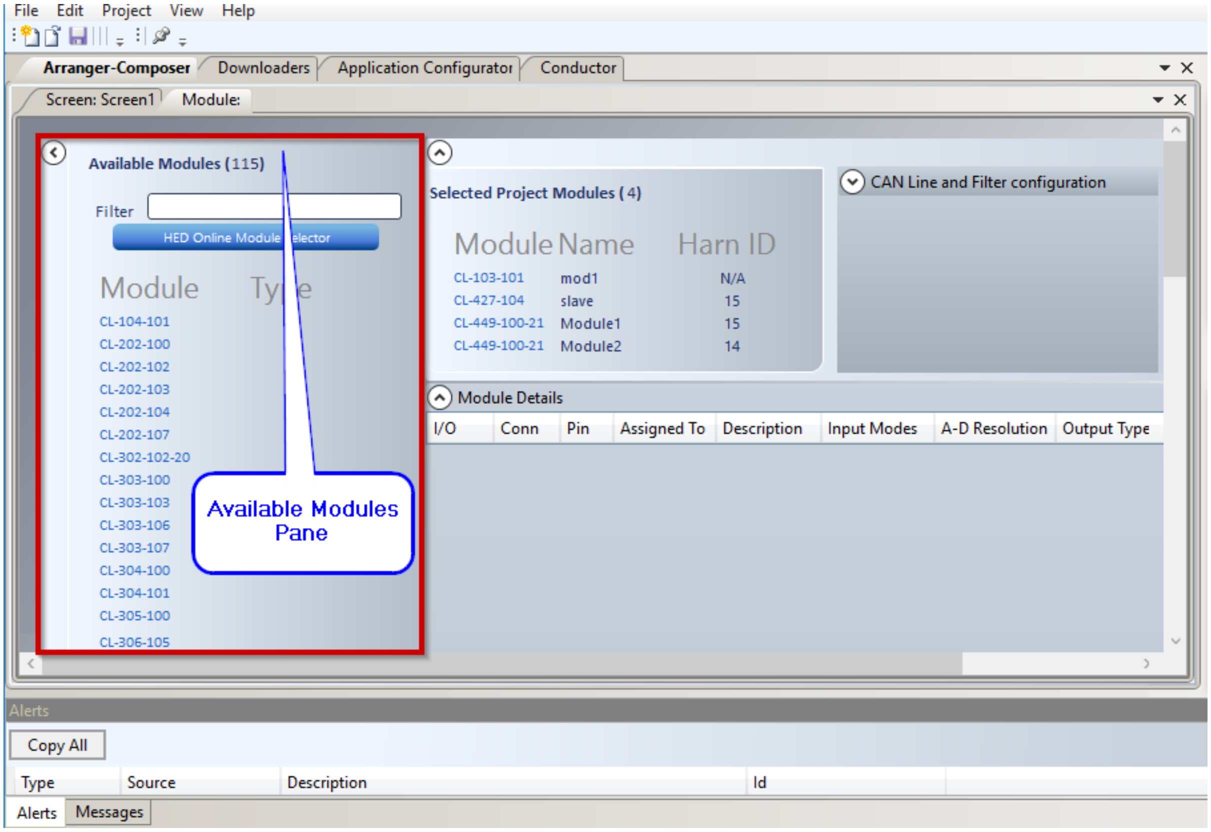
Modules Tab

Related:

Select Module(s) 24
Available Modules 41
Selected Project Modules 42
CAN Line and Filter Configuration 42
Display 46
Module Details 47
Add Modules 272

4.1 Available Modules

This pane is an interactive list of all of the modules supported within Orchestra®, as well as displaying whether or not that particular module is a master module. Once a master module is selected, then all other master modules are removed from the Available Modules list.



Module	Type
CL-104-101	
CL-202-100	
CL-202-102	
CL-202-103	
CL-202-104	
CL-202-107	
CL-302-102-20	
CL-303-100	
CL-303-103	
CL-303-106	
CL-303-107	
CL-304-100	
CL-304-101	
CL-305-100	
CL-306-105	

Module	Name	Harn ID
CL-103-101	mod1	N/A
CL-427-104	slave	15
CL-449-100-21	Module1	15
CL-449-100-21	Module2	14

I/O	Conn	Pin	Assigned To	Description	Input Modes	A-D Resolution	Output Type

Type	Source	Description	Id

Available Modules Pane

HED® Online Module Selector

This button takes the user to the HED® web page to enter in input, outputs, or other search criteria. The webpage lists the modules that fit the profile.

Module

This is an interactive list of all of the modules supported within Orchestra®.

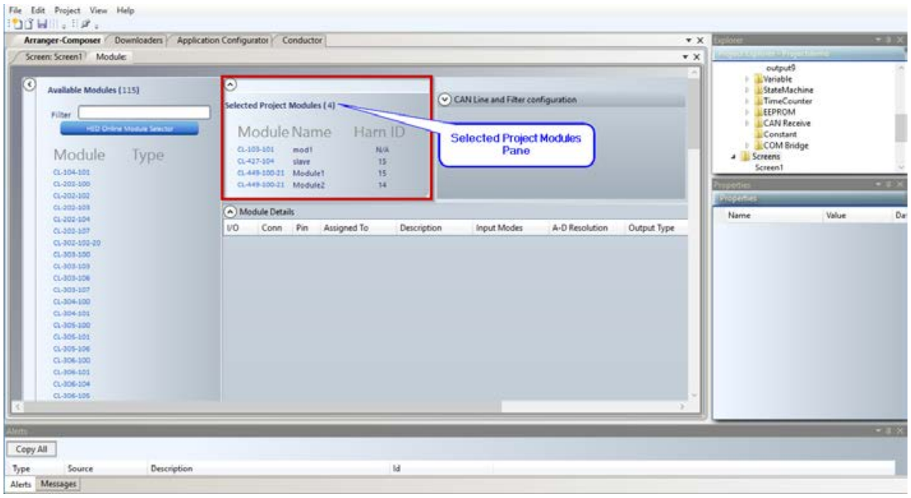
Type

This column shows the type of the module, whether Master or Display.

Related:
Selected Project Modules 42
CAN Line and Filter Configuration 42
Module Details 47

4.2 Selected Project Modules

This pane shows the modules that have been added to the project from the Module List. It shows the module's name and harness ID 263.



Selected Project Modules (4)

Module	Name	Harn ID
CL-103-101	mod1	N/A
CL-427-104	slave	15
CL-449-100-21	Module1	15
CL-449-100-21	Module2	14

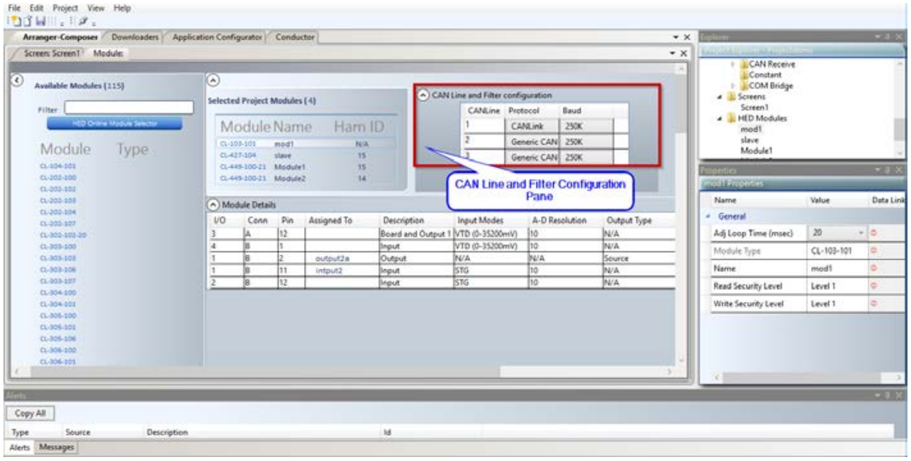
Selected Project Modules Pane

Related:

Module Tab 40
Available Modules 41
CAN Line and Filter Configuration 42
Display 46
Module Details 47

4.3 CAN Line and Filter Configuration

This pane is used to set the CAN Line protocol and baud rate. The pane has different CAN line and filter configuration options for master and client modules.



CAN Line and Filter configuration

CANLine	Protocol	Baud
1	CANLink	250K
2	Generic CAN	250K
3	Generic CAN	250K

Module Details

I/O	Conn	Pin	Assigned To	Description	Input Modes	A-D Resolution	Output Type
3	A	12		Board and Output 1	VTD (0-35200mV)	10	N/A
4	B	1		Input	VTD (0-35200mV)	10	N/A
1	B	2	output2a	Output	N/A	N/A	Source
1	B	11	intput2	Input	STG	10	N/A
2	B	12		Input	STG	10	N/A

CAN Line and Filter Configuration Pane

Related:
Module Tab 40

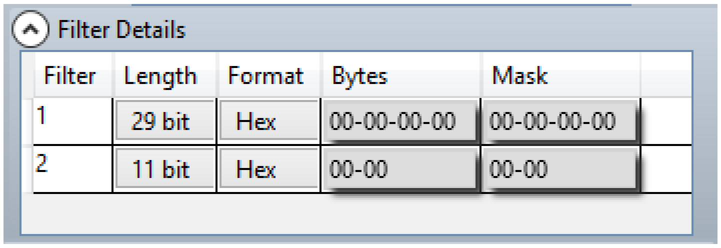
4.3.1 Filter Details

The Filter Details pane configures the hardware filter on the CAN controller of the master module. This hardware filter will block messages from making it to the software. If there is a lot of traffic on a CAN bus, this is used to improve the master module's operation by not getting interrupted on messages that it doesn’t need.

The filter is set to pass all messages by default.

*Note: If the filter is set up to **not** accept the 00EF0090 message, then the module will not be able to accept CANLink® utility messages over CAN.*

Filter Details is activated by setting the master module's CANLine protocol to Generic CAN.



Filter Details

Filter	Length	Format	Bytes	Mask
1	29 bit	Hex	00-00-00-00	00-00-00-00
2	11 bit	Hex	00-00	00-00

Module Filter Details

Length - sets the identifier type. Choose 11 or 29 bit.
Format - choose Hex or Dec.
Bytes - set what the incoming CAN message identifier needs to be.
Mask - These are the bits in the incoming CAN message identifier that need to match the Bytes field. 0 = don’t check bit, 1 = incoming bit must match bit in Bytes field.

Related:
CAN Line and Filter Configuration 42

4.4 Harness ID Selection

The software configurable stores the harness code in persistent memory and there is a CAN message to change it.

The module ID and harness code combination needs to be unique.

The number of harness codes varies per module.

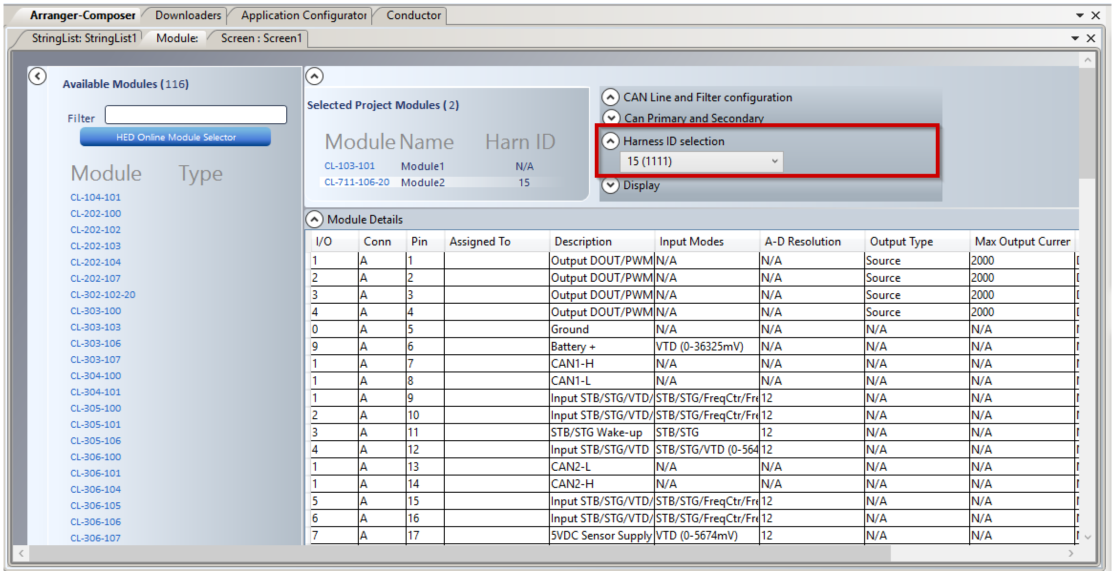
4 Harness = 0-15
3 Harness = 8-15
2 Harness = 12-15
1 Harness = 14-15
0 Harness = 15 or could be software configurable, so then it is 0-15

Format Example

Module A -> ID = 0x0123 & Harness = 15
Module B -> ID = 0x0123 & Harness = 14

or

Module A -> ID = 0x0123 & Harness = 15
Module B -> ID = 0x0124 & Harness = 15



Available Modules (116)

Module	Type
CL-104-101	
CL-202-100	
CL-202-102	
CL-202-103	
CL-202-104	
CL-202-107	
CL-302-102-20	
CL-303-100	
CL-303-103	
CL-303-106	
CL-303-107	
CL-304-100	
CL-304-101	
CL-305-100	
CL-305-101	
CL-305-106	
CL-306-100	
CL-306-101	
CL-306-104	
CL-306-105	
CL-306-106	
CL-306-107	

Selected Project Modules (2)

Module	Name	Harn ID
CL-103-101	Module1	N/A
CL-711-106-20	Module2	15

Harness ID selection: 15 (1111)

Module Details

I/O	Conn	Pin	Assigned To	Description	Input Modes	A-D Resolution	Output Type	Max Output Currer
1	A	1		Output DOUT/PWM	N/A	N/A	Source	2000
2	A	2		Output DOUT/PWM	N/A	N/A	Source	2000
3	A	3		Output DOUT/PWM	N/A	N/A	Source	2000
4	A	4		Output DOUT/PWM	N/A	N/A	Source	2000
0	A	5		Ground	N/A	N/A	N/A	N/A
9	A	6		Battery +	VTD (0-36325mV)	N/A	N/A	N/A
1	A	7		CAN1-H	N/A	N/A	N/A	N/A
1	A	8		CAN1-L	N/A	N/A	N/A	N/A
1	A	9		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A
2	A	10		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A
3	A	11		STB/STG Wake-up	STB/STG	12	N/A	N/A
4	A	12		Input STB/STG/VTD	STB/STG/VTD (0-564	12	N/A	N/A
1	A	13		CAN2-L	N/A	N/A	N/A	N/A
1	A	14		CAN2-H	N/A	N/A	N/A	N/A
5	A	15		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A
6	A	16		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A
7	A	17		5VDC Sensor Supply	VTD (0-5674mV)	12	N/A	N/A

Harness ID Selection Pane

Related:

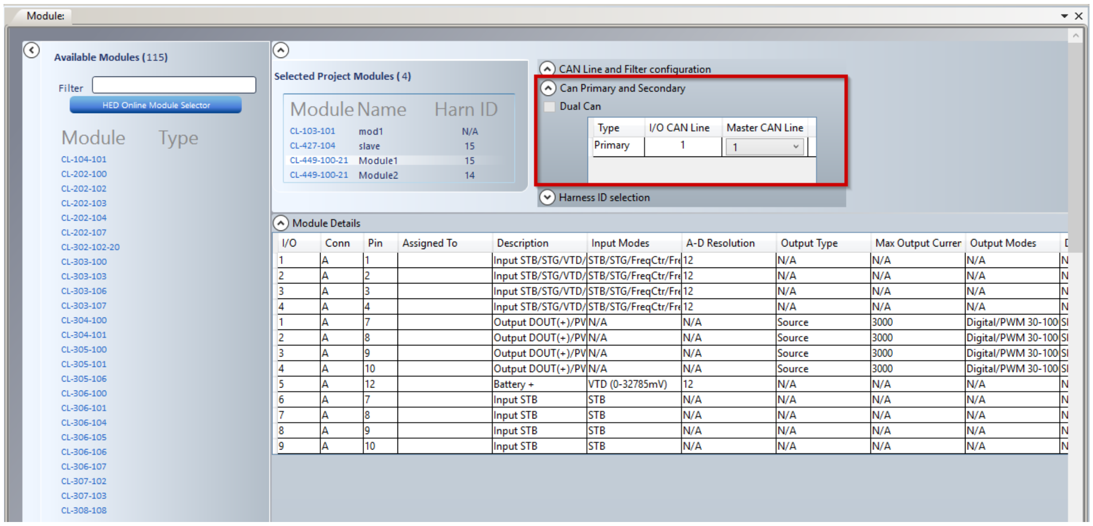
Harness ID Selection 283

4.5 CAN Primary and Secondary

This pane is used to configure which CAN line the client is connected to the master module on.

For example, the master module has 3 CAN lines and the client only has 1. The client could be connected to any of the 3 on the master, but it is up to the user how they configure it.

The clients can use a redundant CAN line, so if a line is broken they will continue to operate. The client and the master would both need 2 CAN lines for that to work, and the client needs to be programmed to support it.


Module:
Available Modules (115)
Filter
HED Online Module Selector
Selected Project Modules (4)
CAN Line and Filter configuration
Can Primary and Secondary
Harness ID selection
Module Details

Module	Type
CL-104-101	
CL-202-100	
CL-202-102	
CL-202-103	
CL-202-104	
CL-202-107	
CL-302-102-20	
CL-303-100	
CL-303-103	
CL-303-106	
CL-303-107	
CL-304-100	
CL-304-101	
CL-305-100	
CL-305-101	
CL-305-106	
CL-306-100	
CL-306-101	
CL-306-104	
CL-306-105	
CL-306-106	
CL-306-107	
CL-307-102	
CL-307-103	
CL-308-108	

Module	Name	Harn ID
CL-103-101	mod1	N/A
CL-427-104	slave	15
CL-449-100-21	Module1	15
CL-449-100-21	Module2	14

☐ Dual Can

Type	I/O CAN Line	Master CAN Line
Primary	1	1

I/O	Conn	Pin	Assigned To	Description	Input Modes	A-D Resolution	Output Type	Max Output Currer	Output Modes	D
1	A	1		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A	N/A	N
2	A	2		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A	N/A	N
3	A	3		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A	N/A	N
4	A	4		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A	N/A	N/A	N
1	A	7		Output DOUT(+)/PV	N/A	N/A	Source	3000	Digital/PWM 30-100	SI
2	A	8		Output DOUT(+)/PV	N/A	N/A	Source	3000	Digital/PWM 30-100	SI
3	A	9		Output DOUT(+)/PV	N/A	N/A	Source	3000	Digital/PWM 30-100	SI
4	A	10		Output DOUT(+)/PV	N/A	N/A	Source	3000	Digital/PWM 30-100	SI
5	A	12		Battery +	VTD (0-32785mV)	12	N/A	N/A	N/A	N
6	A	7		Input STB	STB	N/A	N/A	N/A	N/A	N
7	A	8		Input STB	STB	N/A	N/A	N/A	N/A	N
8	A	9		Input STB	STB	N/A	N/A	N/A	N/A	N
9	A	10		Input STB	STB	N/A	N/A	N/A	N/A	N

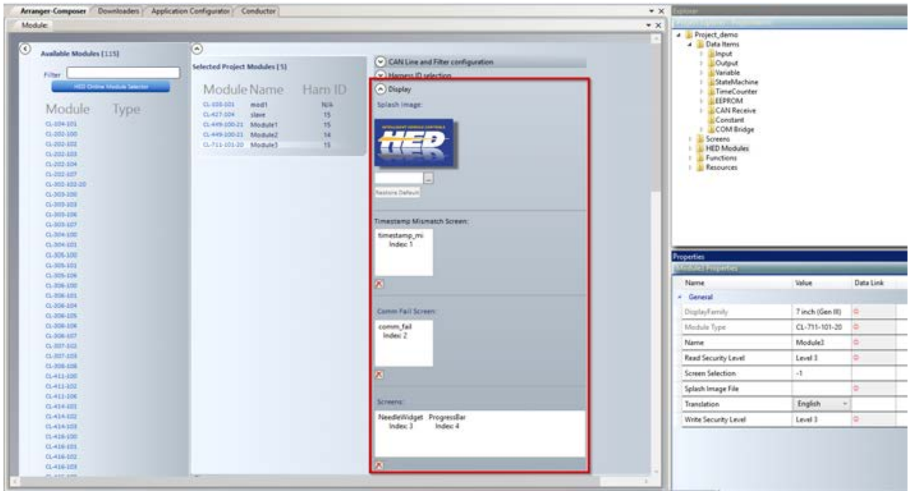
CAN Primary and Secondary Pane

Related:

CAN Line and Filter Configuration 42

4.6 Display

The Display pane is used to assign the created screens to the display module.



Name	Value	Data Link
DisplayFamily	7 inch (Gen III)	
Module Type	CL-711-101-20	
Name	Module3	
Read Security Level	Level 1	
Screen Selection	-1	
Splash Image File		
Translation	English	
Write Security Level	Level 1	

Display Pane

Splash Image

This is the image that will be shown on your display after it has powered on, and your specific vehicle application is being loaded. To select a custom image file, click on the browse button containing ellipses. To restore the default image supplied with Orchestra® Suite, click the 'Restore Default' button

Timestamp Mismatch Screen

This is the screen that will be shown on your display if the application you downloaded to it does not match the application in your master module. There can be only one screen specified, and it must not contain any widgets with linked properties.

Comm Fail Screen

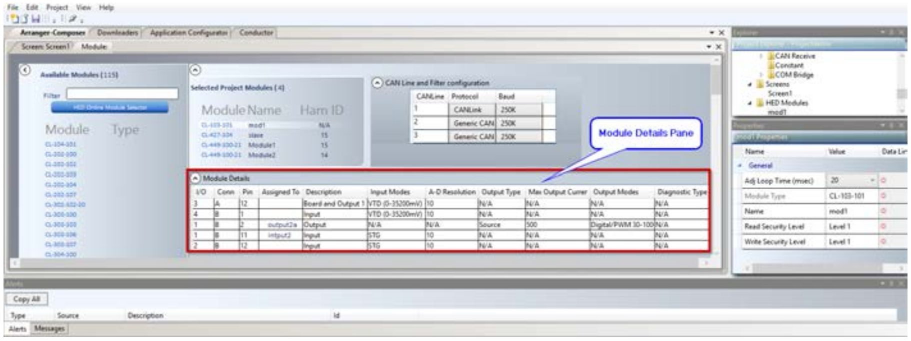
This is the screen that will be shown on your display if there is a communications failure between the display and master module. There can be only one screen specified, and it must not contain any widgets with linked properties.

Screens

This is the set of screens that will be shown on your display. The Composer™ application should contain a data item whose value has been programmed through rung logic to be equal to one of the Screen ID values displayed below each screen in this list. (Presto™ can also be used to manipulate the Data Item value from C code.)

4.7 Module Details

Once a module is selected, the Module Details section fills with all of the detailed pin assignments.



Selected Project Modules (4)

Module	Name	Harn ID
CL-103-101	mod1	N/A
CL-427-104	slave	15
CL-449-100-21	Module1	15
CL-449-100-21	Module2	14

CAN Line and Filter configuration

CANLine	Protocol	Baud
1	CANLink	250K
2	Generic CAN	250K
3	Generic CAN	250K

Module Details

I/O	Conn	Pin	Assigned To	Description	Input Modes	A-D Resolution	Output Type	Max Output Currer	Output Modes	Diagnostic Type
3	A	12		Board and Output 1	VTD (0-35200mV)	10	N/A	N/A	N/A	N/A
4	B	1		Input	VTD (0-35200mV)	10	N/A	N/A	N/A	N/A
1	B	2	output2a	Output	N/A	N/A	Source	500	Digital/PWM 30-100	N/A
1	B	11	intput2	Input	STG	10	N/A	N/A	N/A	N/A
2	B	12		Input	STG	10	N/A	N/A	N/A	N/A

Name	Value	Data Lir
General		
Adj Loop Time (msec)	20	
Module Type	CL-103-101	
Name	mod1	
Read Security Level	Level 1	
Write Security Level	Level 1	

Type	Source	Description	Id

Module Details Pane

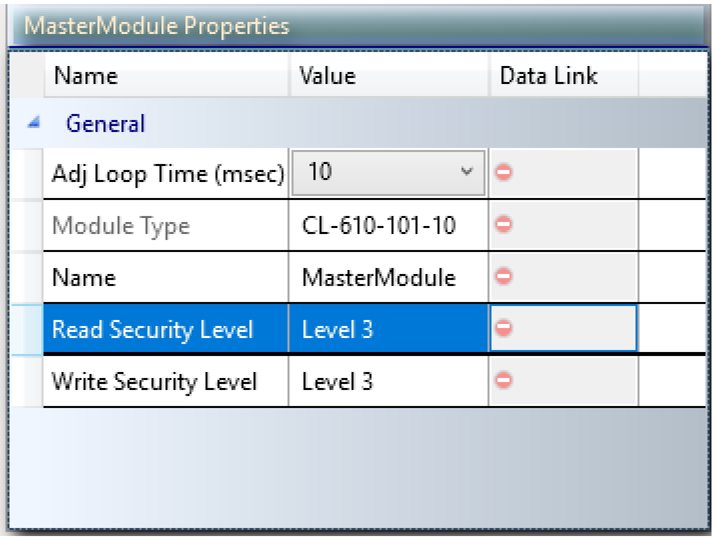
Related:

Data Items 54
Module Properties 48

4.8 Module Properties

The module properties vary depending on the type of module. There are four types of modules: master, display, master-display, and I/O.

Master Module



MasterModule Properties

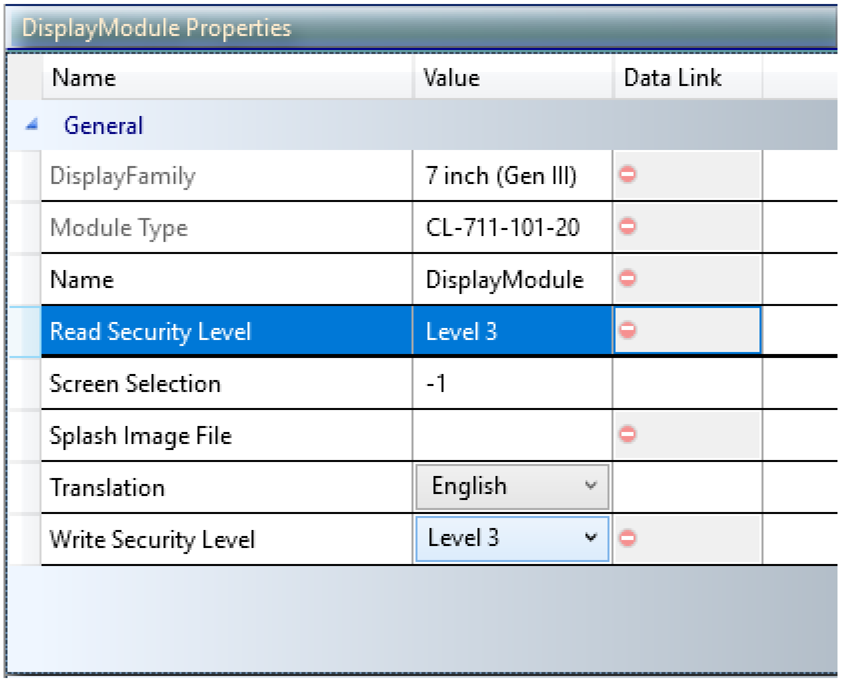
Name	Value	Data Link
General		
Adj Loop Time (msec)	10	⊖
Module Type	CL-610-101-10	⊖
Name	MasterModule	⊖
Read Security Level	Level 3	⊖
Write Security Level	Level 3	⊖

Master Module Properties

Properties

1. Adj Loop Time (msec) 248
2. Module Type 255
3. Name 255
4. Read Security Level 257
5. Write Security Level 260

Display Module



DisplayModule Properties

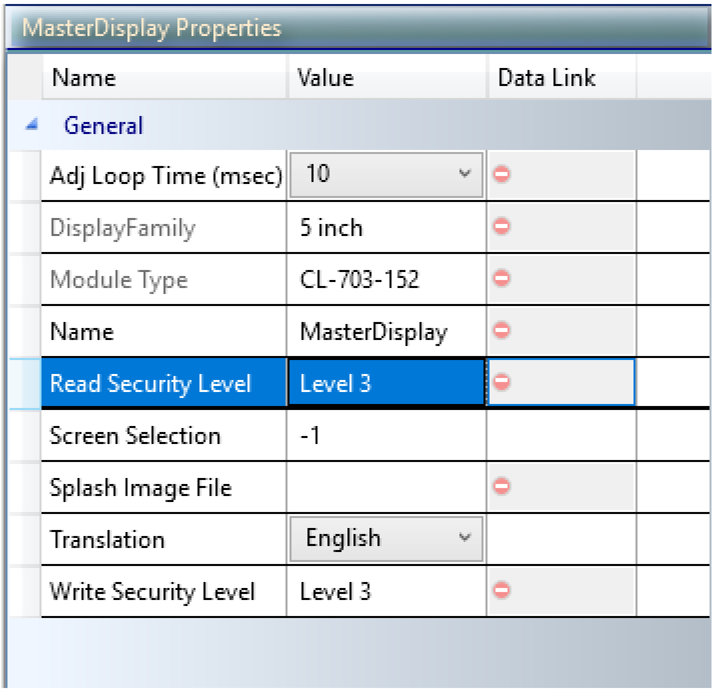
Name	Value	Data Link
◢ General		
DisplayFamily	7 inch (Gen III)	⊖
Module Type	CL-711-101-20	⊖
Name	DisplayModule	⊖
Read Security Level	Level 3	⊖
Screen Selection	-1	
Splash Image File		⊖
Translation	English	
Write Security Level	Level 3	⊖

Display Module Properties

Properties

1. Display Family 251
2. Module Type 255
3. Name 255
4. Read Security Level 257
5. Screen Selection 258
6. Splash Image File 258
7. Translation 260
8. Write Security Level 260

Master-Display Module



MasterDisplay Properties

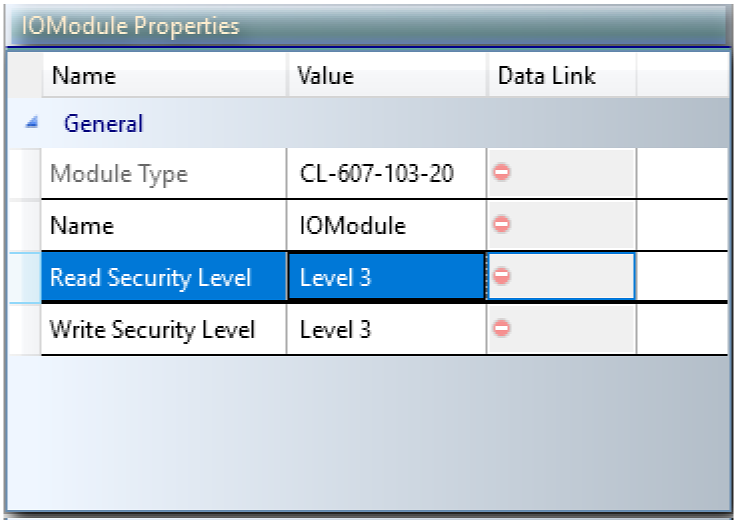
Name	Value	Data Link
General		
Adj Loop Time (msec)	10	⊖
DisplayFamily	5 inch	⊖
Module Type	CL-703-152	⊖
Name	MasterDisplay	⊖
Read Security Level	Level 3	⊖
Screen Selection	-1	
Splash Image File		⊖
Translation	English	
Write Security Level	Level 3	⊖

Master-Display Module Properties

Properties

1. Adj Loop Time (msec 248)
2. Display Family 251
3. Module Type 255
4. Name 255
5. Read Security Level 257
6. Screen Selection 258
7. Splash Image File 248
8. Translation 248
9. Write Security Level 260

I/O Module



IOModule Properties

Name	Value	Data Link
General		
Module Type	CL-607-103-20	⊖
Name	IOModule	⊖
Read Security Level	Level 3	⊖
Write Security Level	Level 3	⊖

I/O Module Properties

Properties

1. Module Type 255
2. Name 255
3. Read Security Level 257
4. Write Security Level 260

Related:

Properties Pane 33

Data Items

5 Data Items

Data items is the generic term for inputs, outputs, variables, EEPROMS, timers, state machines, CAN receives, constants, and COM bridges.

Related:

Data Items Folder 28
Input 54
Output 76
Variable 90
COM Bridge 96
Constant 98
CAN Receive 99
EEPROM 101
Time Counter 103

5.1 Input

The Input data item holds the incoming information or value from the pin of the assigned.

The Input data item is the input on a module. These are read-only values.

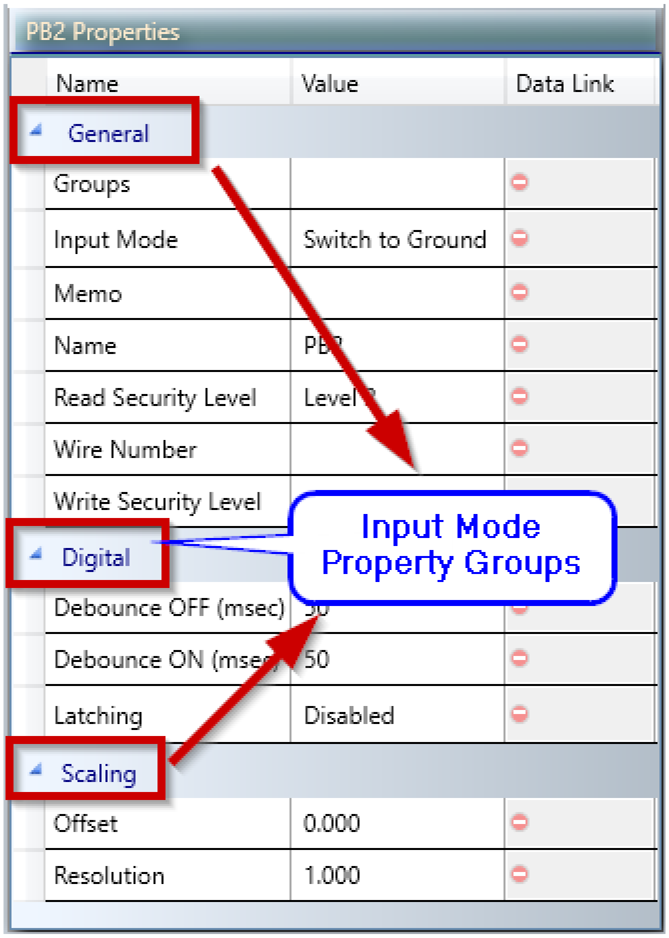
Related:

Input Mode Properties 55
Digital Input Modes 57
Analog Input Modes 62

5.1.1 Input Mode Properties

There are three groups of properties:

1. General
2. Digital/Analog
3. Scaling



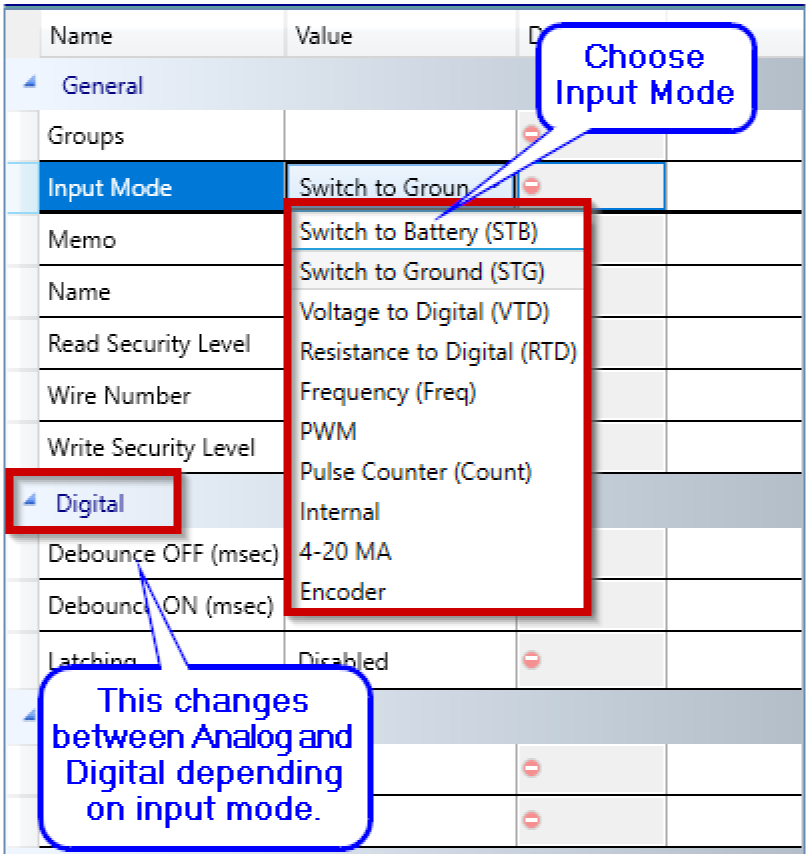
PB2 Properties

Name	Value	Data Link
General		
Groups		⊖
Input Mode	Switch to Ground	⊖
Memo		⊖
Name	PB2	⊖
Read Security Level	Level [illegible]	⊖
Wire Number		⊖
Write Security Level		
Digital		
Debounce OFF (msec)	50	⊖
Debounce ON (msec)	50	⊖
Latching	Disabled	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Input Mode Property Groups

Define the input mode by choosing an option from the Input Mode drop-down menu.

Depending on the input mode, the second property group will change to Digital or Analog. In doing so, the properties contained under that group will change.



Choose Input Mode

This changes between Analog and Digital depending on input mode.

Input Mode Changes Analog/Digital

Note: Properties within all data items are linked at compile time and cannot be changed in real time as the code is running. Any changes made will not be seen until the code is recompiled.

Related:

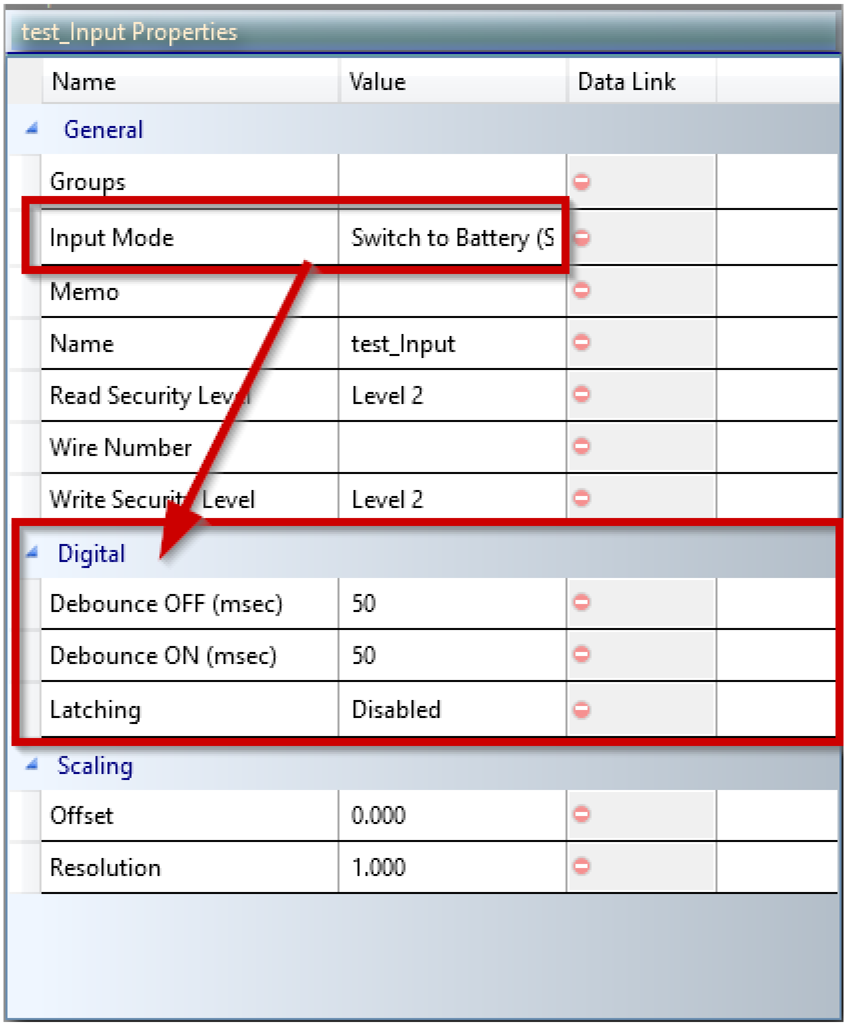
Digital Input Modes 57

Analog Input Modes 62

5.1.2 Digital Input Modes

The Digital input modes are Switch to Battery and Switch to Ground.

Depending on the input mode, the second property group will change to Digital or Analog 62. In doing so, the properties contained under that group will change.



test_Input Properties

Name	Value	Data Link
General		
Groups		⊖
Input Mode	Switch to Battery (S	⊖
Memo		⊖
Name	test_Input	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Digital		
Debounce OFF (msec)	50	⊖
Debounce ON (msec)	50	⊖
Latching	Disabled	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

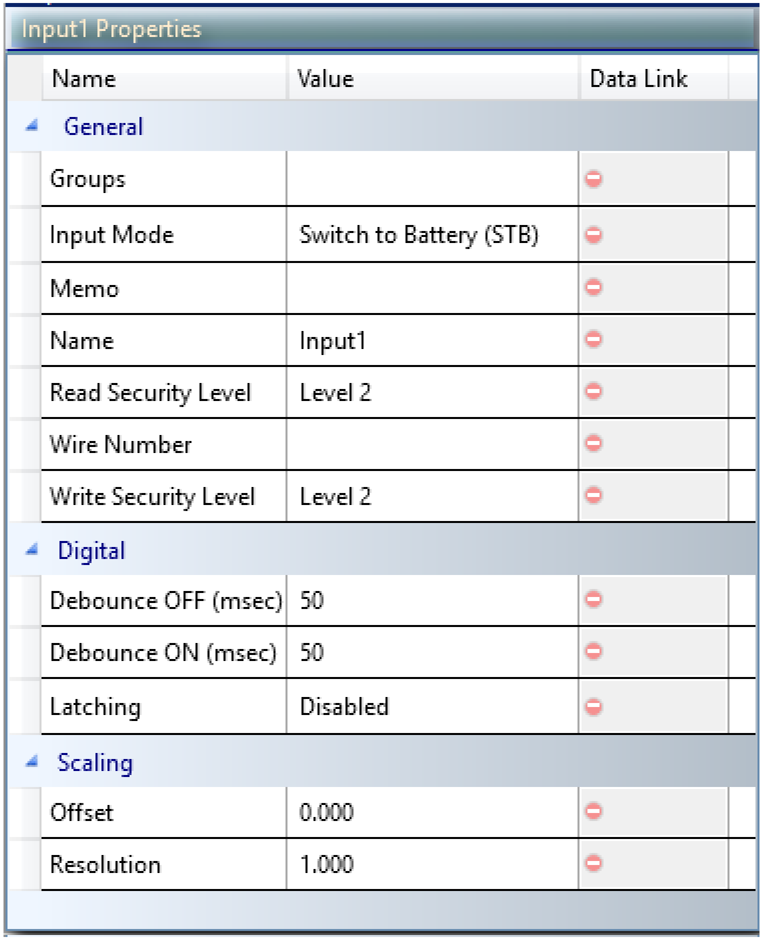
Digital Input Mode Property Group

Related:

Input Mode Properties 55
Switch to Battery (STB) 58
Switch to Ground (STG) 60

5.1.2.1 Switch to Battery (STB)

This is one of two digital modes that the Input can be classified as by adjusting the Input Mode property. In this mode the input is inactive (OFF) when voltage on the pin is below 3.5V, and active (ON) the voltage rises above 7V.



Input1 Properties

Name	Value	Data Link
General		
Groups		⊖
Input Mode	Switch to Battery (STB)	⊖
Memo		⊖
Name	Input1	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Digital		
Debounce OFF (msec)	50	⊖
Debounce ON (msec)	50	⊖
Latching	Disabled	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Switch to Battery Input Mode

Properties

1. Input Mode 254
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Wire Number 260
6. Write Security Level 260
7. Debounce OFF (msec) 250
8. Debounce ON (msec) 250
9. Latching 254
10. Offset 256
11. Resolution 257

Related:

Unknown State 59

Switch to Ground (STG) 60

5.1.2.1.1 Unknown State

There is also a third “unknown” state in which the input is considered neither active nor open. This unknown state is seen when the module first initializes, or if the module is missing. When the module first powers up, the input will stay in the unknown state for however long the debounce time within the properties is set to.

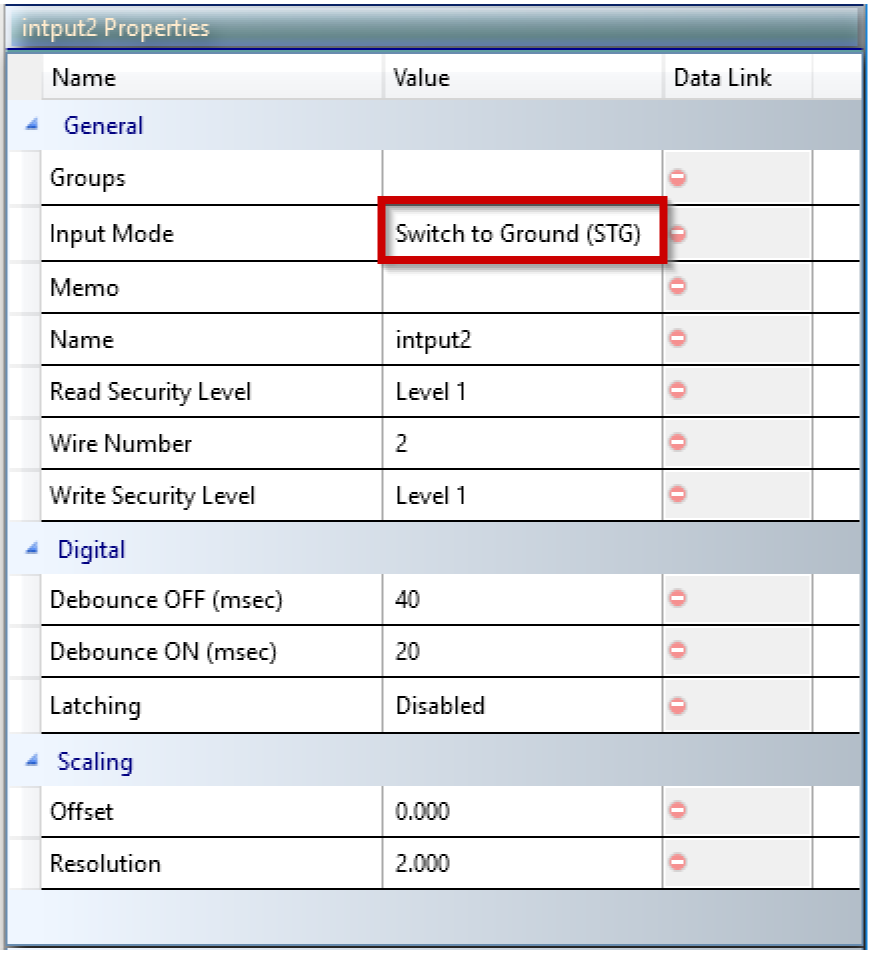
Related:
Switch to Battery (STB) 58

5.1.2.2 Switch to Ground (STG)

STG is the second of the digital modes of the Input data items. This mode has the input inactive when voltage on the pin is above 3.5V and active when the voltage drops below 1.5V. STG has the same digital properties as STB which act the same way.

Properties

1. Groups 253
2. Input Mode 254
3. Memo 255
4. Name 255
5. Read Security Level 257
6. Wire Number 260
7. Write Security Level 260
8. Debounce OFF (msec) 250
9. Debounce ON (msec) 250
10. Latching 254
11. Offset 256
12. Resolution 257



intput2 Properties

Name	Value	Data Link
General		
Groups		⊖
Input Mode	Switch to Ground (STG)	⊖
Memo		⊖
Name	intput2	⊖
Read Security Level	Level 1	⊖
Wire Number	2	⊖
Write Security Level	Level 1	⊖
Digital		
Debounce OFF (msec)	40	⊖
Debounce ON (msec)	20	⊖
Latching	Disabled	⊖
Scaling		
Offset	0.000	⊖
Resolution	2.000	⊖

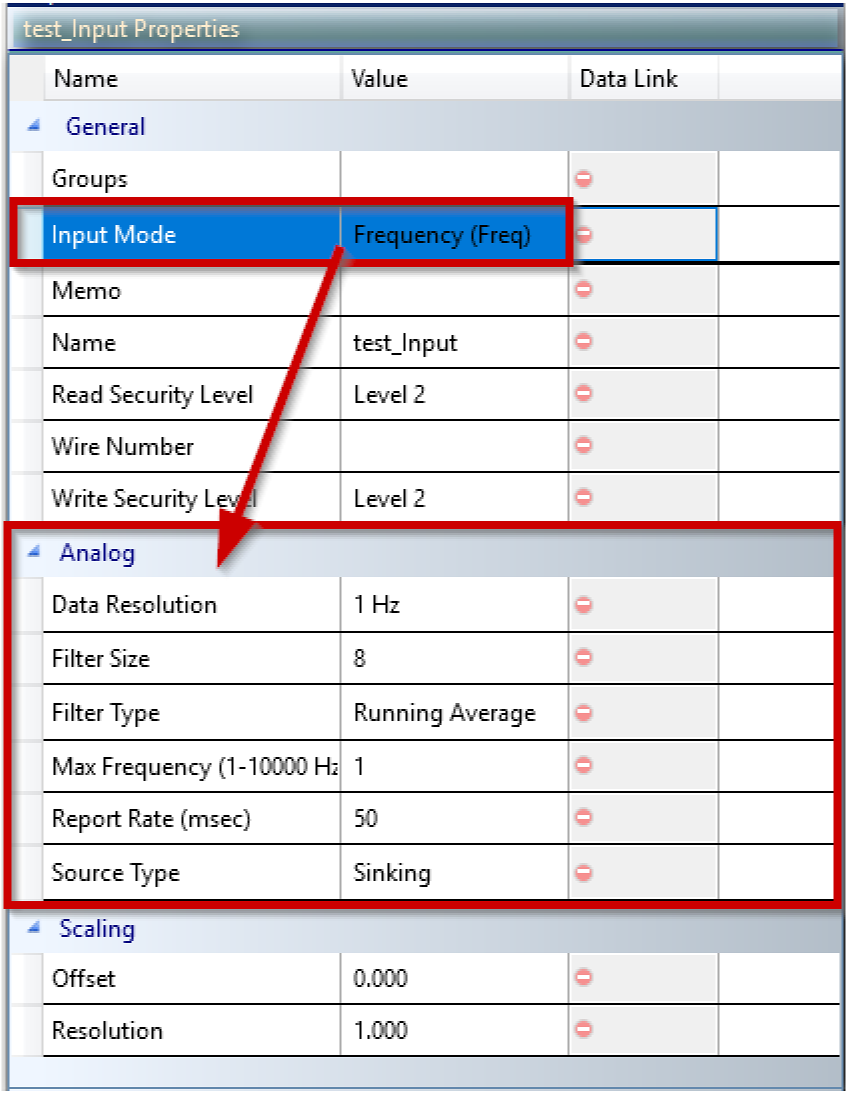
STB/STG Input Properties

Related:
Switch to Battery (STB) 58

5.1.3 Analog Input Modes

The Analog input modes are Voltage to Digital, Resistance to Digital, Frequency, Pulse Width Modulation, Pulse Counter, Internal, Encoder, and 4-20 (mA).

Depending on the input mode, the second property group will change to Digital 57 or Analog. In doing so, the properties contained under that group will change.



test_Input Properties

Name	Value	Data Link
General		
Groups		⊖
Input Mode	Frequency (Freq)	⊖
Memo		⊖
Name	test_Input	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Analog		
Data Resolution	1 Hz	⊖
Filter Size	8	⊖
Filter Type	Running Average	⊖
Max Frequency (1-10000 Hz	1	⊖
Report Rate (msec)	50	⊖
Source Type	Sinking	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Analog Input Mode Property Group

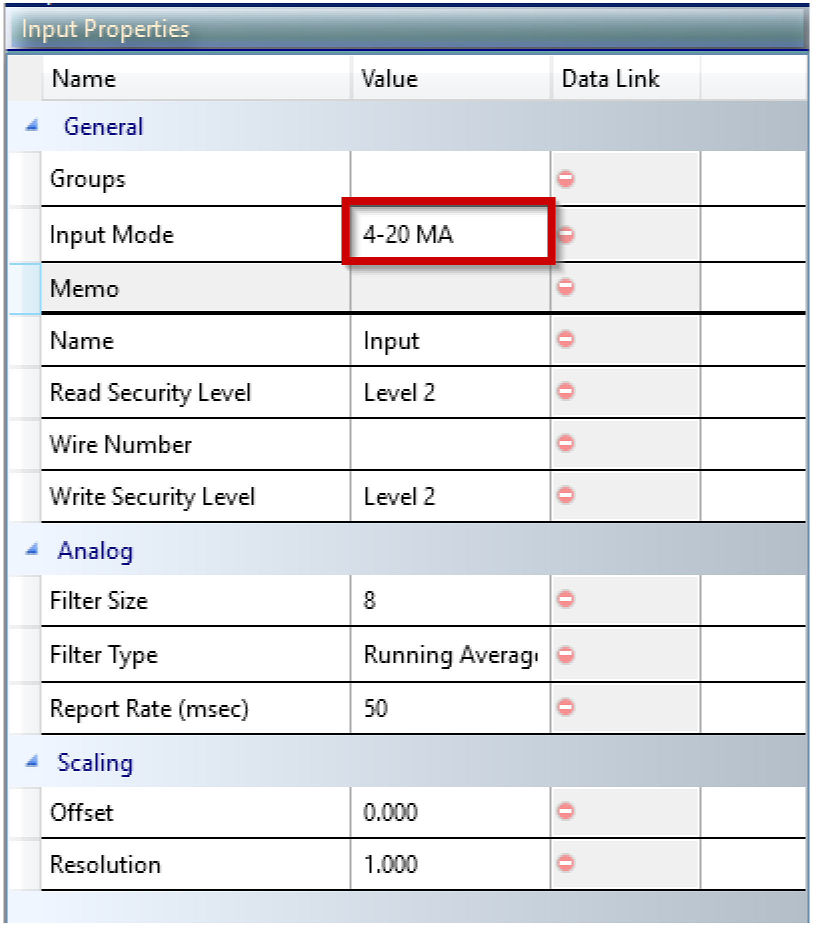
Related:

Input Mode Properties 55
Digital Input Modes 57
4-20 (mA) 63
Voltage to Digital (VTD) 64
Resistance to Digital (RTD) 66
Frequency (Freq) 68
Pulse Width Modulation (PWM) 70
Pulse Counter (Count) 72
Internal 74

Encoder 75

5.1.3.1 4-20 (mA)

This input is used to read sensors that provide a 4 to 20 milliamp (mA) output.



Input Properties

Name	Value	Data Link
General		
Groups		⊖
Input Mode	4-20 MA	⊖
Memo		⊖
Name	Input	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Analog		
Filter Size	8	⊖
Filter Type	Running Averag	⊖
Report Rate (msec)	50	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

4-20 (m A) Input Properties

Properties

1. Groups 253
2. Input Mode 254
3. Memo 255
4. Name 255
5. Read Security Level 257
6. Wire Number 260
7. Write Security Level 260
8. Filter Size 252
9. Filter Type 252
10. Report Rate (msec) 257
11. Offset 256
12. Resolution 257

Related:

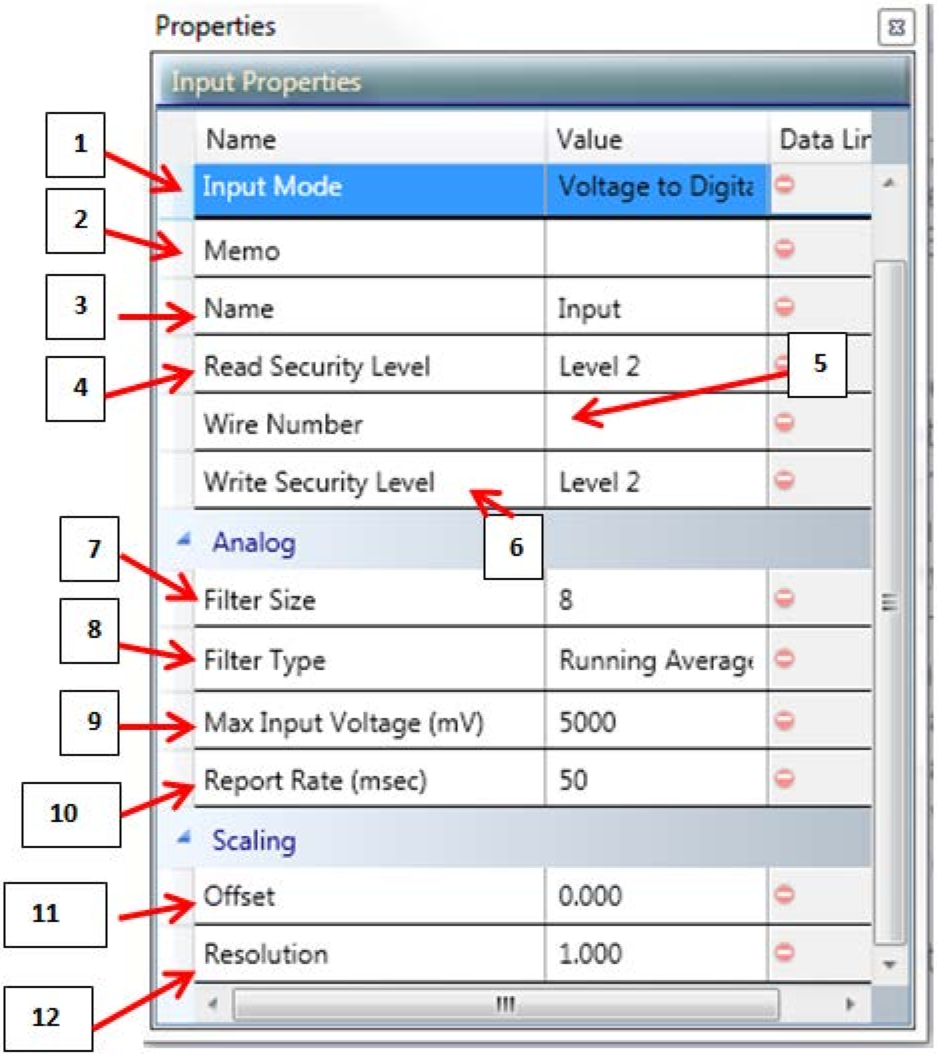
Analog Input Modes 62

5.1.3.2 Voltage to Digital (VTD)

This mode is VTD, and in this mode the input reads in a voltage in millivolts (mV).

Properties

1. Input Mode 254
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Wire Number 260
6. Write Security Level 260
7. Filter Size 252
8. Filter Type 252
9. Max Input Voltage (mV) 255
10. Report Rate (msec) 257
11. Offset 256
12. Resolution 257



Properties

Input Properties

Name	Value	Data Lir
Input Mode	Voltage to Digitε	⊖
Memo		⊖
Name	Input	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
◢ Analog		
Filter Size	8	⊖
Filter Type	Running Averagε	⊖
Max Input Voltage (mV)	5000	⊖
Report Rate (msec)	50	⊖
◢ Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

VTD Input Properties

Related:

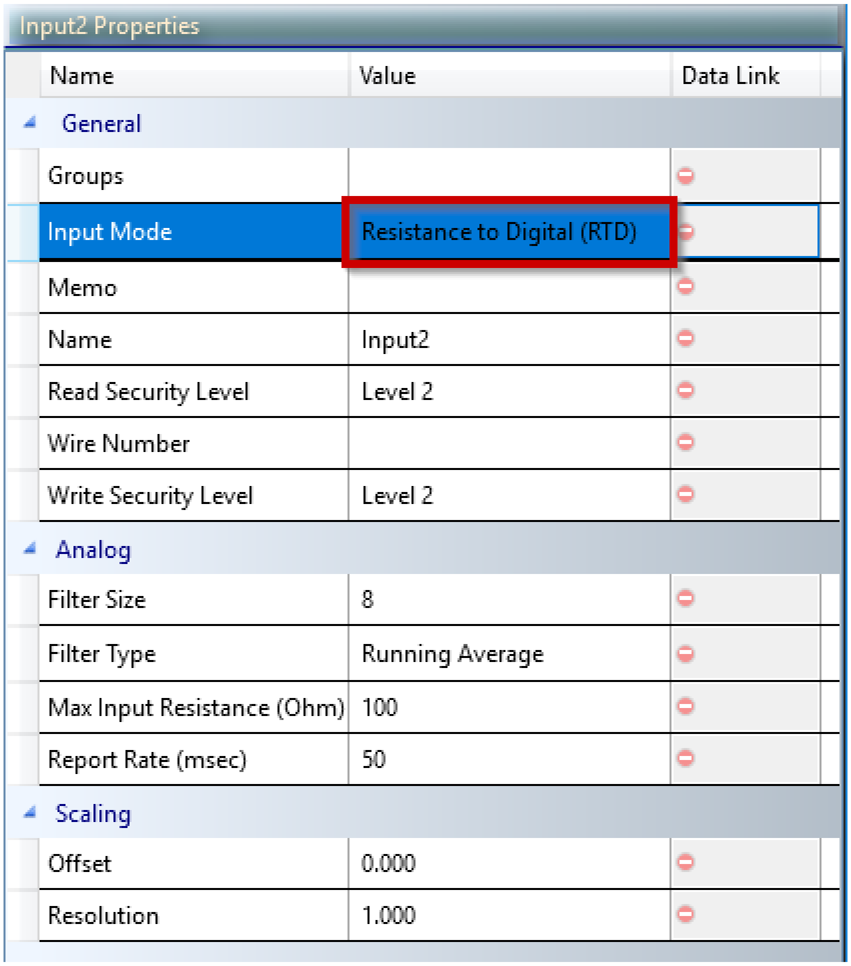
Analog Input Modes 62

5.1.3.3 Resistance to Digital (RTD)

RTD will read in the resistance to ground in Ohms. This input mode's properties work similarly to the VTD mode 64, except that the user defines the Max Input Resistance, in Ohms, instead of the Max Input Voltage. The rest of the properties work exactly the same.

Properties

1. Input Mode 254
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Wire Number 260
6. Write Security Level 260
7. Filter Size 252
8. Filter Type 252
9. Max Input Resistance (?) 255
10. Report Rate (msec) 257
11. Offset 256
12. Resolution 257



Input2 Properties

Name	Value	Data Link
General		
Groups		⊖
Input Mode	Resistance to Digital (RTD)	⊖
Memo		⊖
Name	Input2	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Analog		
Filter Size	8	⊖
Filter Type	Running Average	⊖
Max Input Resistance (Ohm)	100	⊖
Report Rate (msec)	50	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

RTD Input Properties

Related:

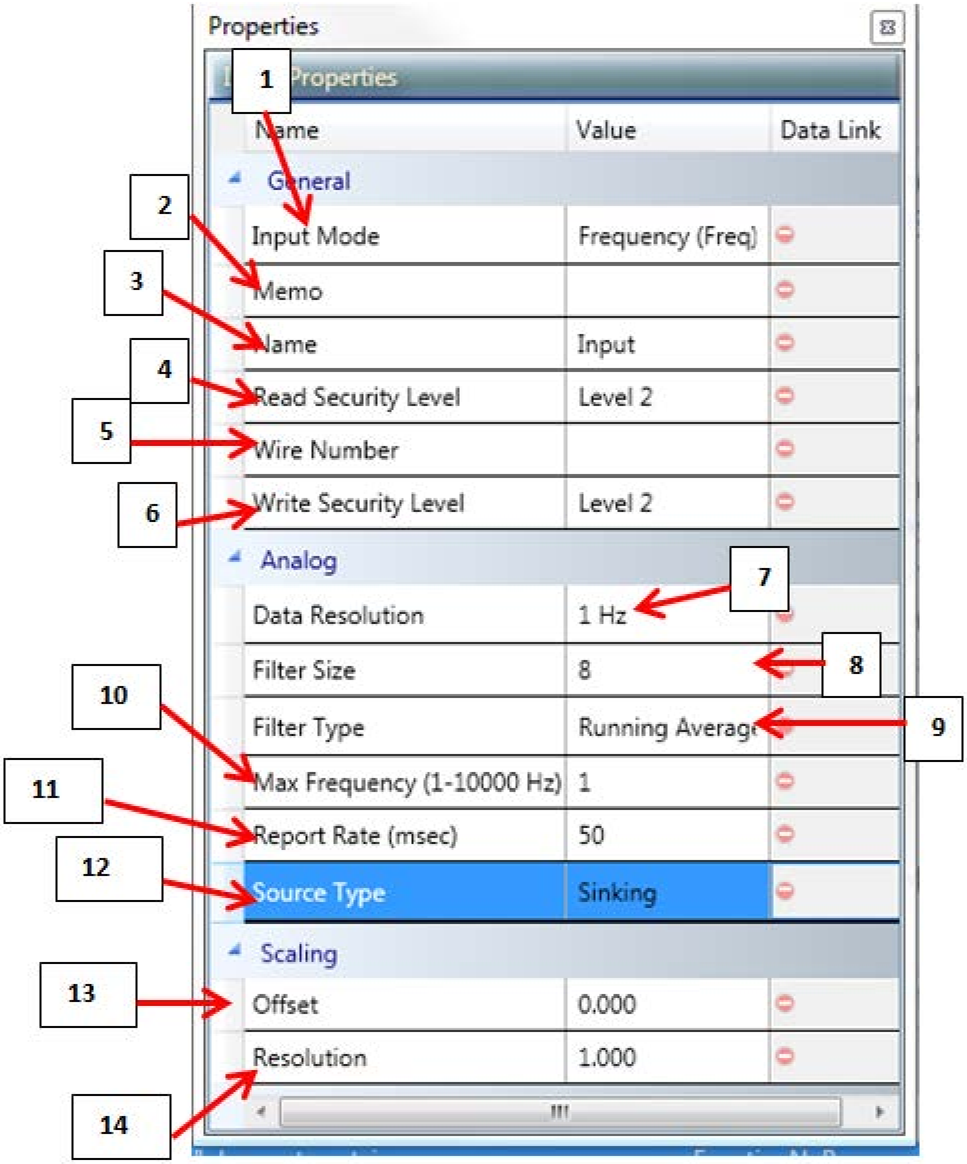
Analog Input Modes 62

5.1.3.4 Frequency (Freq)

This mode reports the frequency measured on the Input Signal in Hertz (Hz).

Properties

1. Input Mode 254
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Wire Number 260
6. Write Security Level 260
7. Data Resolution 250
8. Filter Size 252
9. Filter Type 252
10. Max Frequency (1-10000Hz) 255
11. Report Rate (msec) 257
12. Source Type 258
13. Offset 256
14. Resolution 257



Properties

Properties

Name	Value	Data Link
General		
Input Mode	Frequency (Freq)	⊖
Memo		⊖
Name	Input	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Analog		
Data Resolution	1 Hz	⊖
Filter Size	8	⊖
Filter Type	Running Averag	⊖
Max Frequency (1-10000 Hz)	1	⊖
Report Rate (msec)	50	⊖
Source Type	Sinking	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Freq Input Properties

Related:

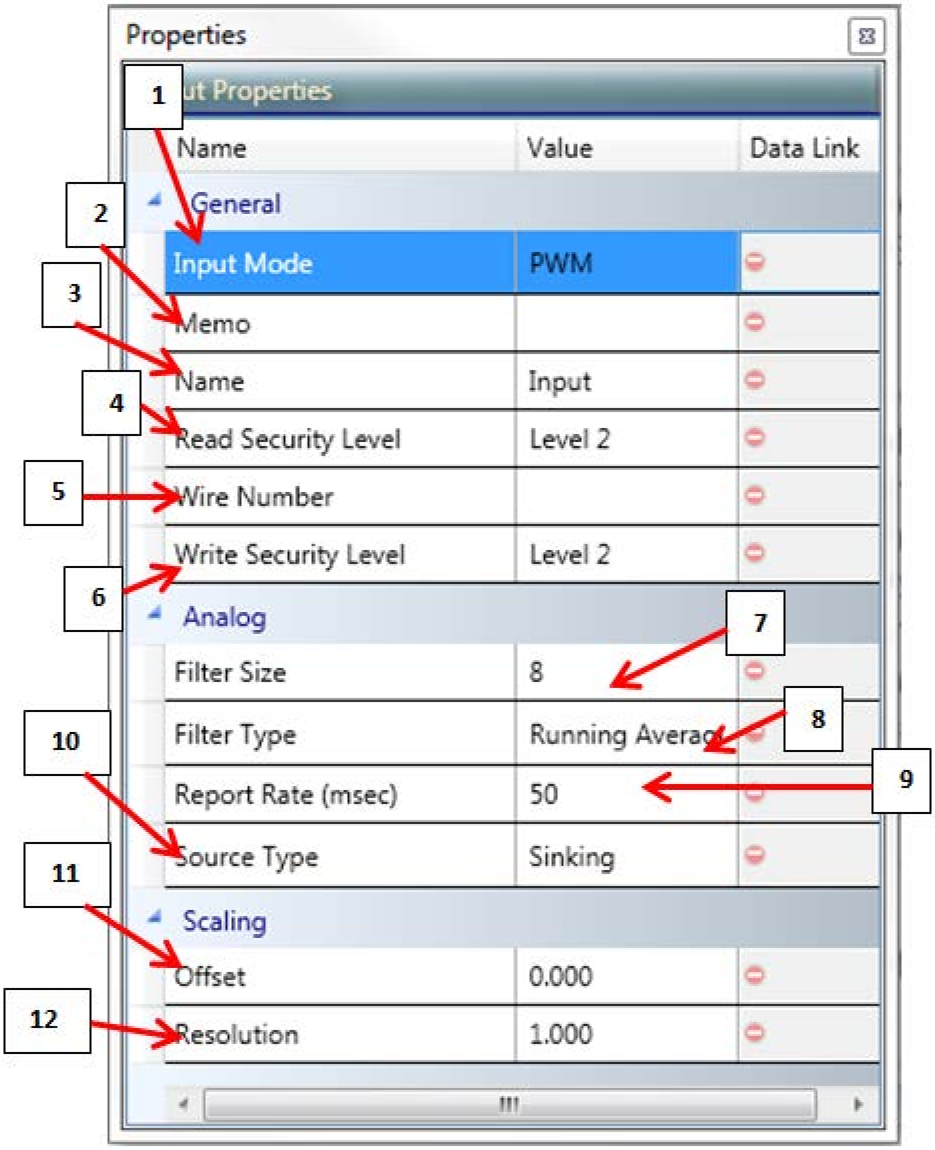
Analog Input Modes 62

5.1.3.5 Pulse Width Modulation (PWM)

PWM is a modulation technique used to control the width of pulses, usually in terms of controlling the power to an electrical device. The PWM input mode measures the duty cycle in 0.1% increments, meaning 0 to 100% is equivalent to values of 0 to 1000 in the data item.

Properties

1. Input Mode 254
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Wire Number 260
6. Write Security Level 260
7. Filter Size 252
8. Filter Type 252
9. Report Rate (msec) 257
10. Source Type 258
11. Offset 256
12. Resolution 257



ut Properties

Name	Value	Data Link
General		
Input Mode	PWM	⊖
Memo		⊖
Name	Input	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Analog		
Filter Size	8	⊖
Filter Type	Running Averag	⊖
Report Rate (msec)	50	⊖
Source Type	Sinking	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

PWM Input Properties

Related:

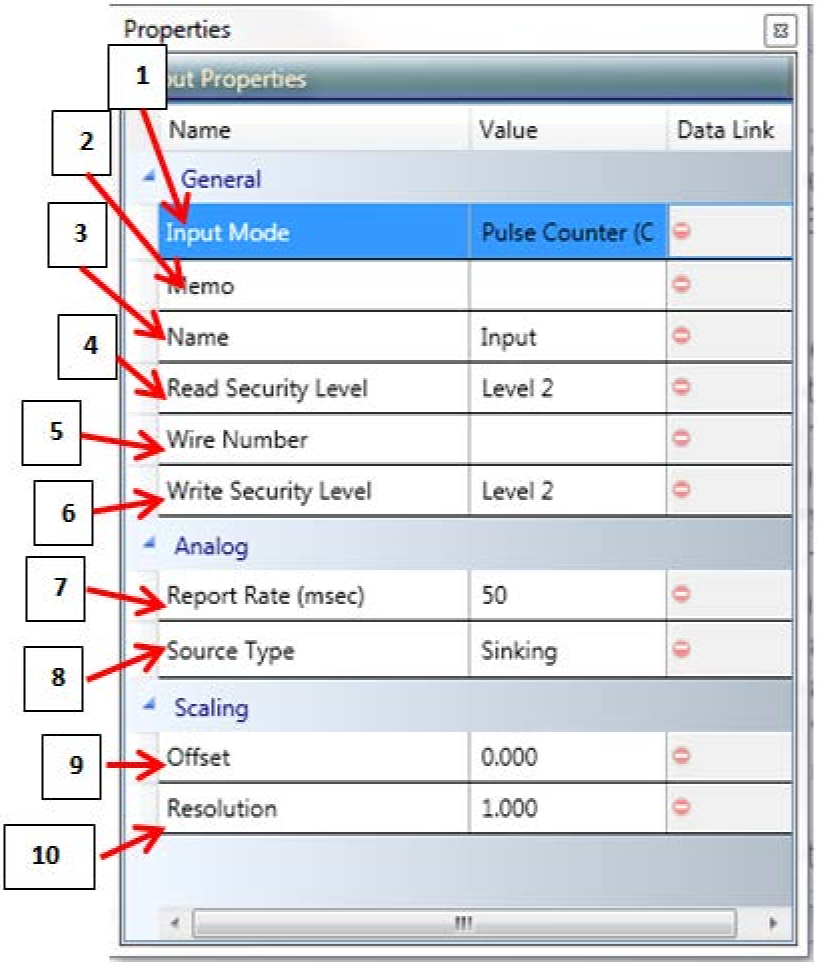
Analog Input Modes 62

5.1.3.6 Pulse Counter (Count)

This mode counts the number of pulses read by the input. The value is reset to 0 with each master loop, or each time a client sends the value in a CAN message as configured by the report rate.

Properties

1. Input Mode 254
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Wire Number 260
6. Write Security Level 260
7. Report Rate (msec) 257
8. Source Type 258
9. Offset 256
10. Resolution 257



Name	Value	Data Link
General		
Input Mode	Pulse Counter (C	
Memo		
Name	Input	
Read Security Level	Level 2	
Wire Number		
Write Security Level	Level 2	
Analog		
Report Rate (msec)	50	
Source Type	Sinking	
Scaling		
Offset	0.000	
Resolution	1.000	

Pulse Counter Input Properties

Related:

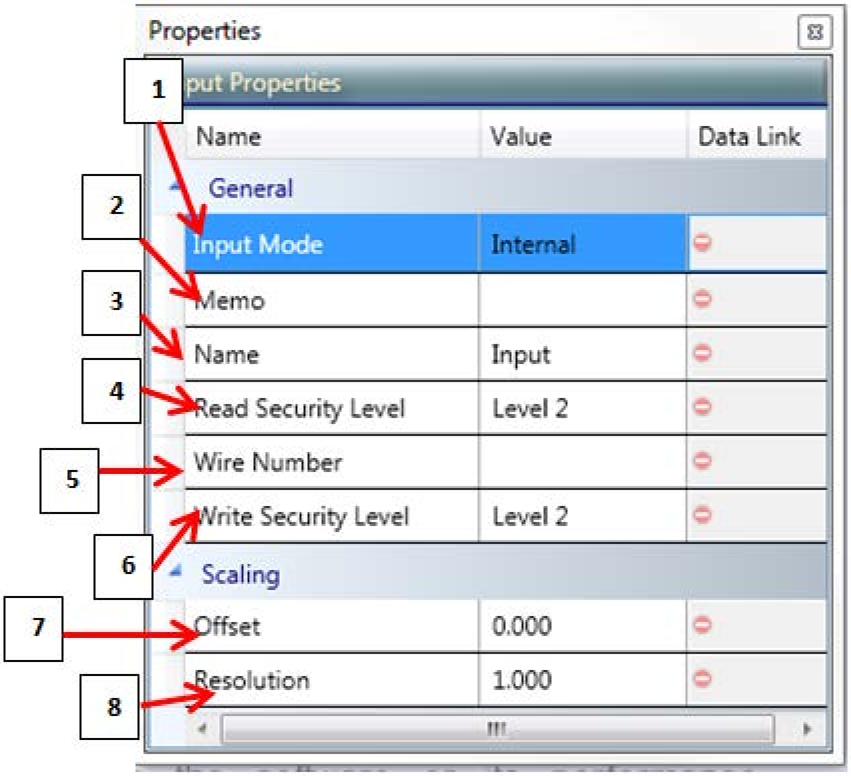
Analog Input Modes 62

5.1.3.7 Internal

The Internal input mode is used to monitor a signal internal to the module, such as a Real Time Clock (RTC), accelerometer, etc.

Properties

1. Input Mode 254
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Wire Number 260
6. Write Security Level 260
7. Offset 256
8. Resolution 257



Name	Value	Data Link
General		
Input Mode	Internal	⊖
Memo		⊖
Name	Input	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Internal Input Properties

Related:

Analog Input Modes 62

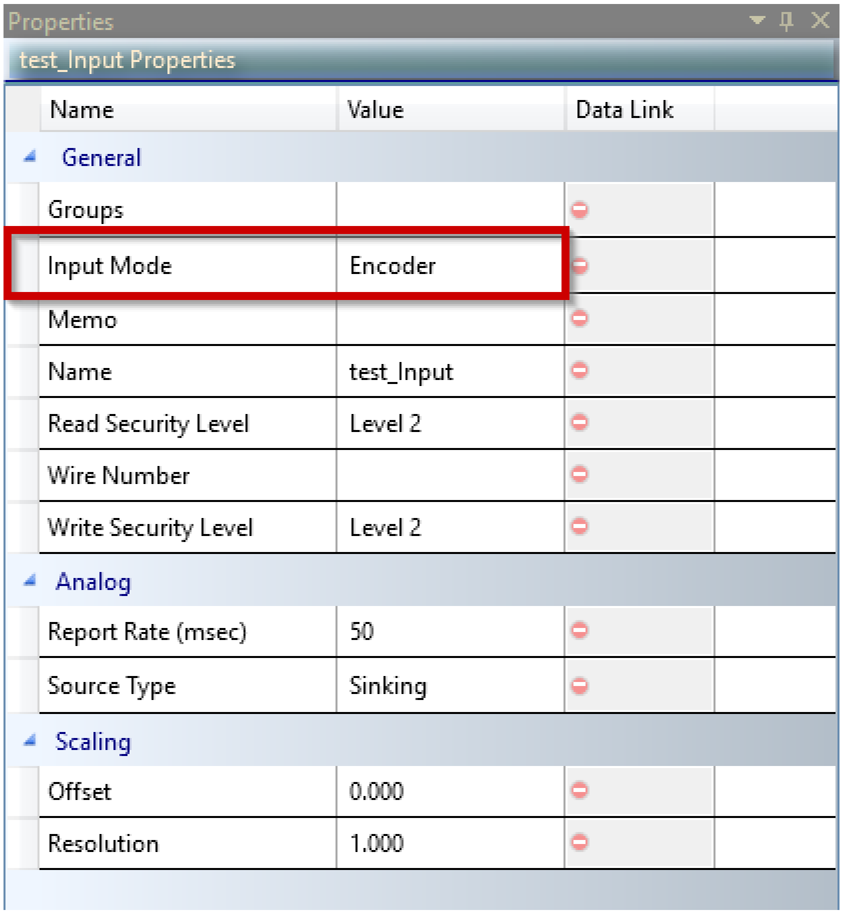
5.1.3.8 Encoder

The Encoder is a combination of the Frequency 68 and Pulse Counter 72 inputs. For modules that support the Encoder input, there are Encoder A and Encoder B pins that must be used as a pair. The order does not matter, but one pin must be a Frequency Input and the other a Pulse Counter.

The Pulse Counter essentially becomes the direction indicator, where spinning clockwise will increment the value up to 1000 and spinning counterclockwise will decrement the value. The Frequency Input will indicate how fast the position is changing.

Properties

For properties, please refer to the Frequency 68 and Pulse Counter 72 input sections.



Name	Value	Data Link
General		
Groups		
Input Mode	Encoder	
Memo		
Name	test_Input	
Read Security Level	Level 2	
Wire Number		
Write Security Level	Level 2	
Analog		
Report Rate (msec)	50	
Source Type	Sinking	
Scaling		
Offset	0.000	
Resolution	1.000	

Encoder Input Properties

Related:
Analog Input Modes 62

5.2 Output

An output within Orchestra® is a pin on the module that has its behavior set by the module. The output will hold a value that the module will use to perform an action based on how the application was written. They're

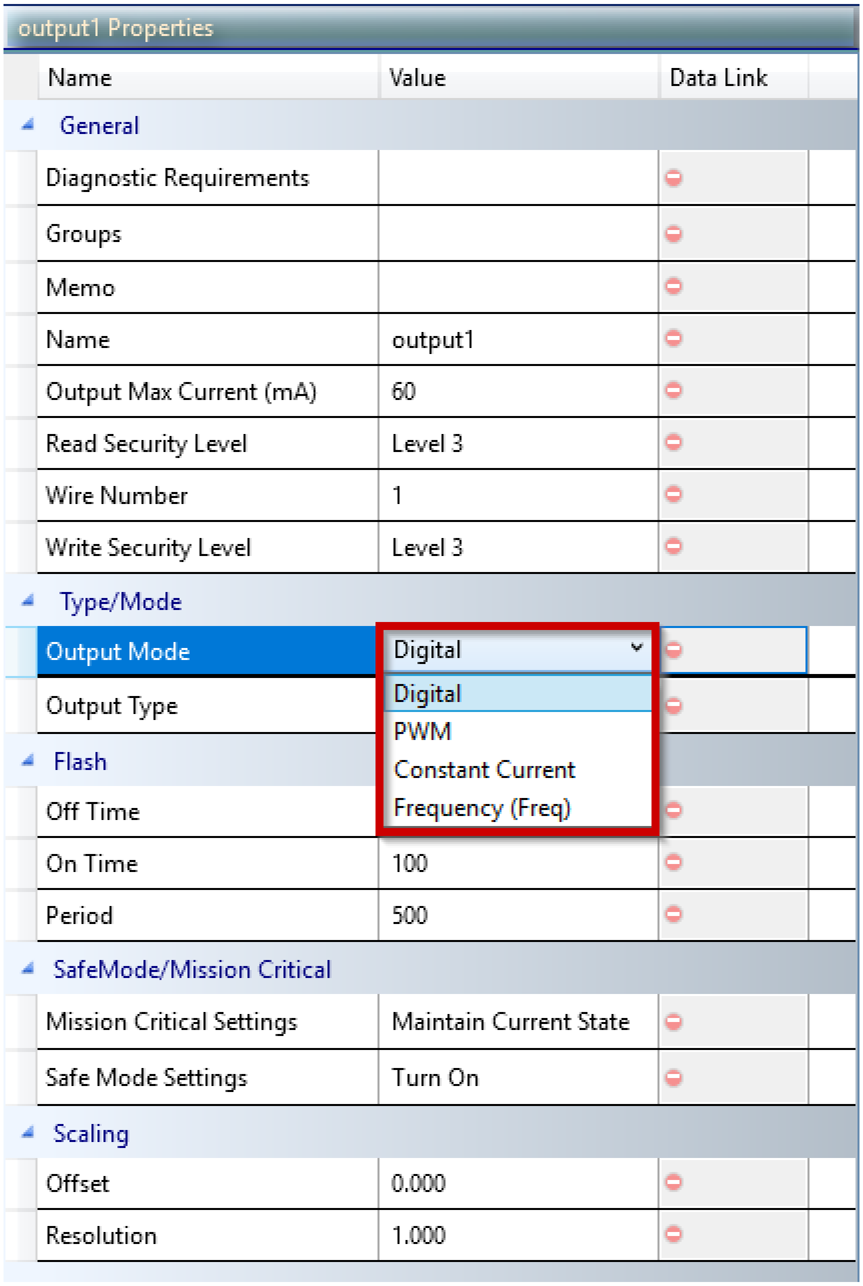
The Output data items are read/write values. Some module outputs can generate status or current data item sub types that can be used in the comparison blocks. All outputs generate a flashing data item sub type that can be used in Comparison and Operator blocks.

Related:

Input 54
Output Mode Properties 77
Operator Blocks 117

5.2.1 Output Mode Properties

Like the Input data item, the Output data item has a number of properties and modes that will define the output so that the application can react and produce expected results.



output1 Properties

Name	Value	Data Link
General		
Diagnostic Requirements		⊖
Groups		⊖
Memo		⊖
Name	output1	⊖
Output Max Current (mA)	60	⊖
Read Security Level	Level 3	⊖
Wire Number	1	⊖
Write Security Level	Level 3	⊖
Type/Mode		
Output Mode	Digital	⊖
Output Type		⊖
Flash		
Off Time		⊖
On Time	100	⊖
Period	500	⊖
SafeMode/Mission Critical		
Mission Critical Settings	Maintain Current State	⊖
Safe Mode Settings	Turn On	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Output Mode Type

Related:

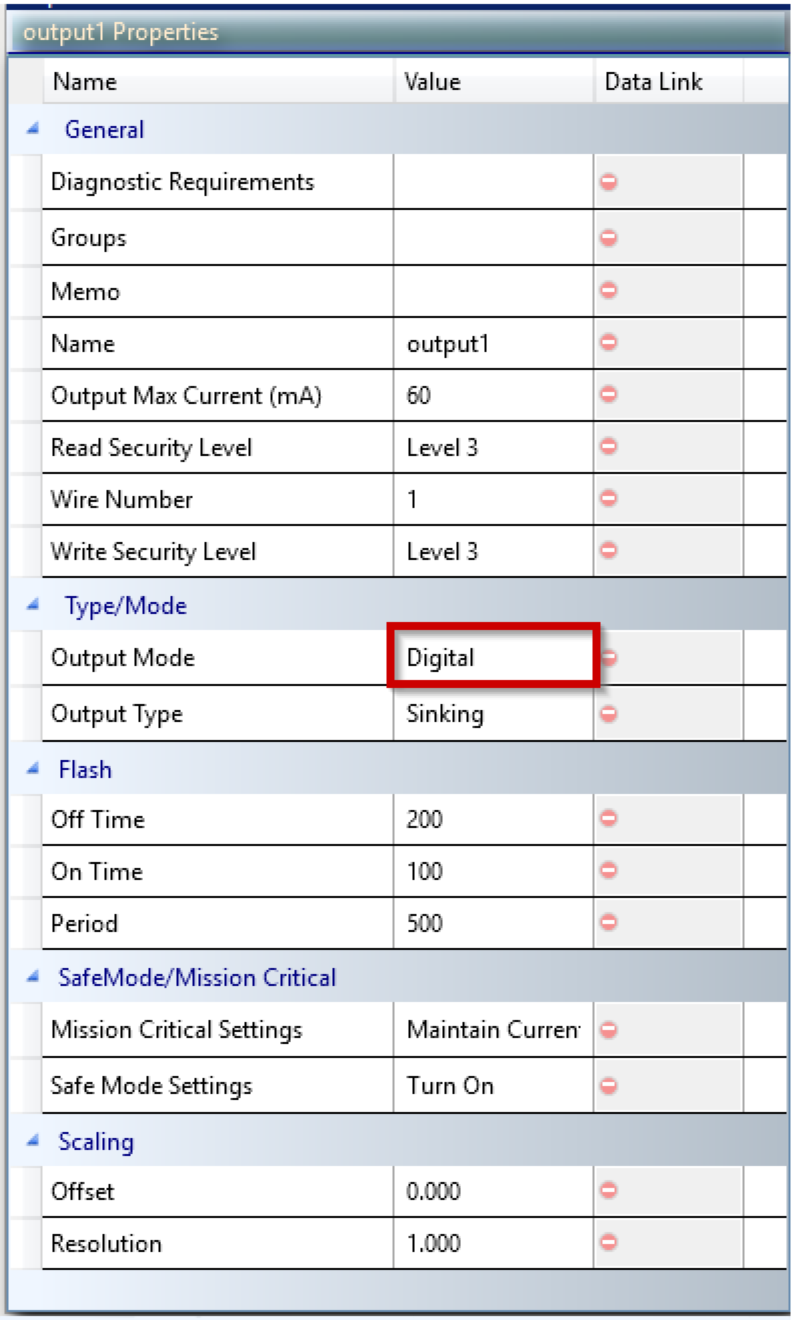
Input 54

5.2.2 Digital

In this mode, the output can be set to be either On or Off.

Properties

1. Current Report Rate 249
2. Diagnostic Requirements 250
3. Groups 253
4. Memo 255
5. Name 255
6. Output Max Current 256
7. Read Security Level 257
8. Wire Number 260
9. Write Security Level 260
10. Current Feedback Type 249
11. Output Mode 256
12. Output Type 256
13. Delay (0-2550 mS) 250
14. Set Point 258
15. Off Time 255
16. On Time 256
17. Period 257
18. Mission Critical Settings 255
19. Safe Mode Settings 258
20. Offset 256
21. Resolution 257



output1 Properties

Name	Value	Data Link
◢ General		
Diagnostic Requirements		⊖
Groups		⊖
Memo		⊖
Name	output1	⊖
Output Max Current (mA)	60	⊖
Read Security Level	Level 3	⊖
Wire Number	1	⊖
Write Security Level	Level 3	⊖
◢ Type/Mode		
Output Mode	Digital	⊖
Output Type	Sinking	⊖
◢ Flash		
Off Time	200	⊖
On Time	100	⊖
Period	500	⊖
◢ SafeMode/Mission Critical		
Mission Critical Settings	Maintain Curren	⊖
Safe Mode Settings	Turn On	⊖
◢ Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Digital Output Properties

Related:

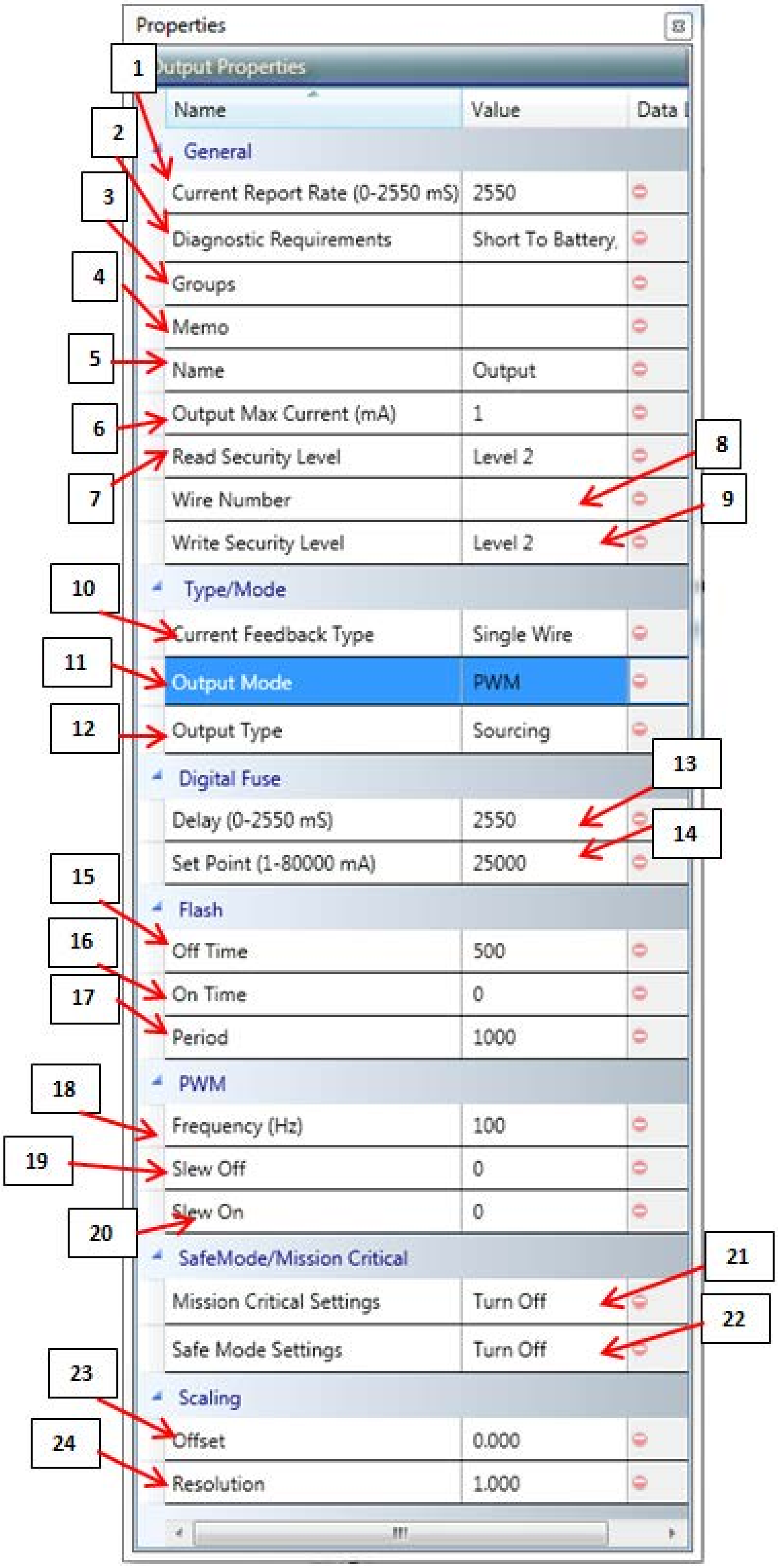
Output Mode Properties 77

5.2.3 PWM

This mode sets the output to produce a PWM signal at a frequency defined by the Frequency 85 property between 40 and 5000Hz. Currently because of firmware limitations, setting the frequency property below 40Hz will produce undesired results.

Properties

1. Current Report Rate 249
2. Diagnostic Requirements 250
3. Groups 253
4. Memo 255
5. Name 255
6. Output Max Current 256
7. Read Security Level 257
8. Wire Number 260
9. Write Security Level 260
10. Current Feedback Type 249
11. Output Mode 256
12. Output Type 256
13. Delay (0-2550 mS) 250
14. Set Point 258
15. Off Time 255
16. On Time 256
17. Period 257
18. Frequency (Hz) 253
19. Slew Off 258
20. Slew On 258
21. Mission Critical Settings 255
22. Safe Mode Settings 258
23. Offset 256
24. Resolution 257



Properties

Output Properties

Name	Value	Data I
General		
Current Report Rate (0-2550 mS)	2550	
Diagnostic Requirements	Short To Battery,	
Groups		
Memo		
Name	Output	
Output Max Current (mA)	1	
Read Security Level	Level 2	
Wire Number		
Write Security Level	Level 2	
Type/Mode		
Current Feedback Type	Single Wire	
Output Mode	PWM	
Output Type	Sourcing	
Digital Fuse		
Delay (0-2550 mS)	2550	
Set Point (1-80000 mA)	25000	
Flash		
Off Time	500	
On Time	0	
Period	1000	
PWM		
Frequency (Hz)	100	
Slew Off	0	
Slew On	0	
SafeMode/Mission Critical		
Mission Critical Settings	Turn Off	
Safe Mode Settings	Turn Off	
Scaling		
Offset	0.000	
Resolution	1.000	

PWM Output Properties

Related:
Frequency 87

5.2.4 Current Controlled (Single Wire)

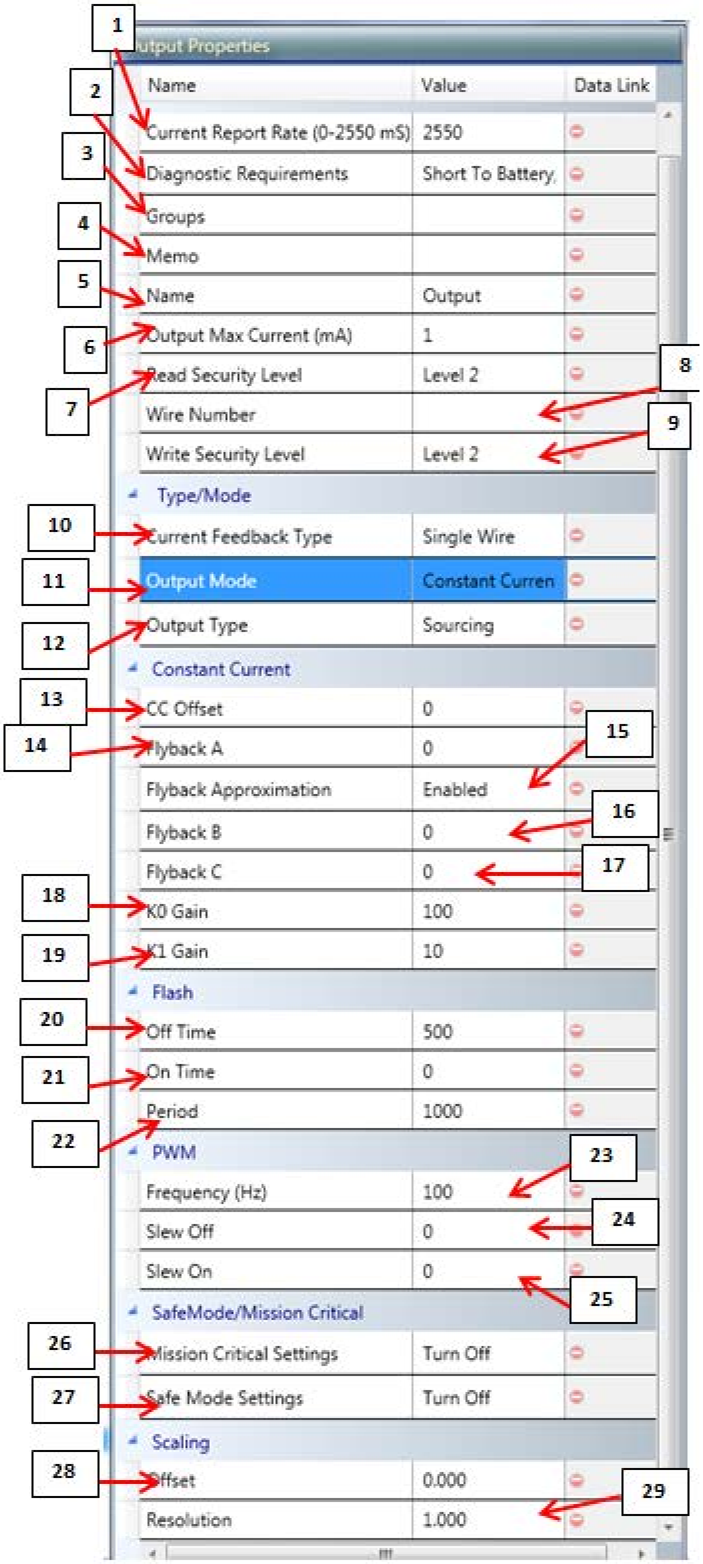
There are two Current Controlled output types: Single Wire and Dual Wire 85. The Single Wire option is descried below.

In this mode, the user sets a current the output should drive. The module will adjust the duty cycle until the current matches the requested value. This is essentially a PWM Output 81 that has a closed loop control to maintain a specific current.

The Single Wire version of this output does not have a feedback line for the current to return. In order to perform a closed loop current control for the output, additional properties are needed to approximate the return.

Properties

1. Current Report Rate 249
2. Diagnostic Requirements 250
3. Groups 253
4. Memo 255
5. Name 255
6. Output Max Current 256
7. Read Security Level 257
8. Wire Number 260
9. Write Security Level 260
10. Current Feedback Type 249
11. Output Mode 256
12. Output Type 256
13. CC Offset 248
14. Flyback A 252
15. Flyback Approximation 252
16. Flyback B 252
17. Flyback C 252
18. K0 Gain 254
19. K1 Gain 254
20. Off Time 255
21. On Time 256
22. Period 257
23. Frequency (Hz) 253
24. Slew Off 258
25. Slew On 258
26. Mission Critical Settings 255
27. Safe Mode Settings 258
28. Offset 256
29. Resolution 257



utput Properties

Name	Value	Data Link
Current Report Rate (0-2550 mS)	2550	⊖
Diagnostic Requirements	Short To Battery,	⊖
Groups		⊖
Memo		⊖
Name	Output	⊖
Output Max Current (mA)	1	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Type/Mode		
Current Feedback Type	Single Wire	⊖
Output Mode	Constant Curren	⊖
Output Type	Sourcing	⊖
Constant Current		
CC Offset	0	⊖
Flyback A	0	⊖
Flyback Approximation	Enabled	⊖
Flyback B	0	⊖
Flyback C	0	⊖
K0 Gain	100	⊖
K1 Gain	10	⊖
Flash		
Off Time	500	⊖
On Time	0	⊖
Period	1000	⊖
PWM		
Frequency (Hz)	100	⊖
Slew Off	0	⊖
Slew On	0	⊖
SafeMode/Mission Critical		
Mission Critical Settings	Turn Off	⊖
Safe Mode Settings	Turn Off	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Current Controlled Single Wire Output Properties

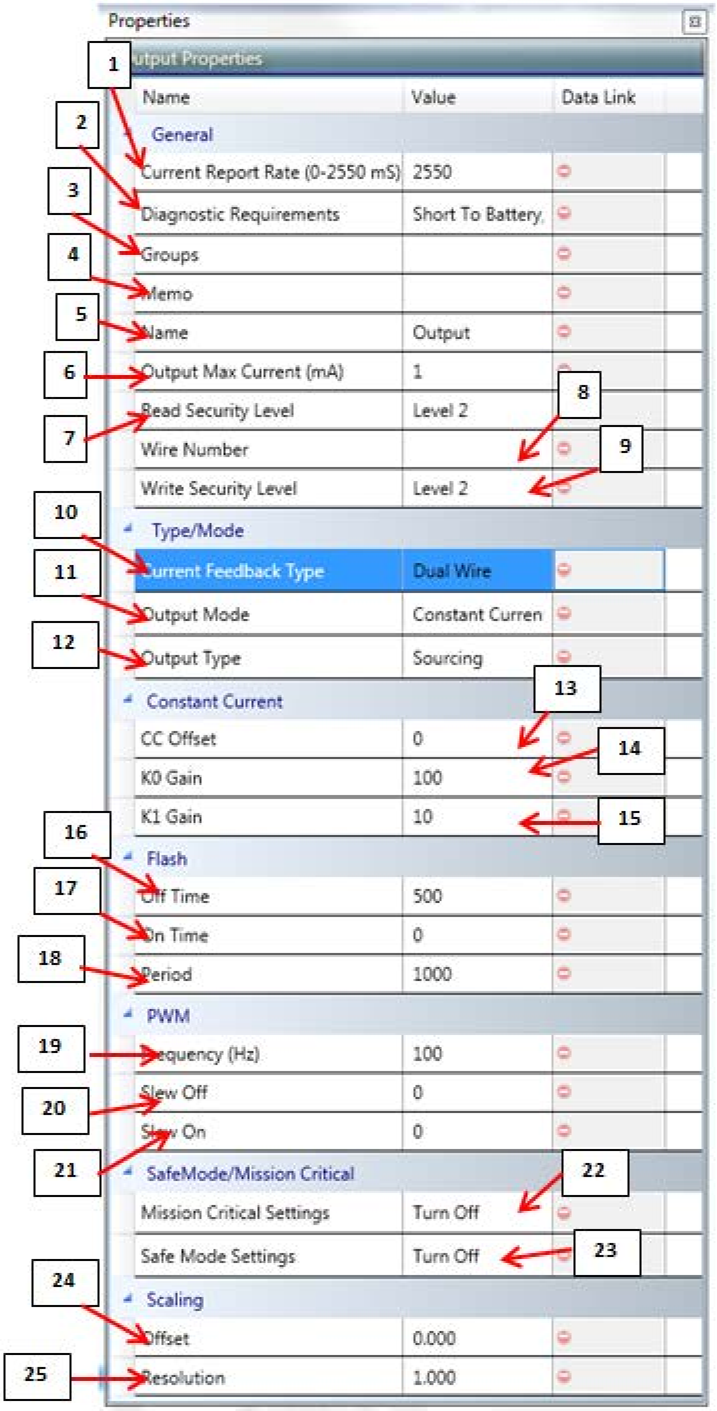
Related:
Current Controlled (Dual Wire) 85
PWM 81

5.2.5 Current Controlled (Dual Wire)

In this mode, the user sets a current that the output should drive, and the module will adjust the duty cycle until the current matches the requested value. This is essentially a PWM Output 81 that has a closed loop control to maintain a specific current. The Dual Wire version of this output has a feedback line for the current to return and does not need the extra properties to approximate the current, since it is measured directly.

Properties

1. Current Report Rate 249
2. Diagnostic Requirements 250
3. Groups 253
4. Memo 255
5. Name 255
6. Output Max Current 256
7. Read Security Level 257
8. Wire Number 260
9. Write Security Level 260
10. Current Feedback Type 249
11. Output Mode 256
12. Output Type 256
13. CC Offset 248
14. K0 Gain 254
15. K1 Gain 254
16. Off Time 255
17. On Time 256
18. Period 257
19. Frequency (Hz) 253
20. Slew Off 258
21. Slew On 258
22. Mission Critical Settings 255
23. Safe Mode Settings 258
24. Offset 256
25. Resolution 257



Properties

utput Properties

Name	Value	Data Link
General		
Current Report Rate (0-2550 mS)	2550	⊖
Diagnostic Requirements	Short To Battery,	⊖
Groups		⊖
Memo		⊖
Name	Output	⊖
Output Max Current (mA)	1	⊖
Read Security Level	Level 2	⊖
Wire Number		⊖
Write Security Level	Level 2	⊖
Type/Mode		
Current Feedback Type	Dual Wire	⊖
Output Mode	Constant Curren	⊖
Output Type	Sourcing	⊖
Constant Current		
CC Offset	0	⊖
K0 Gain	100	⊖
K1 Gain	10	⊖
Flash		
Off Time	500	⊖
On Time	0	⊖
Period	1000	⊖
PWM		
Frequency (Hz)	100	⊖
Slew Off	0	⊖
Slew On	0	⊖
SafeMode/Mission Critical		
Mission Critical Settings	Turn Off	⊖
Safe Mode Settings	Turn Off	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Current Controlled Dual Wire Output Properties

Related:

Current Controlled (Single Wire) 83
PWM 81

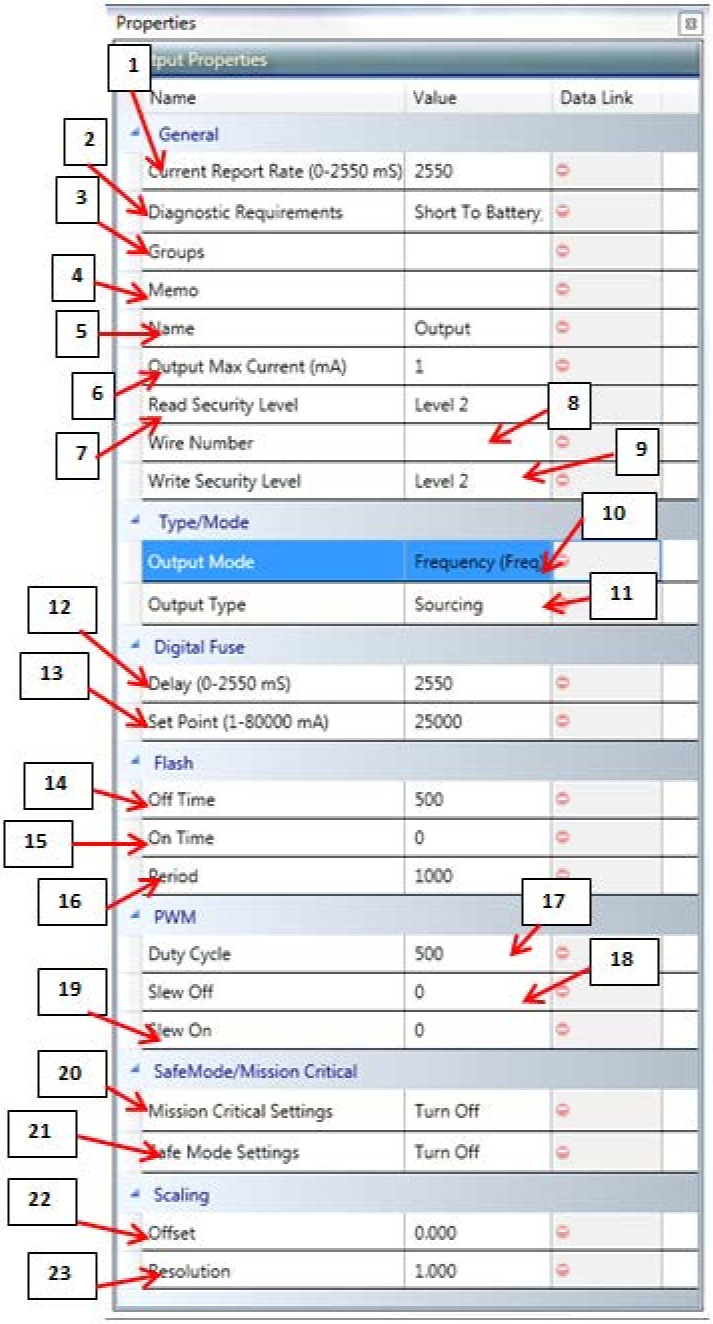
5.2.6 Frequency

The output has the ability to produce a variable frequency (duty cycle constant). The duty cycle for the frequency can be adjusted by setting the Duty Cycle property to a value between 100 and 900 corresponding to 10% and 90%.

Like the PWM Output 81, the Slew On and Off rates can be adjusted to control how fast the frequency will ramp up the set point and back down to zero.

Properties

1. Current Report Rate 249
2. Diagnostic Requirements 250
3. Groups 253
4. Memo 255
5. Name 255
6. Output Max Current 256
7. Read Security Level 257
8. Wire Number 260
9. Write Security Level 260
10. Output Mode 256
11. Output Type 256
12. Delay (0-2550 mS) 250
13. Set Point (1-80000 mA) 258
14. Off Time 255
15. On Time 256
16. Period 257
17. Frequency (Hz) 253
18. Slew Off 258
19. Slew On 258
20. Mission Critical Settings 255
21. Safe Mode Settings 258
22. Offset 256
23. Resolution 257



Properties

tput Properties

Name	Value	Data Link
General		
Current Report Rate (0-2550 mS)	2550	⊖
Diagnostic Requirements	Short To Battery,	⊖
Groups		⊖
Memo		⊖
Name	Output	⊖
Output Max Current (mA)	1	⊖
Read Security Level	Level 2	
Wire Number		⊖
Write Security Level	Level 2	⊖
Type/Mode		
Output Mode	Frequency (Freq)	⊖
Output Type	Sourcing	
Digital Fuse		
Delay (0-2550 mS)	2550	⊖
Set Point (1-80000 mA)	25000	⊖
Flash		
Off Time	500	⊖
On Time	0	⊖
Period	1000	⊖
PWM		
Duty Cycle	500	⊖
Slew Off	0	⊖
Slew On	0	⊖
SafeMode/Mission Critical		
Mission Critical Settings	Turn Off	⊖
Safe Mode Settings	Turn Off	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Frequency Output Properties

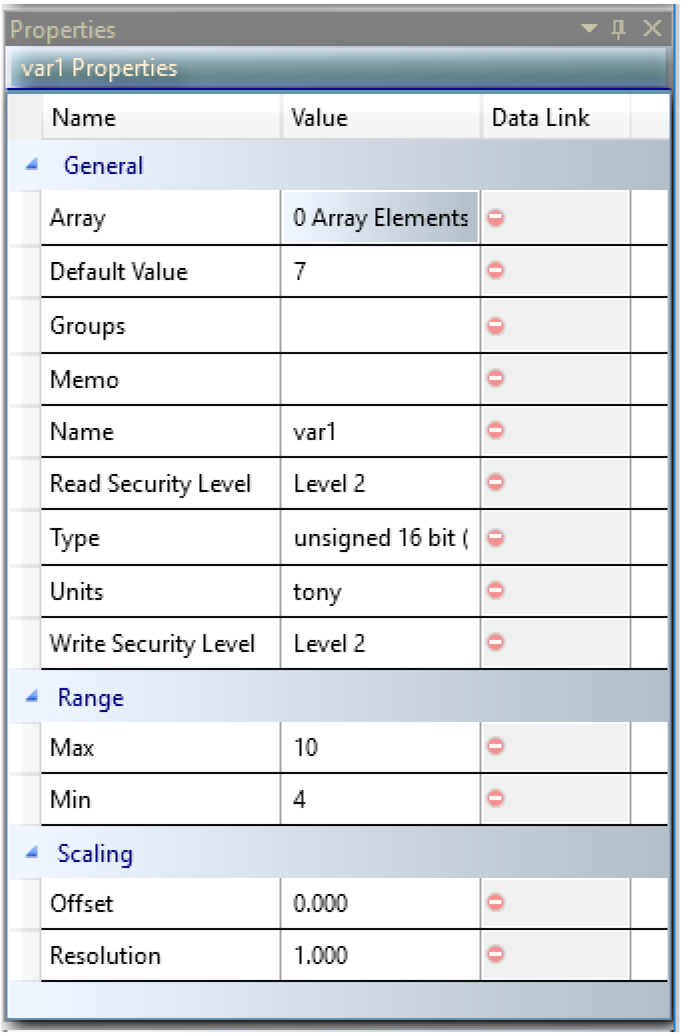
Note: The PWM and Constant Current have two extra Output Types to choose from: PVG and EDC. Those two settings are hardware specific settings that are used when HED® electronics are interfacing with a specific competitor's hydraulic equipment.

Related:

PWM 81

5.3 Variable

The Variable data item is used to save volatile and calculated values as a single value or as an array with multiple values.



Name	Value	Data Link
General		
Array	0 Array Elements	⊖
Default Value	7	⊖
Groups		⊖
Memo		⊖
Name	var1	⊖
Read Security Level	Level 2	⊖
Type	unsigned 16 bit (	⊖
Units	tony	⊖
Write Security Level	Level 2	⊖
Range		
Max	10	⊖
Min	4	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Variable Properties

Properties

1. Array 248
2. Default Value 250
3. Groups 253
4. Memo 255
5. Name 255
6. Read Security Level 257
7. Type 260
8. Units 260
9. Write Security Level 260
10. Max 254

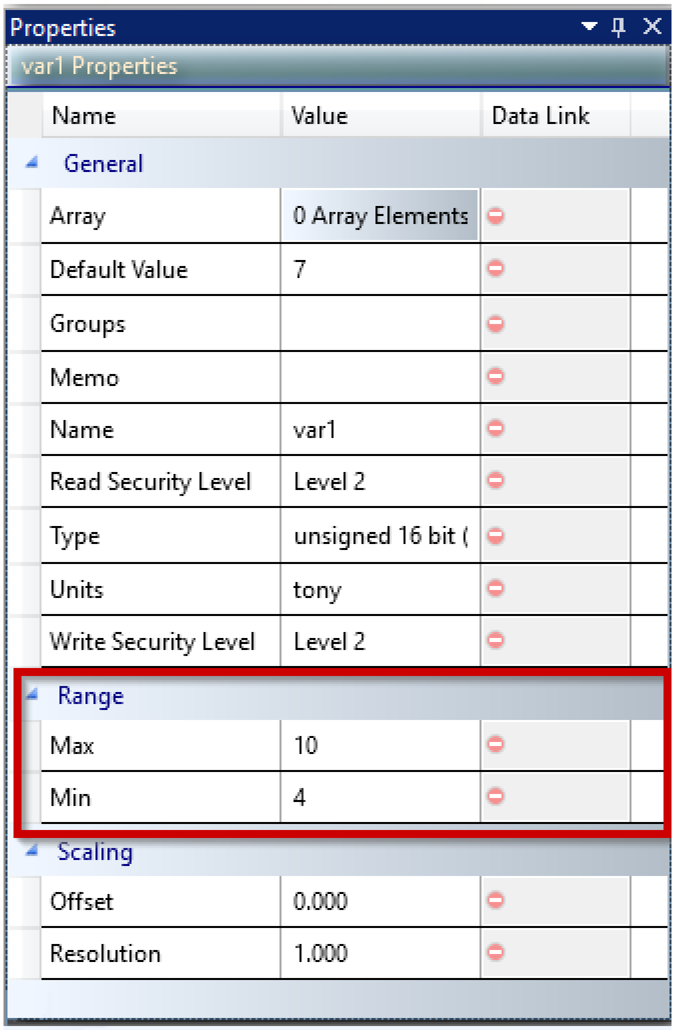
11. Min 255
12. Offset 256
13. Resolution 257

Related:

Define Variable Range 92
Set Variable Array Size 92
State Machine 95

5.3.1 Define Variable Range

Define the Max and Min values within the Range portion of the Property pane; with the Max value being limited by the type of Variable. The Variable can be defined as an unsigned 16 bit (0-65,535), unsigned 32 bit (0-4,294,967,295), and an alarm.



Properties

var1 Properties

Name	Value	Data Link
General		
Array	0 Array Elements	⊖
Default Value	7	⊖
Groups		⊖
Memo		⊖
Name	var1	⊖
Read Security Level	Level 2	⊖
Type	unsigned 16 bit (	⊖
Units	tony	⊖
Write Security Level	Level 2	⊖
Range		
Max	10	⊖
Min	4	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖

Variable Range

Related:

Set Variable Array Size 92
State Machine 95
Variable 90

5.3.2 Set Variable Array Size

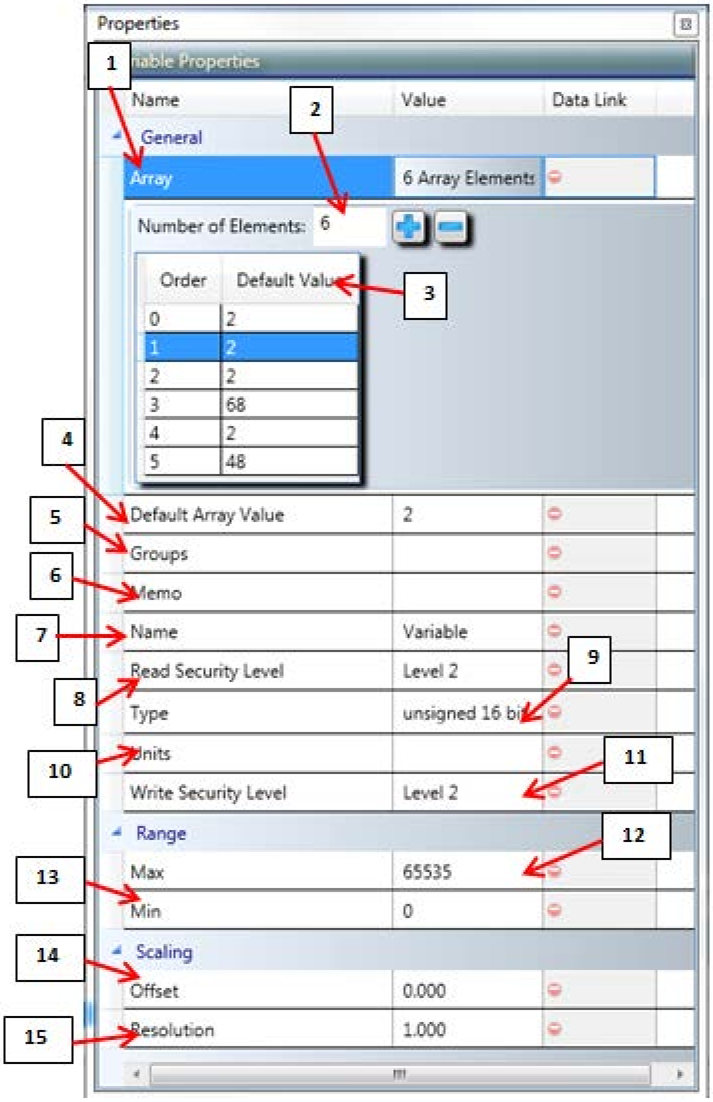
The size of the array is chosen by left clicking on the Array property and either typing in the size of the array desired or by clicking the "+" button.

An array with size zero will be a variable that can hold a single value. As the array size increases, the user can define the default values of each element in the array using the small table that appears below the "Number of Elements" in the Property pane. The

Default Array Value property below the Array property sets a global default for the array, so each additional element added will be initialized to that global value.

Properties

1. Array 248
2. Number of Elements 255
3. Default Value 250
4. Default Array Value 250
5. Groups 253
6. Memo 255
7. Name 255
8. Read Security Level 257
9. Type 260
10. Units 260
11. Write Security Level 260
12. Max 254
13. Min 255
14. Offset 256
15. Resolution 257



Properties

riable Properties

Name	Value	Data Link
General		
Array	6 Array Elements	

Number of Elements: 6

Order	Default Valu
0	2
1	2
2	2
3	68
4	2
5	48

Name	Value	Data Link
Default Array Value	2	
Groups		
Memo		
Name	Variable	
Read Security Level	Level 2	
Type	unsigned 16 bi	
Units		
Write Security Level	Level 2	
Range		
Max	65535	
Min	0	
Scaling		
Offset	0.000	
Resolution	1.000	

Variable Properties

Related:

[Variable] 90

[Define Variable Range] 92

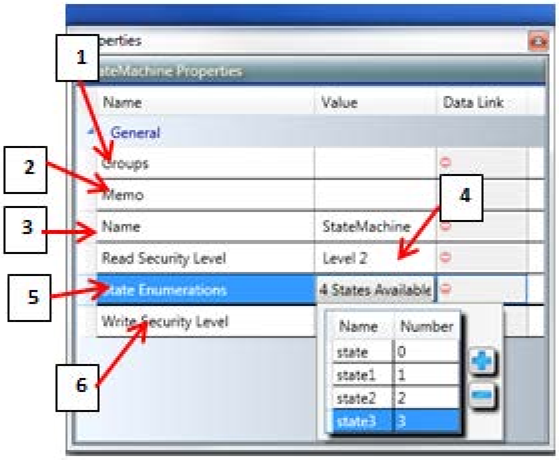
[State Machine] 95

5.3.3 State Machine

The State Machine data item is a volatile data item. The largest difference is that the value(s) of the state machine only get updated at the end of each loop. A State Enumerations property allows the user to define the number of states available, as well as define the names and numbers for those states. Each state must have a unique number and name, so Orchestra® will not allow the user to enter duplicate states.

Properties

1. Groups 253
2. Memo 255
3. Name 255
4. Read Security Level 257
5. State Enumerations 259
6. Write Security Level 260



Name	Value	Data Link
General		
Groups		
Memo		
Name	StateMachine	
Read Security Level	Level 2	
State Enumerations	4 States Available	
Write Security Level		

Name	Number
state	0
state1	1
state2	2
state3	3

State Machine Properties

Related:

Variable 90
Define Variable Range 92
Set Variable Array Size 92

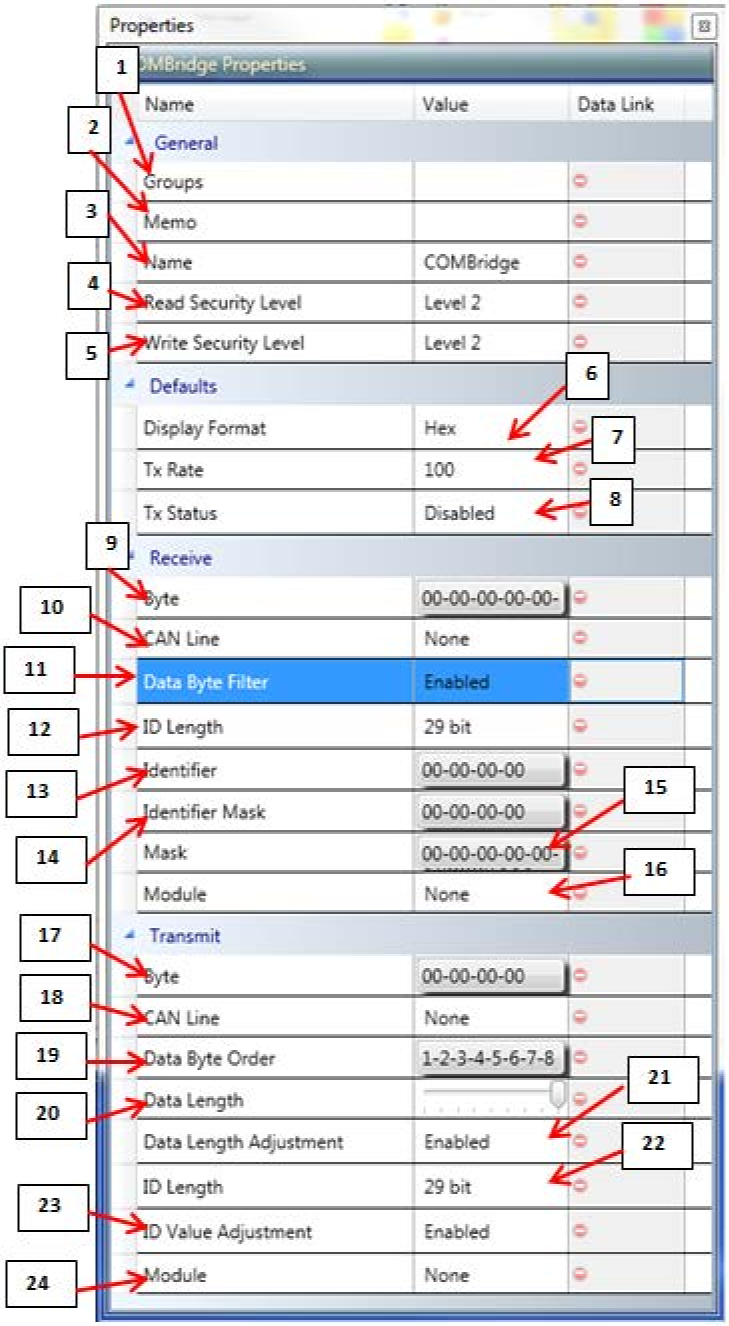
5.4 COM Bridge

A COM Bridge data item is used to pass some or all CAN messages based on an identifier and/or data. This is useful for acting as a CAN filter, a CAN Bus extender, or as a way to reorder the messages coming through.

The COM Bridge is limited to passing only messages with a similar Identifier. If there are multiple messages with different Identifiers, a COM Bridge would have to be created for each of those messages.

Properties

1. Groups 253
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Write Security Level 260
6. Display Format 251
7. Tx Rate 260
8. Tx Status 260
9. Byte 248
10. CAN Line 248
11. Data Byte Filtering 249
12. ID Length 253
13. Identifier 254
14. Identifier Mask 254
15. Mask 254
16. Module 255
17. Byte 248
18. CAN Line 248
19. Data Byte Order 250
20. Data Length 250
21. Data Length Adjustment 250
22. ID Length 253
23. ID Value Adjustment 254
24. Module 255
25. CAN Line 248
26. Min Transmit Period 248
27. Module 248
28. Max 248
29. Min 248
30. Units 248
31. Default Rx Status 248
32. Default Rx Value 248
33. Offset 248



Properties

MBridge Properties

Name	Value	Data Link
General		
Groups		
Memo		
Name	COMBridge	
Read Security Level	Level 2	
Write Security Level	Level 2	
Defaults		
Display Format	Hex	
Tx Rate	100	
Tx Status	Disabled	
Receive		
Byte	00-00-00-00-00-	
CAN Line	None	
Data Byte Filter	Enabled	
ID Length	29 bit	
Identifier	00-00-00-00	
Identifier Mask	00-00-00-00	
Mask	00-00-00-00-00-	
Module	None	
Transmit		
Byte	00-00-00-00	
CAN Line	None	
Data Byte Order	1-2-3-4-5-6-7-8	
Data Length		
Data Length Adjustment	Enabled	
ID Length	29 bit	
ID Value Adjustment	Enabled	
Module	None	

COM Bridge Properties

Related:

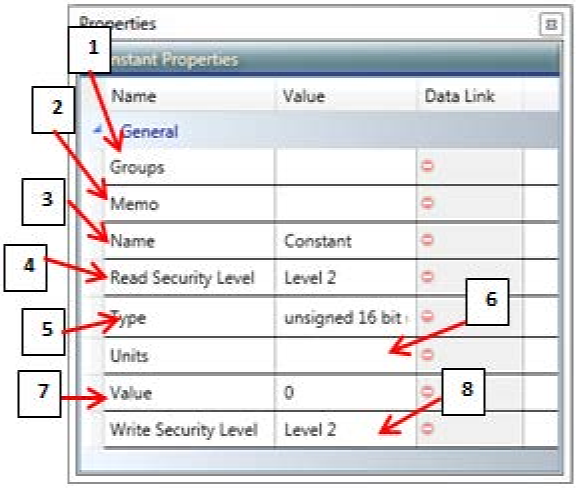
Data Items 54

5.5 Constant

The Constant data item is a static value defined by the user through the properties.

Properties

1. Groups 253
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Type 260
6. Units 260
7. Value 260
8. Write Security Level 260



Properties

Constant Properties

Name	Value	Data Link
General		
Groups		⊖
Memo		⊖
Name	Constant	⊖
Read Security Level	Level 2	⊖
Type	unsigned 16 bit	⊖
Units		⊖
Value	0	⊖
Write Security Level	Level 2	⊖

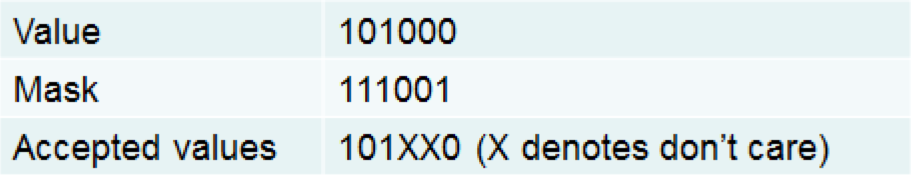
Constant Properties

Related:

Data Items 54

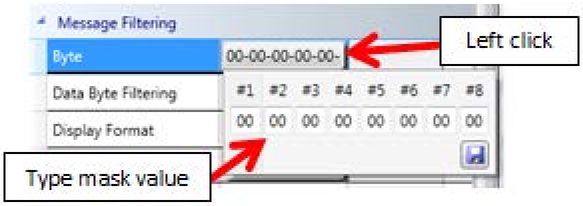
5.6 CAN Receive

CAN Receive is used to read in a piece of data from received CAN messages off of the CAN bus and then sets a status to 1 (Received) from a 0 (Clear) each time that message is received. Up to 32 bits can be read per CAN Receive, so to read in an entire message it may take multiple CAN Receives.



Value	101000
Mask	111001
Accepted values	101XX0 (X denotes don't care)

Mask Example



Message Filtering

Byte	00-00-00-00-00-
Data Byte Filtering	
Display Format	

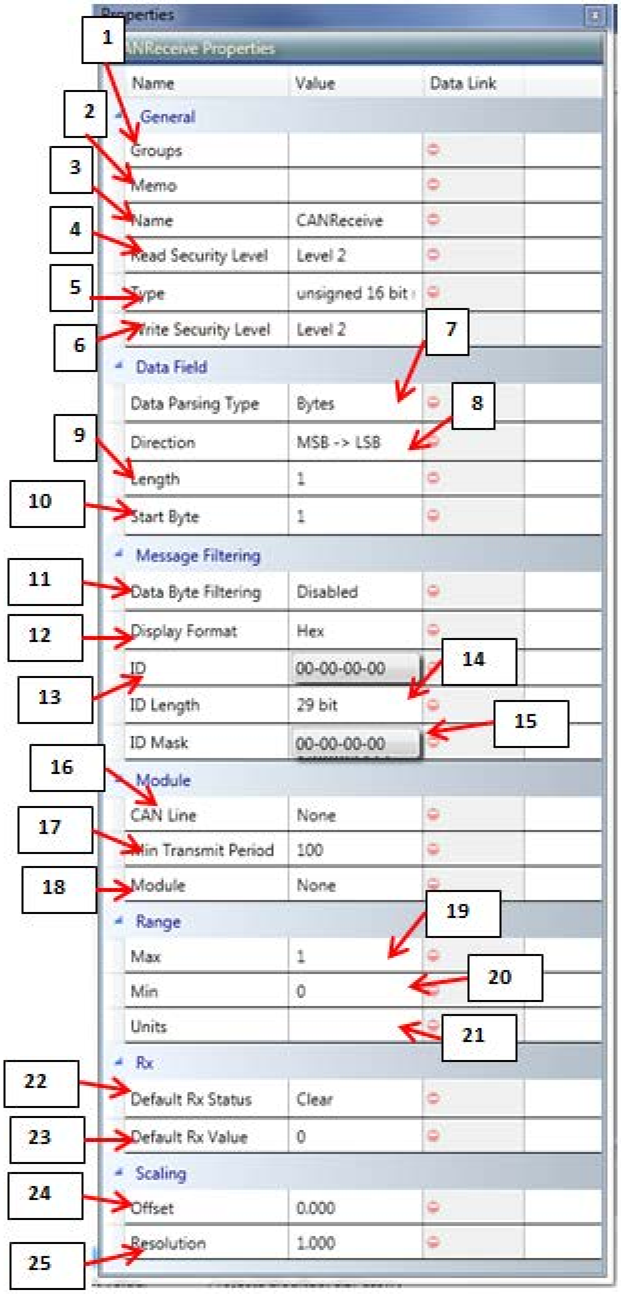
#1	#2	#3	#4	#5	#6	#7	#8
00	00	00	00	00	00	00	00

Mask Bit Selection

Properties

1. Groups 253
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Type 260
6. Write Security Level 260
7. Data Parsing Type 250
8. Direction CAN 250
9. Length 254
10. Start Byte 258
11. Data Byte Filtering 249
12. Display Format 251
13. ID 253
14. ID Length 253
15. ID Mask 253
16. CAN Line 248
17. Min Transmit Period 255
18. Module 255
19. Max 254
20. Min 255
21. Units 260

22. Default Rx Status 250
23. Default Rx Value 250
24. Offset 256
25. Resolution 257



Properties

CANReceive Properties

Name	Value	Data Link
General		
Groups		
Memo		
Name	CANReceive	
Read Security Level	Level 2	
Type	unsigned 16 bit	
Write Security Level	Level 2	
Data Field		
Data Parsing Type	Bytes	
Direction	MSB -> LSB	
Length	1	
Start Byte	1	
Message Filtering		
Data Byte Filtering	Disabled	
Display Format	Hex	
ID	00-00-00-00	
ID Length	29 bit	
ID Mask	00-00-00-00	
Module		
CAN Line	None	
Min Transmit Period	100	
Module	None	
Range		
Max	1	
Min	0	
Units		
Rx		
Default Rx Status	Clear	
Default Rx Value	0	
Scaling		
Offset	0.000	
Resolution	1.000	

CAN Receive Properties

Related:

Data Items 54

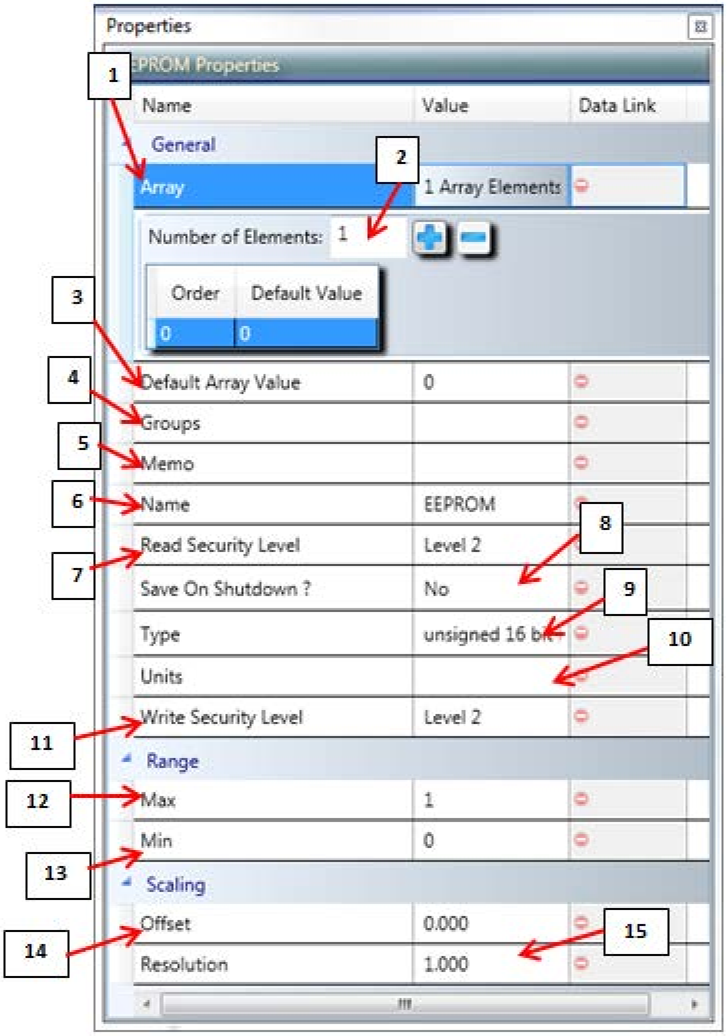
5.7 EEPROM

The EEPROM data item has similar properties to a Timer 103 and a Variable 90. This data item is useful for allowing the customer or end user some flexibility in their application to adjust the values of other data items by using the EEPROM.

The values from EEPROM get read and placed into variables before any rungs within Orchestra® are processed. When writing to the EEPROM, the program will actually write to the variable location. In order to save the EEPROM values, the option to save on shutdown within the Properties pane must be changed to Yes.

Properties

1. Array 248
2. Number of Elements 255
3. Default Value 250
4. Default Array Value 250
5. Groups 253
6. Memo 255
7. Name 255
8. Read Security Level 257
9. Type 260
10. Units 260
11. Write Security Level 260
12. Max 254
13. Min 255
14. Offset 256
15. Resolution 257



Properties

EPROM Properties

Name	Value	Data Link
General		
Array	1 Array Elements	
Number of Elements: 1		
Default Array Value	0	
Groups		
Memo		
Name	EEPROM	
Read Security Level	Level 2	
Save On Shutdown ?	No	
Type	unsigned 16 bit	
Units		
Write Security Level	Level 2	
Range		
Max	1	
Min	0	
Scaling		
Offset	0.000	
Resolution	1.000	

Order	Default Value
0	0

EEPROM Properties

Related:

Variable 90

Time Counter 103

5.8 Time Counter

Time Counters are data items that increment or decrement a set number of times within a defined time frame.

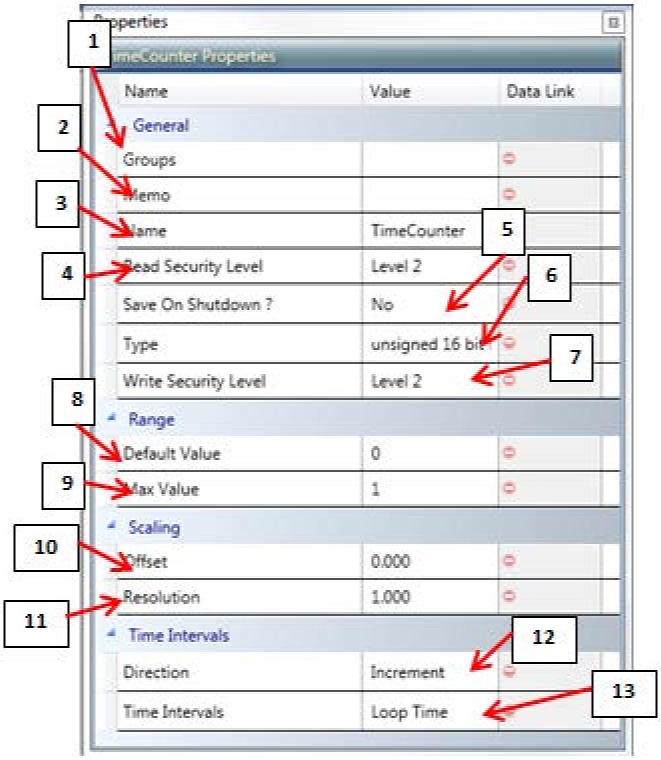
Note: There is no Min Value to set, it is always zero.

The counter will decrement or increment once per Time Interval; so, the total amount of time it will take the counter to fully increment or decrement is a product of the Time Interval and the value that the user sets.

In order to properly implement the timer within Rung Logic, the Time Counter's sub state must be set to Run, since the default state of the timer is Paused. This is also true within C code if using an Orchestra® Time Counter.

Properties

1. Groups 253
2. Memo 255
3. Name 255
4. Read Security Level 257
5. Save On Shutdown? 258
6. Type 260
7. Write Security Level 260
8. Default Value 250
9. Max Value 254
10. Offset 256
11. Resolution 257
12. Direction 250
13. Time Interval 259



Properties

TimeCounter Properties

Name	Value	Data Link
General		
Groups		⊖
Memo		⊖
Name	TimeCounter	
Read Security Level	Level 2	⊖
Save On Shutdown ?	No	
Type	unsigned 16 bit	⊖
Write Security Level	Level 2	⊖
Range		
Default Value	0	⊖
Max Value	1	⊖
Scaling		
Offset	0.000	⊖
Resolution	1.000	⊖
Time Intervals		
Direction	Increment	⊖
Time Intervals	Loop Time	⊖

Time Counter Properties

Related:

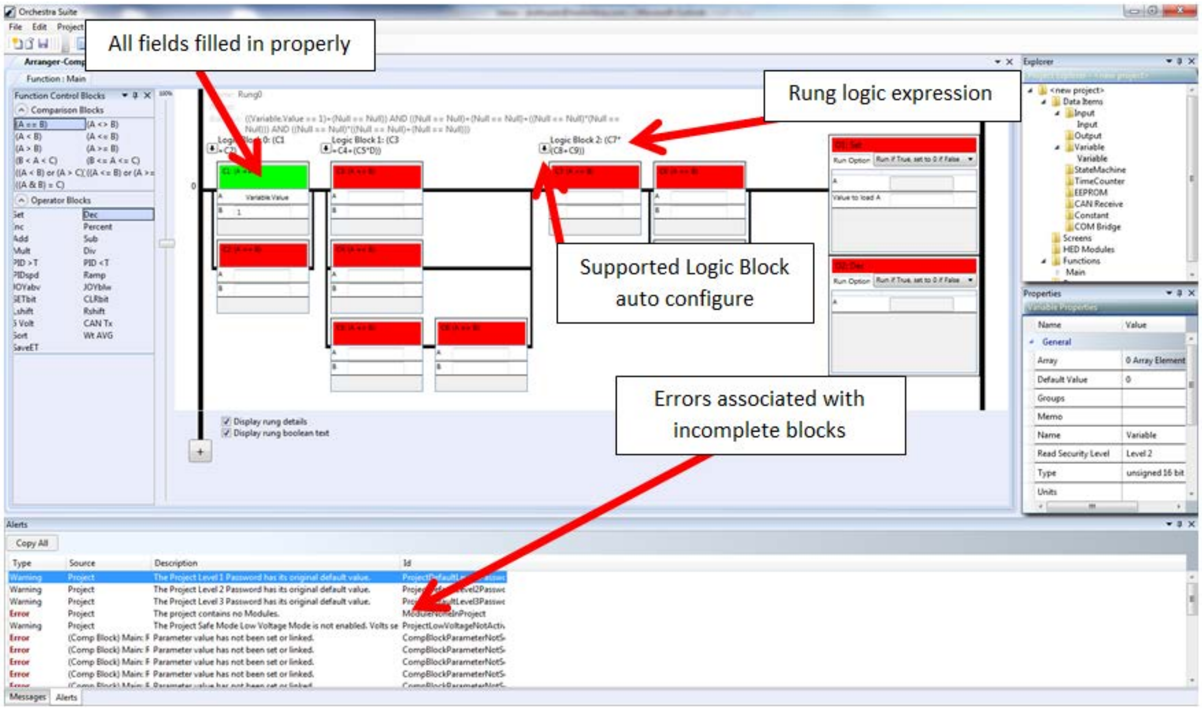
Rungs 106
Data Items 54
Adj. Loop Time 248

Rungs

6 Rungs

To code within Orchestra®, Rungs or Presto™ with Rungs must be selected as the compile option 34. Selecting one of those options activates the Functions folder in the Explorer Pane.

The rung can support up to a max of 25 Operator Blocks. Each block has a color band on it, either red or green. The block will remain red and an error will appear in the Alerts pane as long as the block does not have all of the require fields filled in properly. Once each field is filled in properly, the block will turn green and the errors will disappear from the Alerts pane.


All fields filled in properly

Rung logic expression

Supported Logic Block auto configure

Errors associated with incomplete blocks

☑ Display rung details
☑ Display rung boolean text

Copy All

Type	Source	Description	Id
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passw
Warning	Project	The Project Level 2 Password has its original default value.	ProjectDefaultLevel2Passw
Warning	Project	The Project Level 3 Password has its original default value.	ProjectDefaultLevel3Passw
Error	Project	The project contains no Modules.	ModuleNoneInProject
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActi
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS-
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS-
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS-
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS-
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS-

Messages Alerts

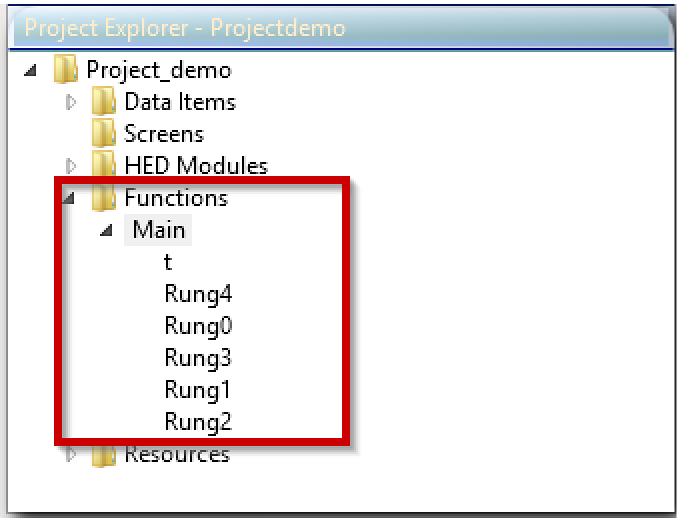
Rung Example

Related:

Coding Within Orchestra 2 324
Compile Option 35
Functions 30

6.1 Function: Main

The Functions folder is where the Main program will be stored. Each rung of the Main program is listed out under the folder for users to go to.



Project Explorer - Projectdemo

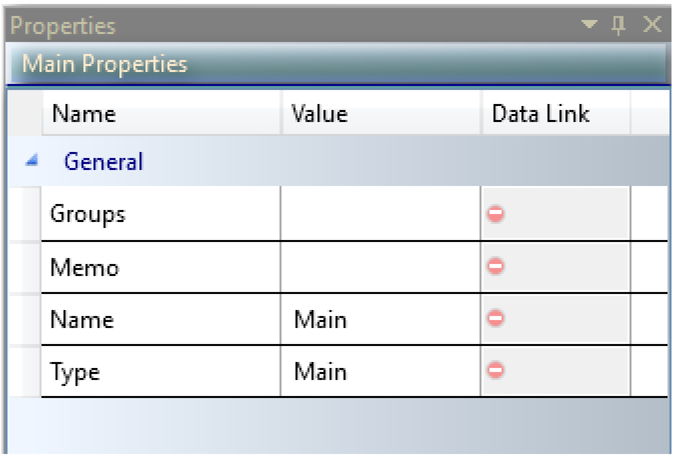
- Project_demo
  - Data Items
  - Screens
  - HED Modules
  - Functions
    - Main
      - t
      - Rung4
      - Rung0
      - Rung3
      - Rung1
      - Rung2
  - Resources

Functions: Main Folder

When a new project is created, there is no logic contained within the Function: Main folder.

Properties:

Groups: Feature is not active.
Memo: Enter a description of the function
Name: Enter a name for the function
Type: Choose Main or User Defined. User Defined option is not active.



Properties

Main Properties

Name	Value	Data Link
General		
Groups		⊖
Memo		⊖
Name	Main	⊖
Type	Main	⊖

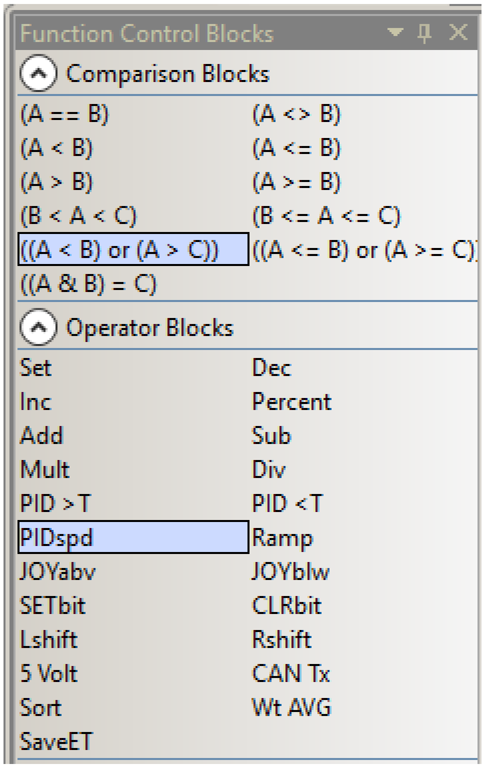
Functions: Main Properties

Related:

Functions Folder 30
Comparison Logic Blocks 109

6.2 Function Control Blocks

Orchestra® uses two kinds of blocks in its rung logic, Comparison Blocks and Operator Blocks 116. The Comparison blocks are used to create the logic for the rung program, while the Operator Blocks perform some kind of action whether or not the Comparison Block logic is true or false.



Function Control Blocks

Comparison Blocks

(A == B)	(A <> B)
(A < B)	(A <= B)
(A > B)	(A >= B)
(B < A < C)	(B <= A <= C)
((A < B) or (A > C))	((A <= B) or (A >= C)
((A & B) = C)	

Operator Blocks

Set	Dec
Inc	Percent
Add	Sub
Mult	Div
PID >T	PID <T
PIDspd	Ramp
JOYabv	JOYblw
SETbit	CLRbit
Lshift	Rshift
5 Volt	CAN Tx
Sort	Wt AVG
SaveET	

Function Control Blocks Pane

Related:

Comparison Blocks 112
Operator Blocks 117

6.2.1 Comparison Logic Blocks

The Comparison Blocks are grouped together into logic blocks. Each logic block can have up to five Comparison Blocks in each logic block, and each rung can support up to five logic blocks for a max of 25 blocks pending configuration.

Related:

Supported Configurations 110
Logic 110
Comparison Blocks 112

6.2.1.1 Logic

The rung works based on simple AND/OR logic performed on the Comparison Blocks. Based on the outcome of the logic compares, the Operator Block(s) will perform their specific action. The program performs from left to right, top to bottom; meaning the left-most Comparison Block will be performed first, and the top-most Operator Block will operate first.

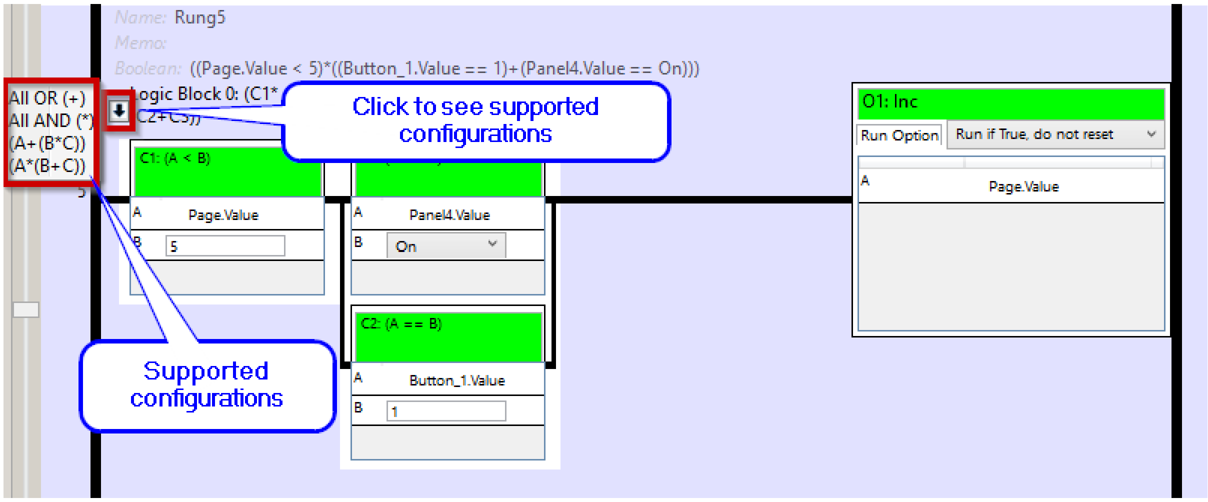
Related:

Comparison Blocks 112

Operator Blocks 117

6.2.1.2 Supported Configurations

To determine supported configurations, click the small downward arrow box to the left of the logic block.



Name: Rung5

Memo:

Boolean: ((Page.Value < 5)*((Button_1.Value == 1)+(Panel4.Value == On)))

Logic Block 0: (C1*(C2+C3))

C1: (A < B)
A Page.Value
B 5

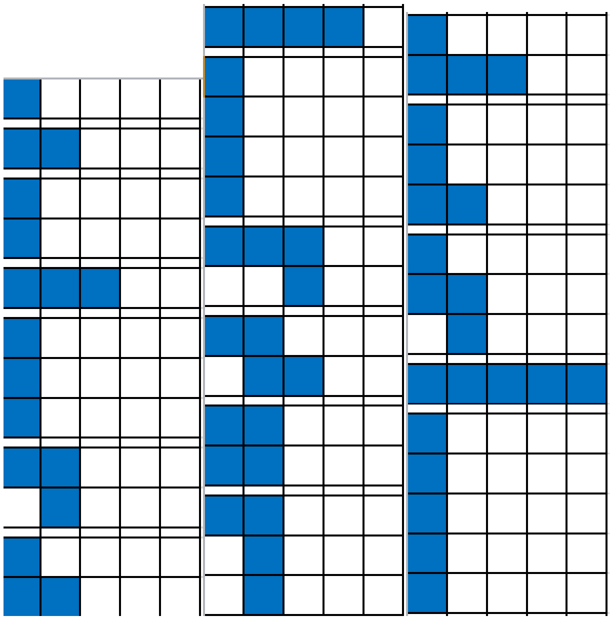
A Panel4.Value
B On

C2: (A == B)
A Button_1.Value
B 1

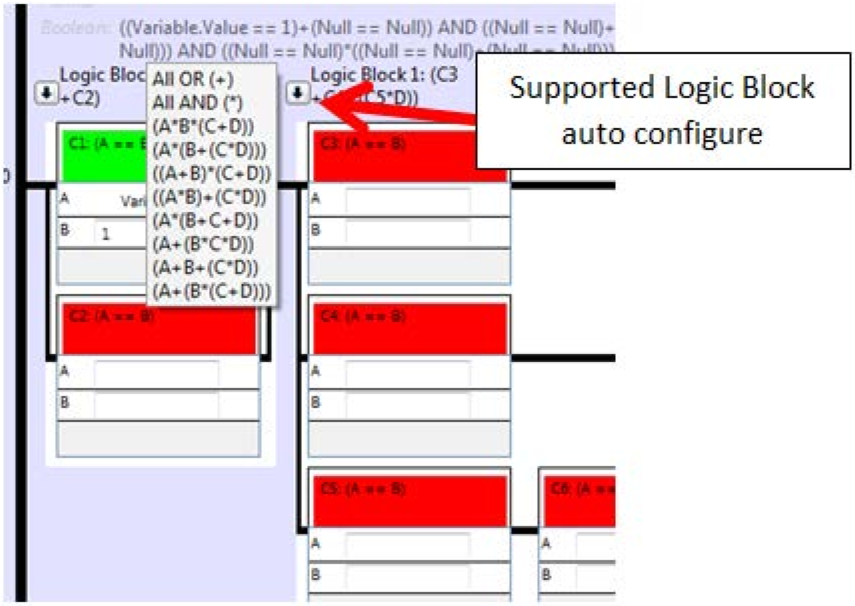
O1: Inc
Run Option: Run if True, do not reset
A Page.Value

Supported Configurations

The drop-down menu will display all supported logic variations with that number of comparison blocks within the logic block, and rearrange the blocks automatically once selected.



Supported Comparison Block Configurations



Boolean: ((Variable.Value == 1)+(Null == Null)) AND ((Null == Null)+ Null))) AND ((Null == Null)*((Null == Null)+(Null == Null)))

Logic Bloc +C2)

Logic Block 1: (C3 +C (C5*D))

C1: (A == B

A Vari

B 1

C2: (A == B)

C3: (A == B)

C4: (A == B)

C5: (A == B)

C6: (A ==

Supported Logic Block auto configure

Supported Logic Block Auto Configure Example

Related:

[Logic] 110
[Comparison Blocks] 112
[Operator Blocks] 117

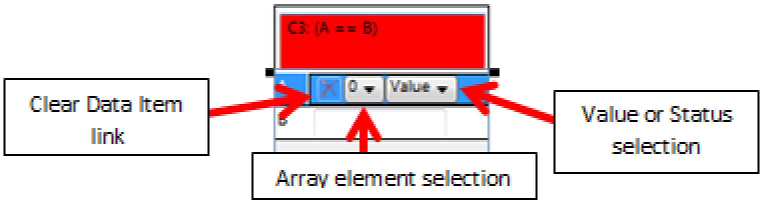
6.3 Comparison Blocks

The Comparison Blocks have two or three data fields, “A”, “B”, or “C”, that the user can fill in. The user can either fill those fields in with their own numbers or use data items, so data items and numbers can be compared to other data items or numbers.

Related:
Comparison Block Types 112

6.3.1 Using Data Item Properties with Comparison Blocks

Some data items have extra properties besides just a value of a number, such as states, indexes within arrays, sub-types, etc. that define the data item. To tell what property of the data item is being used in the comparison, Orchestra® attaches a “.Value” or “.Status” extension to the data item name within the Comparison Block. Those properties can be used for comparisons within Comparison Blocks.



Comparison Block Properties Example

The user should ensure that the values used within the Comparison Blocks fall within the range of the data item or defined value, otherwise the block may never become true and could potentially prevent that particular rung from ever executing.

Note: Purposely creating a condition in which the Comparison Block is never true can be a simple way of commenting out a rung within the program, and can be useful for troubleshooting.

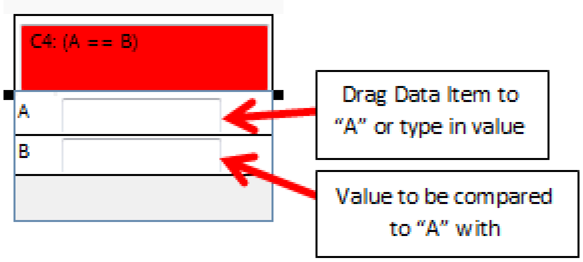
Related:
Data Items 54
Comparison Block Types 112

6.3.2 Comparison Block Types

Related:
Equals (A==B) 113
Less Than (A<B) 113
Greater Than (A>B) 114

6.3.2.1 Equals (A==B)

The Equals Comparison Block performs a logic check to determine whether or not the two values, “A” and “B”, in the block are equal or not.


C4: (A == B)
A
B
Drag Data Item to “A” or type in value
Value to be compared to “A” with

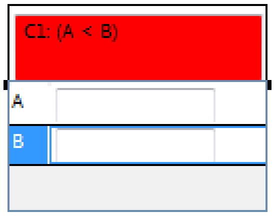
Equals Comparison Block

Related:

Not Equal (A<>B) 114
AND Equal To ((A && B) = C) 116

6.3.2.2 Less Than (A<B)

This block compares the value of what is in “A” to what is in “B”, and returns a true if the current value or status of the “A” is less than the compared value or status in “B”.



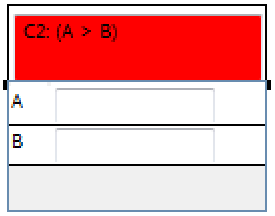
Less Than Comparison Block

Related:

Greater Than (A>B) 114
Less Than or Equal To (A<=B) 114
Less Than OR Greater Than ((A<B) or (A>C)) 116
Less Than or Equal To OR Greater Than or Equal To ((A<=B) or (A>=C)) 116

6.3.2.3 Greater Than (A>B)

This block is true if the value of the data item in “A” is greater than the defined value of “B”.

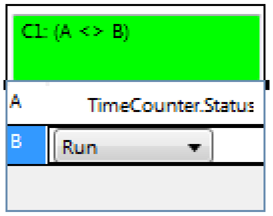


Greater Than Comparison Block

Related:
Less Than (A<B) 113
Greater Than or Equal To (A>=B) 115
Greater Than, Less Than (B<A<C) 115
Greater Than or Equal To, Less Than or Equal To (B<=A<=C) 115

6.3.2.4 Not Equal (A<>B)

This block is true as long as the item in “A” does not equal the value in “B.



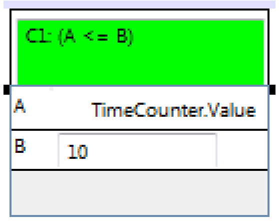
A TimeCounter.Status
B Run

Not Equal Comparison Block

Related:
Equals (A==B) 113
AND Equal To ((A && B) = C) 116

6.3.2.5 Less Than or Equal To (A<=B)

This block works the same way as the Less Than block 113 , except that the value to cause the block to return a true can include the value in “B”. The value of A must be greater than “B” for this block to be false.



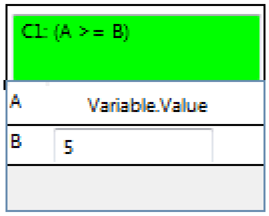
A TimeCounter.Value
B 10

Less Than or Equal To Comparison Block

Related:
Less Than (A<B) 113
Greater Than or Equal To (A>=B) 115

6.3.2.6 Greater Than or Equal To (A>=B)

This block works just like the Greater Than block 114, but the range of values that causes the block to return as true includes the value designated in “B”. Any value less than the value in “B” will cause this block to return a false.



C1: (A > = B)

A	Variable.Value
B	5

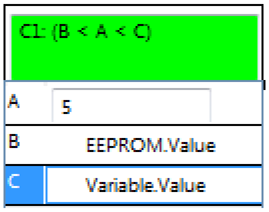
Greater Than or Equal To Comparison Block

Related:

Greater Than (A>B) 114
Less Than or Equal To (A<=B) 114

6.3.2.7 Greater Than, Less Than (B<A<C)

The Greater Than, Less Than block checks to see if the value of “A” falls in between the values set by “B” and “C”, and returns true if it does.



C1: (B < A < C)

A	5
B	EEPROM.Value
C	Variable.Value

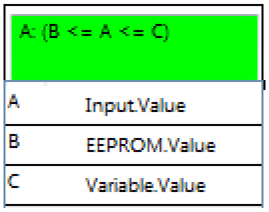
Greater Than, Less Than Comparison Block

Related:

Greater Than (A>B) 114
Less Than (A<B) 113

6.3.2.8 Greater Than or Equal To, Less Than or Equal To (B<=A<=C)

This block works by comparing the value in “A” to the other two values in “B” and “C”, and returns a true if it falls between them or equals either of them.



A: (B < = A < = C)

A	Input.Value
B	EEPROM.Value
C	Variable.Value

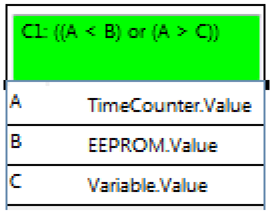
Greater Than or Equal To, Less Than or Equal To Comparison Block

Related:

Less Than or Equal To OR Greater Than or Equal To ((A<=B) or (A>=C)) 116

6.3.2.9 Less Than OR Greater Than ((A<B) or (A>C))

The value of “A” is compared to both the values of “B” and “C”, and if either “A” is less than “B” or “A” is greater than “C” this block returns true.



C1: ((A < B) or (A > C))

A	TimeCounter.Value
B	EEPROM.Value
C	Variable.Value

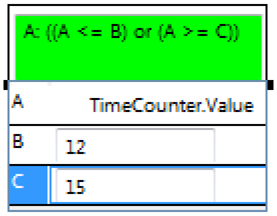
Less Than OR Greater Than Comparison Block

Related:

Less Than (A<B) 113
Greater Than (A>B) 114

6.3.2.10 Less Than or Equal To OR Greater Than or Equal To ((A<=B) or (A>=C))

If “A” is either greater than or equal to the value in “C” or is less than or equal to the value in “B”, this block returns a true.



A: ((A <= B) or (A >= C))

A	TimeCounter.Value
B	12
C	15

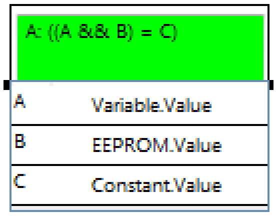
Less Than or Equal To OR Greater Than or Equal To Comparison Block

Related:

Greater Than or Equal To, Less Than or Equal To (B<=A<=C) 115

6.3.2.11 AND Equal To ((A && B) = C)

This block performs a bit-wise AND operation between the value in “A” and the value in “B”, and then compares the result to “C”. If the result is equal to the value of “C” the block returns a true.



A: ((A && B) = C)

A	Variable.Value
B	EEPROM.Value
C	Constant.Value

AND Equal To Comparison Block

Related:

Equals (A==B) 113

6.4 Operator Blocks

Operator Blocks appear on the right hand side of the rung and perform a specific action if all of the logic from the Comparison Blocks is true. The Operator Blocks can accept a combination of data items, predefined values from drop-down menus, and user defined values within the data fields where permissible.

Some areas can only accept data items, and those areas can be identified by being unable to click on them and have a cursor appear to input user values. Data items function in the Operator Blocks similarly to the way they function within the Comparison Blocks 106, in that particular properties or portions of the data item can be selected to perform the operation.

Related:
Comparison Logic Blocks 109
Data Items 54

6.4.1 Run Options

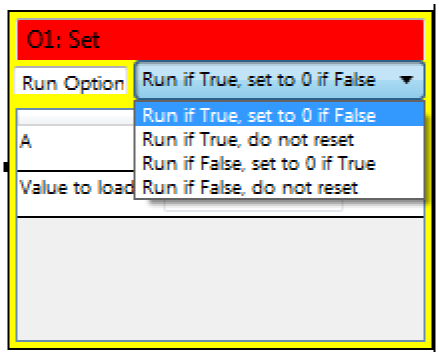
Each Operator Block, with the exception of three, has a Run Option field at the top of the block that contains four possible options; Run if True set to 0 if False, Run if True do not reset, Run if False set to 0 if True, and Run if False do not reset.

The three that do not have that option will be explained within their respective sections following.

The **Run if True set to 0 if False** option will allow that Operator Block to only execute if the rung logic is true. If the rung logic is false, that Operator Block will produce a value of zero.

Run if True do not reset will execute the operation if the is true and whatever the outcome of the operation is; that value will be held until that data item or user defined value is operated upon again.

The **Run if False set to 0 if True and Run if False** do not reset work in the same manner as their Run if True counterparts.



O1: Set

Run Option: Run if True, set to 0 if False

Run if True, set to 0 if False
Run if True, do not reset
Run if False, set to 0 if True
Run if False, do not reset
A
Value to load

Operator Block Run Options

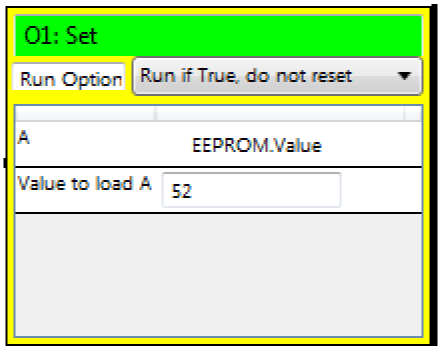
Related:
Operator Block Types 118
Data Items 54

6.4.2 Operator Block Types

Related:
Set 119
Dec 119
Inc 120
Percent 120
Add 121
Sub 121
Mult 122
Div 122
PID>T 123
PID<T 124
PIDspd 125
Ramp 126
JOYabv 126
JOYblw 127
SETbit 128
CLRbit 129
Lshift 130
Rshift 130
5 Volt 131
CAN Tx 131
Sort 133
Wt AVG 133
SaveET 134
Comparison Block Types 112

6.4.2.1 Set

The Set Operator Block requires a data item in “A”, while the “Value to load A” can accept a data item or a user defined value. If the run logic is true, and depending on what the run option is for the Set block, the value of the data item in “A” will change to the “Value to load A”.


O1: Set

Run Option: Run if True, do not reset

A: EEPROM.Value

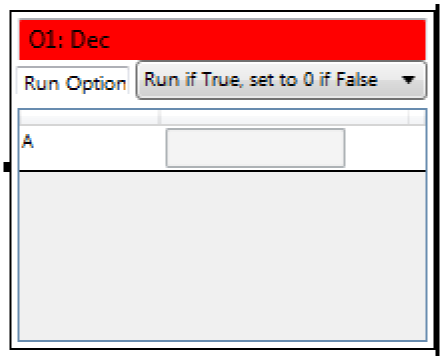
Value to load A: 52

Set Operator Block

Related:
Data Items 54

6.4.2.2 Dec

The data item that gets assigned to “A” in the Dec block will have its value decremented by one whenever the rung logic satisfies the run option selected for the Operator Block. The decrement will only occur once for each time the rung is true, so to decrement multiple times, the rung must transition from true to false and back to true.


O1: Dec

Run Option: Run if True, set to 0 if False

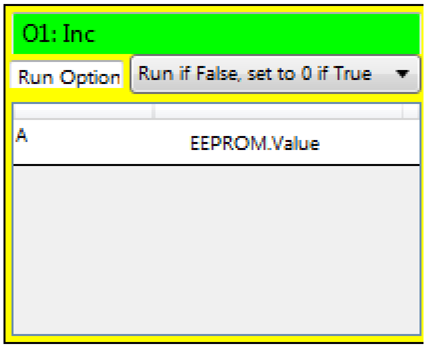
A:

Dec Operator Block

Related:
Run Options 117
Logic 110

6.4.2.3 Inc

The Inc block will increment the value of the data item in “A” by one if the rung logic fulfills the run option requirement chosen. Similar to the Dec block 119, this operation will only occur once for each transition to a true state from a false state.


01: Inc

Run Option: Run if False, set to 0 if True

A	EEPROM.Value

Inc Operator Block

Related:

Dec 119

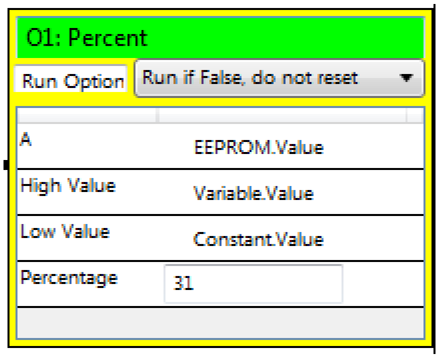
Logic 110

6.4.2.4 Percent

The Percent Operator Block will return a value based on a user-or-data item-specified percentage over a specific range of values whenever the run option for the block is met, using the following formula:

$$A = (High\ Value - Low\ Value) \times Percentage + Low\ Value$$

Equation: Percent Operator Block Equation


01: Percent

Run Option: Run if False, do not reset

A	EEPROM.Value
High Value	Variable.Value
Low Value	Constant.Value
Percentage	31

Percent Operator Block

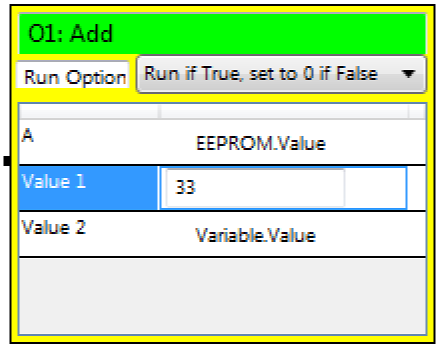
Note: Within Orchestra® percentages are from 0 to 1000, where 1000 is equal to 100%.

Related:

Data Items 54

6.4.2.5 Add

The Add block takes two values, adds them together, and places the result in the data item specified in “A” each time the rung meets the run option criteria. If the criteria is met, each time the program loops the addition will take place.



01: Add

Run Option: Run if True, set to 0 if False

A	EEPROM.Value
Value 1	33
Value 2	Variable.Value

Add Operator Block

$$A = (Value\ 1 + Value\ 2)$$

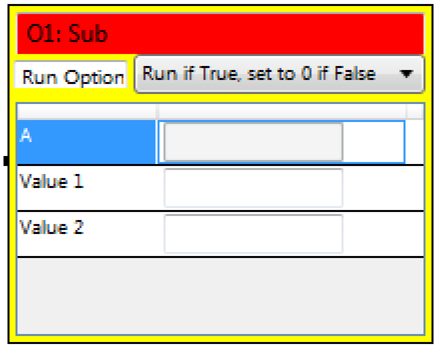
Equation: Add Operator Block Equation

Related:

[Data Items 54]

6.4.2.6 Sub

The Sub block will subtract “Value 2” from “Value 1” and place the result in “A” whenever the run option criterion is met.



01: Sub

Run Option: Run if True, set to 0 if False

A	
Value 1	
Value 2	

Sub Operator Block

Like the Add block 120, if this rung criteria is met, then each time the program loops a subtraction occurs.

$$A = (Value\ 1 - Value\ 2)$$

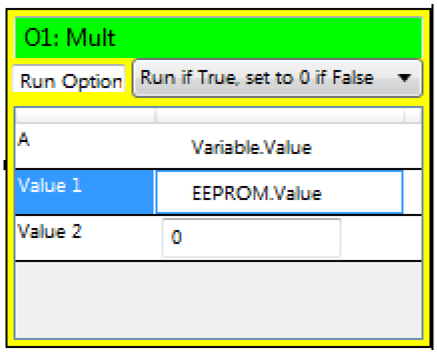
Equation: Sub Operator Block Equation

Related:

Add 121

6.4.2.7 Mult

Mult will multiply “Value 1” and “Value 2” together, and place the resulting value in the data item assigned to “A” whenever the rung logic and the run option of the block coincide. This block will continue to execute each time the program loops as long as the rung is held true.



01: Mult

Run Option: Run if True, set to 0 if False

A	Variable.Value
Value 1	EEPROM.Value
Value 2	0

Mult Operator Block

$$A = (Value\ 1 \times Value\ 2)$$

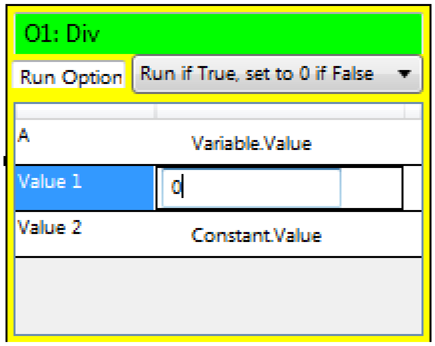
Equation: Mult Operator Block Equation

Related:

Run Options 117
Logic 110
Data Items 54

6.4.2.8 Div

This block will divide the value in “Value 1” by the value in “Value 2” then place the resulting value in “A” if the run option is met. This block will continue to execute each time the program loops if the rung is held true.



01: Div

Run Option: Run if True, set to 0 if False

A	Variable.Value
Value 1	0
Value 2	Constant.Value

Div Operator Block

$$A = (Value\ 1 \div Value\ 2)$$

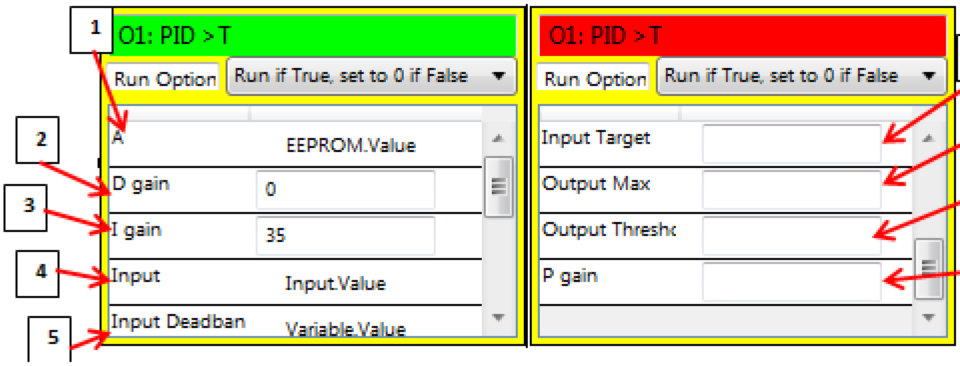
Equation: Div Operator Block Equation

Related:

Run Options 117
Logic 110

6.4.2.9 PID>T

This block is used to provide closed loop control for an output using the error correction from a PID operation as long as target value is less than the input value, i.e. the lowering of a robotic arm on a refuse vehicle to its resting position after being raised.



O1: PID > T

Run Option: Run if True, set to 0 if False

A	EEPROM.Value
D gain	0
I gain	35
Input	Input.Value
Input Deadban	Variable.Value

O1: PID > T

Run Option: Run if True, set to 0 if False

Input Target	
Output Max	
Output Threshc	
P gain	

PID>T Operator Block

Properties

1. A 248
2. D gain 249
3. I gain 253
4. Input 254
5. Input Deadband 254
6. Input Target 254
7. Output Max 256
8. Output Threshold 256
9. P gain 257

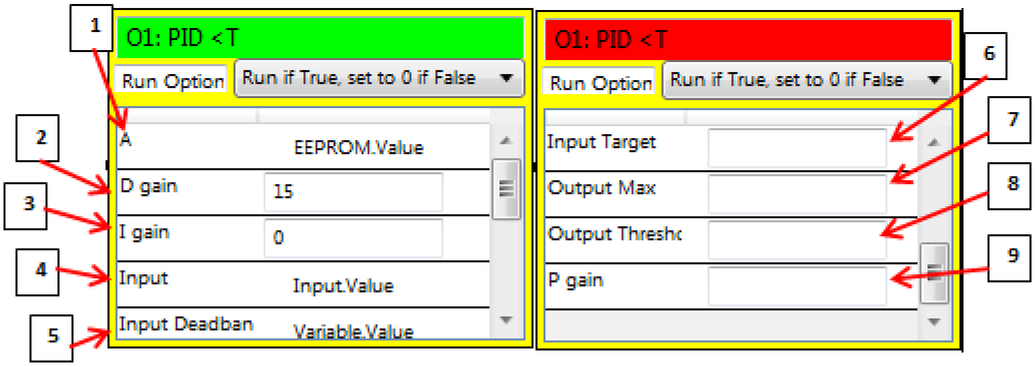
Related:

PID<T 124
PIDspd 125

6.4.2.10 PID<T

This block is used to provide closed loop control for an output using the error correction from a PID operation as long as the target value is greater than the input value, i.e. the raising of a robotic arm on a refuse vehicle to some point above its resting position.

Note: In most cases the PID<T and PID>T blocks are used in conjunction to control an output, such as raising and lowering of a robotic arm in a smooth and controlled manner. The output will shut off once the target is reached.



O1: PID <T

Run Option: Run if True, set to 0 if False

A	EEPROM.Value
D gain	15
I gain	0
Input	Input.Value
Input Deadban	Variable.Value

O1: PID <T

Run Option: Run if True, set to 0 if False

Input Target	
Output Max	
Output Threshc	
P gain	

PID<T Operator Block

Properties

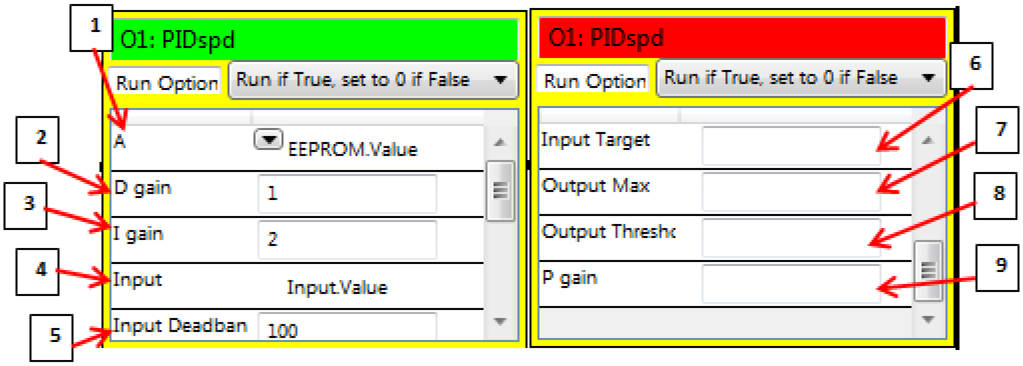
1. A 248
2. D gain 249
3. I gain 253
4. Input 254
5. Input Deadband 254
6. Input Target 254
7. Output Max 256
8. Output Threshold 256
9. P gain 257

Related:

PID>T 123
PIDspd 125

6.4.2.11 PIDspd

This block is used to provide closed loop control for an output using the error correction from a PID operation as long as the target value is not equal to the input value, i.e. cruise control within a vehicle. Based on the outcome of the operation, an Output will be driven accordingly to maintain a target value.


O1: PIDspd

Run Option	Run if True, set to 0 if False
A	EEPROM.Value
D gain	1
I gain	2
Input	Input.Value
Input Deadban	100

O1: PIDspd

Run Option	Run if True, set to 0 if False
Input Target	
Output Max	
Output Threshc	
P gain	

PIDspd Operator Block

Properties

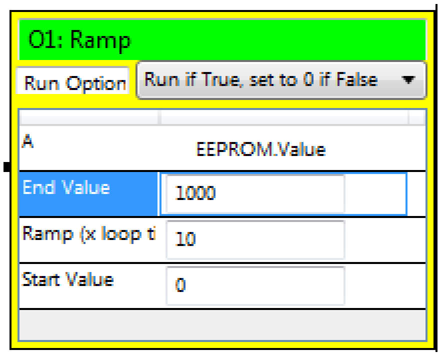
1. A 248
2. D gain 249
3. I gain 253
4. Input 254
5. Input Deadband 254
6. Input Target 254
7. Output Max 256
8. Output Threshold 256
9. P gain 257

Related:

PID>T 123
PID<T 124

6.4.2.12 Ramp

The Ramp Operator Block will gradually change the value passed to “A” from the value entered into the “Start Value” to the “End Value”. How gradually it changes is controlled by the value assigned to the “Ramp” data field. Each loop through the program will change the value in “A” by the “Ramp” amount up to the “End Value” as long as the run option criterion is met.



O1: Ramp

Run Option: Run if True, set to 0 if False

A	EEPROM.Value
End Value	1000
Ramp (x loop ti	10
Start Value	0

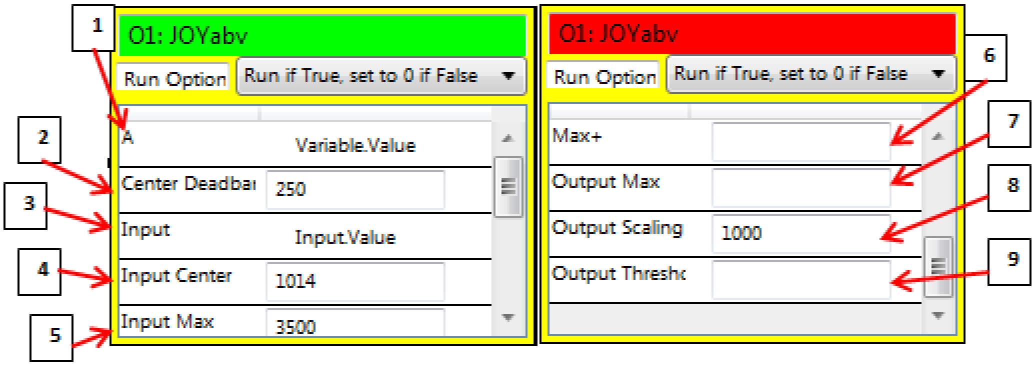
Ramp Operator Block

Related:

Run Options 117

6.4.2.13 JOYabv

The JOYabv block converts an Input value into a linearly proportional Output value when the Input value is above the defined Input Center.



O1: JOYabv

Run Option: Run if True, set to 0 if False

A	Variable.Value
Center Deadbai	250
Input	Input.Value
Input Center	1014
Input Max	3500

O1: JOYabv

Run Option: Run if True, set to 0 if False

Max+	
Output Max	
Output Scaling	1000
Output Threshc	

JOYabv Operator Block

Properties

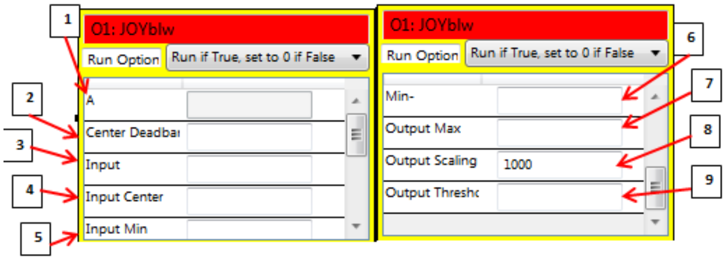
1. A 248
2. Center Deadband 248
3. Input 254
4. Input Center 254
5. Input Max 254
6. Max+ 255
7. Output Max 256
8. Output Scaling 256

9. Output Threshold 256

Related:
JOYblw 127
Using JOYabv and JOYblw 127

6.4.2.14 JOYblw

The JOYblw block converts an Input value into a linearly proportional Output value when the Input value is below the defined Input Center.



O1: JOYblw

Run Option: Run if True, set to 0 if False

A	
Center Deadbar	
Input	
Input Center	
Input Min	
Min-	
Output Max	
Output Scaling	1000
Output Threshc	

JOYblw Operator Block

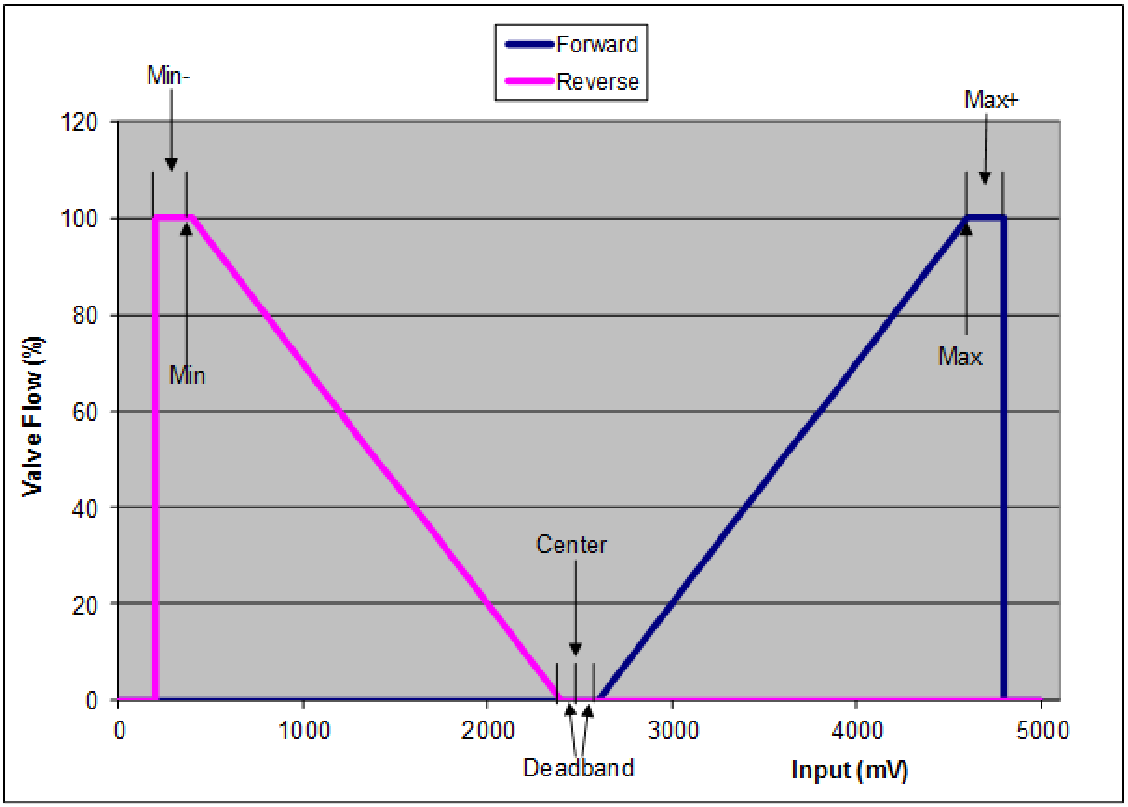
Properties

1. A 248
2. Center Deadband 248
3. Input 254
4. Input Center 254
5. Input Min 254
6. Min- 255
7. Output Max 256
8. Output Scaling 256
9. Output Threshold 256

Related:
Using JOYabv and JOYblw 127
JOYabv 126

6.4.2.14.1 Using JOYabv and JOYblw

The JOYabv and JOYblw are usually used together in a setting that would need an output to be linearly driven by an input value, such as a joystick. As the position input of the joystick increases the output increases proportionally to possibly open something (JOYabv), while the decreasing of the joystick position past the center point would increase an output (JOYblw) that could possibly close what was opened.



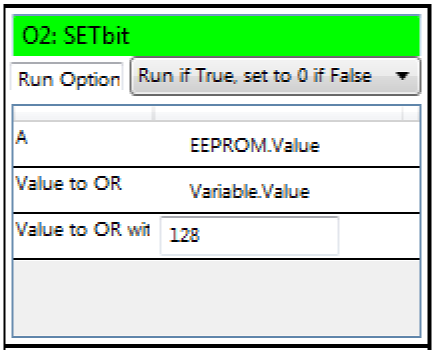
Graphical Reference of JOYabv and JOYblw Properties

Related:

[JOYabv 126]
[JOYblw 127]

6.4.2.15 SETbit

The SETbit block performs a bitwise OR operation on the value in the “Value to OR” field with a value in the “Value to OR with”. The resulting value is placed in “A” if the rung logic satisfies the run option of the block.



O2: SETbit

Run Option: Run if True, set to 0 if False

A	EEPROM.Value
Value to OR	Variable.Value
Value to OR wit	128

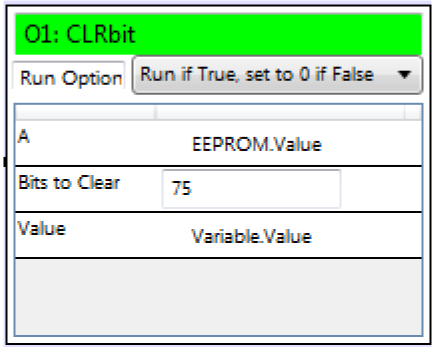
SETbit Operator Block

Related:

[Run Options 117]
[Logic 110]
[CLRbit 129]

6.4.2.16 CLRbit

This block will use the “Bits to Clear” value as a mask to toggle the selected high bits (1) in the “Value” data field to low (0) if the run option is met, and place the resulting value in “A”.



O1: CLRbit

Run Option: Run if True, set to 0 if False

A	EEPROM.Value
Bits to Clear	75
Value	Variable.Value

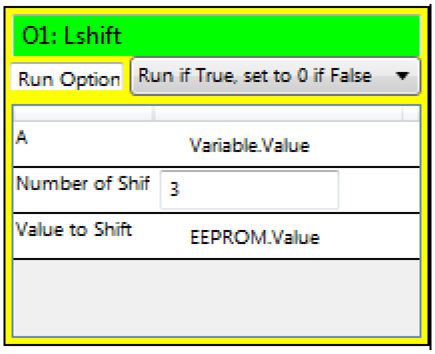
CLRbit Operator Block

Related:

SETbit 128
Run Options 117

6.4.2.17 Lshift

The Lshift Operator Block performs a logical shift left on the value in the “Value to Shift” field by the number of times of the value in “Number of Shifts” field and places the result in “A” if the run option is met.



01: Lshift

Run Option: Run if True, set to 0 if False

A	Variable.Value
Number of Shif	3
Value to Shift	EEPROM.Value

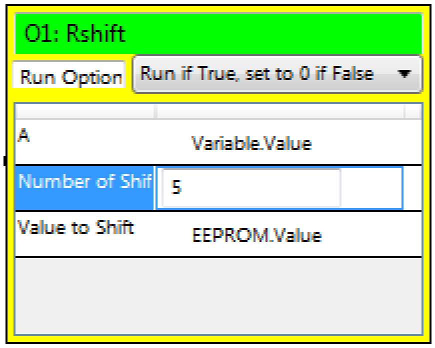
Lshift Operator Block

Related:

[Rshift] 130
[Run Options] 117

6.4.2.18 Rshift

The Rshift Operator Block performs a logical shift right on the value in the “Value to Shift” field by the number of times of the value in “Number of Shifts” field and places the result in “A” if the run option is met.



01: Rshift

Run Option: Run if True, set to 0 if False

A	Variable.Value
Number of Shif	5
Value to Shift	EEPROM.Value

Rshift Operator Block

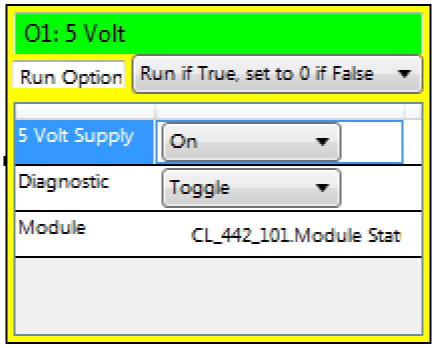
Related:

[Run Options] 117
[Lshift] 130

6.4.2.19 5 Volt

This Operator Block will turn the 5 Volt supply within the module on or off when the run option is met. The user also has the ability to select whether or not they would like to turn a diagnostic on and off, or toggle the diagnostic on and off.

The diagnostic uses a pull up resistor to monitor and report back whether or not the output is shorted to battery, ground, open when closed expected, etc. The toggle feature will turn the diagnostic on for 500ms then off for 500ms and repeat.



O1: 5 Volt

Run Option: Run if True, set to 0 if False

5 Volt Supply	On
Diagnostic	Toggle
Module	CL_442_101.Module Stat

5 Volt Operator Block

Related:

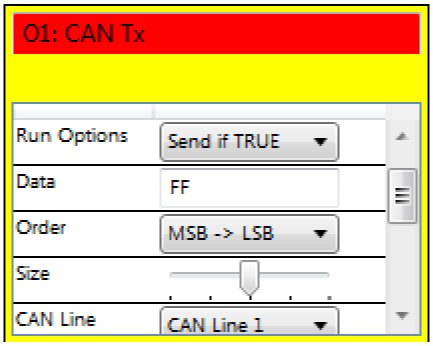
Run Options 117

6.4.2.20 CAN Tx

The CAN Tx block transmits a user defined CAN message from the selected module over the chosen CAN channel whenever the run option is met. The message can have either a hex or decimal format and can be a defined data length of up to 8 bytes.

With the addition of each byte, the user can choose the size of the data within the message i.e. a defined length of 3 bytes can have 3 separate 1 byte data values or 1 2 byte and a 1 byte value. The data can have maximum of 1 8 byte value, 2 4 byte values, and so on, down to 8 1 byte values. If the data is greater than 1 byte, the user must define the “Order” of the bytes from MSB->LSB or LSB->MSB.

The final two fields within the block are the “ID” and “ID Size” fields. The “ID Size” can be 11-bit or 29-bit and defines the size of the identifier for the CAN message that the user specifies.



O1: CAN Tx

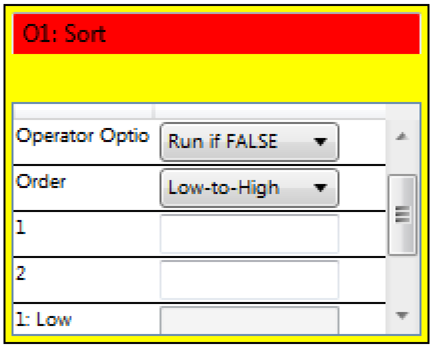
Run Options	Send if TRUE
Data	FF
Order	MSB -> LSB
Size	
CAN Line	CAN Line 1

CAN Tx Operator Block

Related:
Run Options 117

6.4.2.21 Sort

The Sort block is another one of the few blocks that do not have a “Run Option” and instead has an “Operator Option”. That option only allows the block to execute if the rung is either True or False, depending on what the user chooses. This block can take up to four values and sort them in either a highest to lowest or lowest to highest order. The sorted order is placed into corresponding data items within the respective fields.



O1: Sort

Operator Optio: Run if FALSE

Order: Low-to-High

1

2

1: Low

Sort Operator Block

Related:

Logic 110

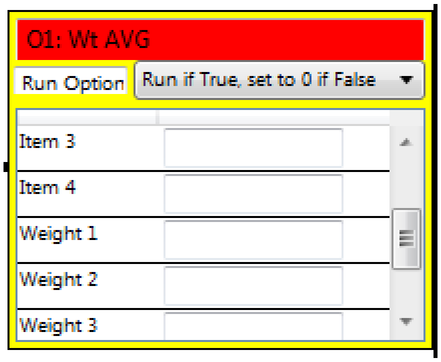
Data Items 54

6.4.2.22 Wt AVG

This block computes a weighted average of up to four values and four different weights, then places the average in the data item within the “AVG stored here” data field each time the rung logic meets the run option criteria. Each “Item” has a corresponding “Weight” and is utilized in the following way:

$$Wt\ AVG = (Item1 \times Weight1 + Item2 \times Weight2 + Item3 \times Weight3 + Item4 \times Weight4) \div Sum\ of\ weights$$

Equation: Weighted Average Equation



O1: Wt AVG

Run Option: Run if True, set to 0 if False

Item 3

Item 4

Weight 1

Weight 2

Weight 3

Wt AVG Operator Block

Related:

Data Items 54

Logic 110

6.4.2.23 SaveET

The SaveET block is a block without a “Run Option” field. Instead, there is an “Operator Option” field that allows the user to select a “Save if True” or “Save if False” option that dictates when the Operator Block will run.

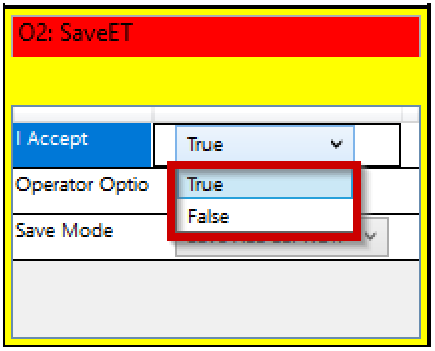
Related:

Using EEPROMS and Timers with Save ET 134
Save Mode 135

6.4.2.23.1 Using EEPROMS and Timers with Save ET

This operator block takes the temporarily stored EEPROM 101 and Timer values 103 and stores them to the processor's non-volatile memory locations. This storing process could cause the system to have an unknown delay in its ability to execute its logic and timing capability (extended loop time).

In order to activate this block, the user must change the “I Accept” field to True and agree to the terms stated previously.



O2: SaveET

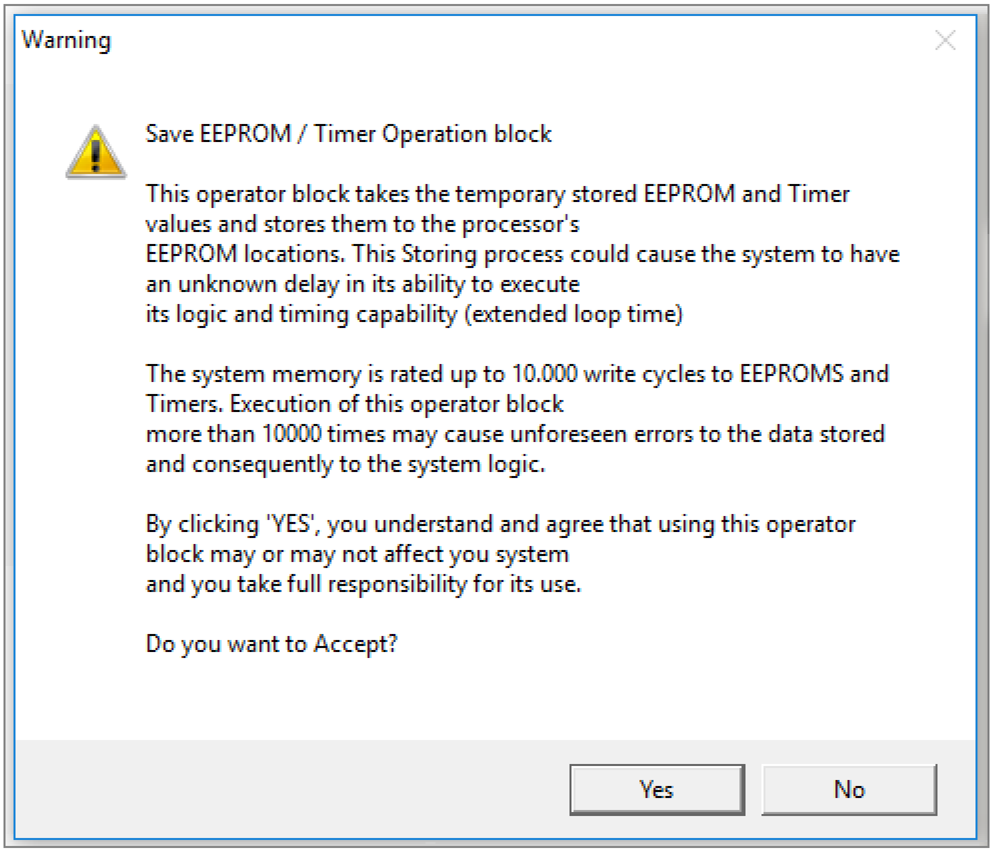
I Accept: True

Operator Optio

Save Mode

I Accept Field

The system memory is rated up to 10,000 write cycles for EEPROMS and Timers. Execution of this operator block more than 10,000 times may cause unforeseen errors to the data stored and consequently to the system logic.



Save EEPROM / Timer Operation block

This operator block takes the temporary stored EEPROM and Timer values and stores them to the processor's
EEPROM locations. This Storing process could cause the system to have an unknown delay in its ability to execute
its logic and timing capability (extended loop time)

The system memory is rated up to 10.000 write cycles to EEPROMS and Timers. Execution of this operator block
more than 10000 times may cause unforeseen errors to the data stored and consequently to the system logic.

By clicking 'YES', you understand and agree that using this operator block may or may not affect you system
and you take full responsibility for its use.

Do you want to Accept?

Save EEPROM/Timer Operation Block Warning

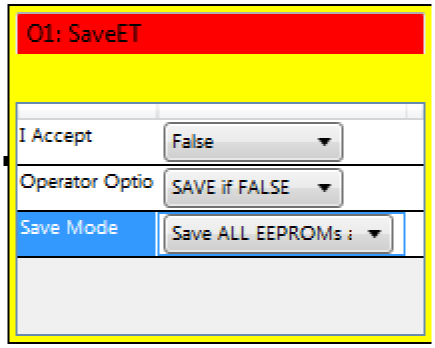
Related:
SaveET 134
EEPROM 101

6.4.2.23.2 Save Mode

EEPROMS 101 and Timers 103 each have a Save Mode. When that option is selected, this Operator Block is what performs the actual save. There are five options for saving:

1. Save ALL EEPROMs with feature enabled
2. Save ALL Timers with feature enabled
3. Save ALL EEPROMs and Timers with feature enabled
4. Save single EEPROM
5. Save single Timer

If either of the save single options are selected, the user will have to designate which one to save by dragging that data item to the “Select” field.



O1: SaveET

I Accept	False
Operator Optio	SAVE if FALSE
Save Mode	Save ALL EEPROMs a

SaveET Operator Block

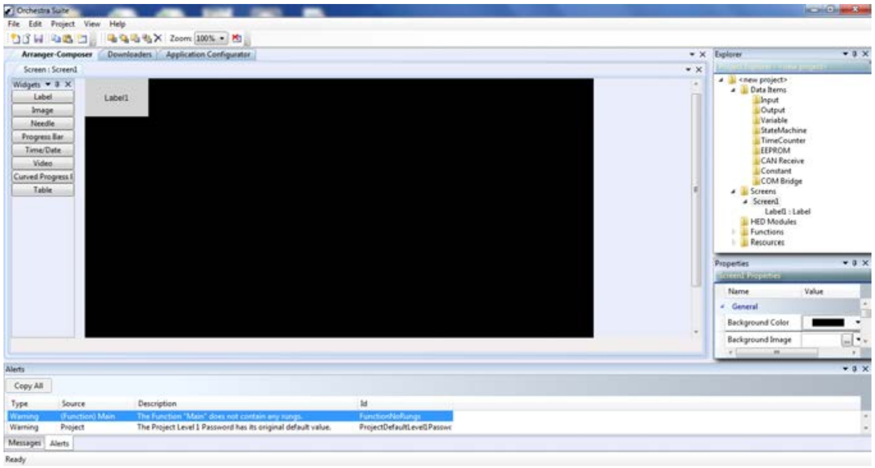
Related:

SaveET 134
Using EEPROMS and Timers with Save ET 134
EEPROM 101
Time Counter 103

Arranger™

7 Arranger™

The Arranger™ tool enables the user to create screens by placing the available widgets on a blank screen. These screens are then downloaded to a display that conveys the desired information to the outside world.



Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc

Arranger™ Panel

After the user has programmed the logic to collect the data, they can use Arranger™ to choose which widgets will show the data on the module screen.

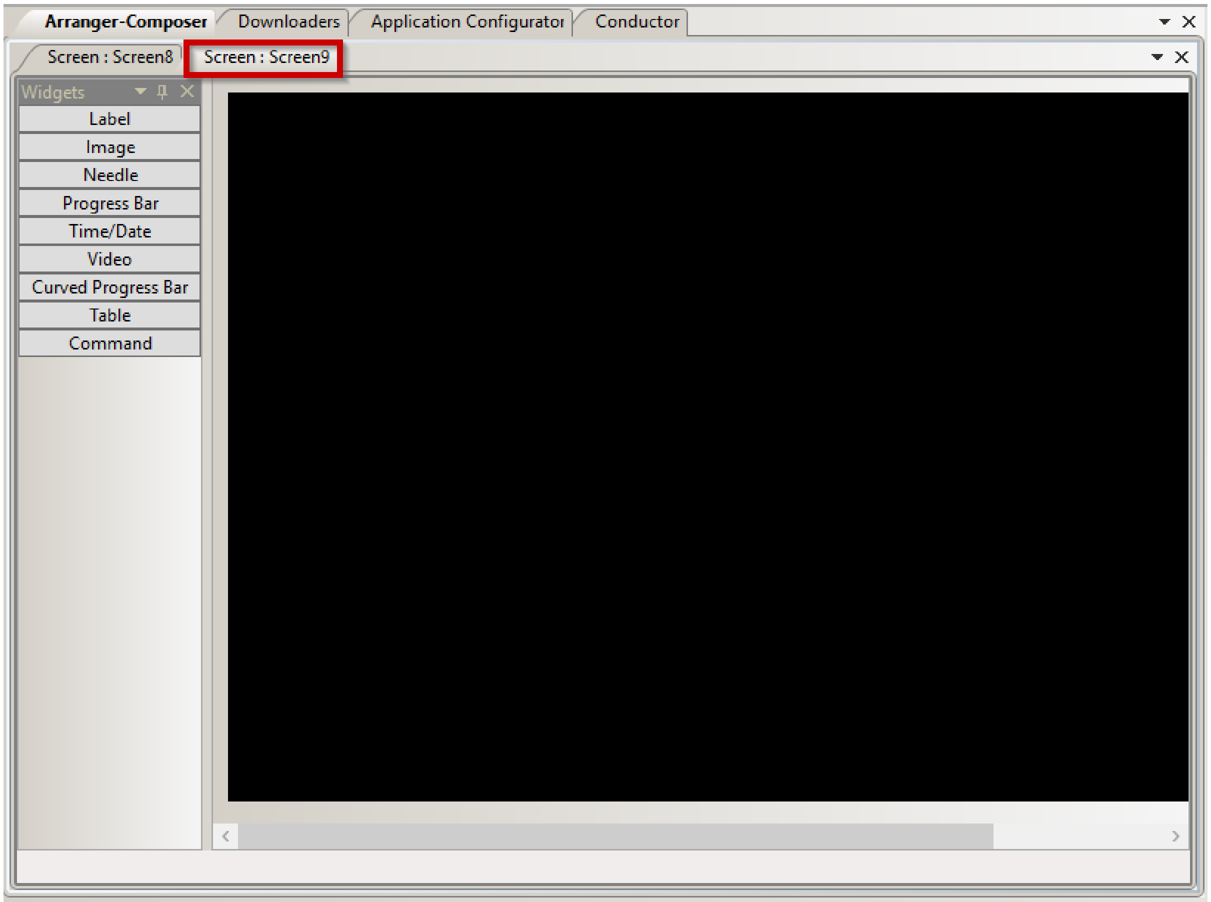
Related:

Arranger Screen 139
Using Arranger Widgets 141

7.1 Arranger™ Screen

A screen in your project is a container for the set of graphical elements, also referred to as widgets, that you want shown simultaneously on one or more of the Display Modules in your vehicle.

Screens are added to your project, and configured independently from the process of assigning Screens to specific Display Modules. This gives you the flexibility to start designing your screens before you have defined which specific modules will be on your vehicle, and also enables you to reuse any screen by later assigning it to multiple Display Modules on the vehicle.



Arranger™ Screen

Properties

1. Background Color 248
2. Background Image 248
3. Color Depth 249
4. Display Index 263
5. Display Model 251
6. Name 248
7. Resolution 248

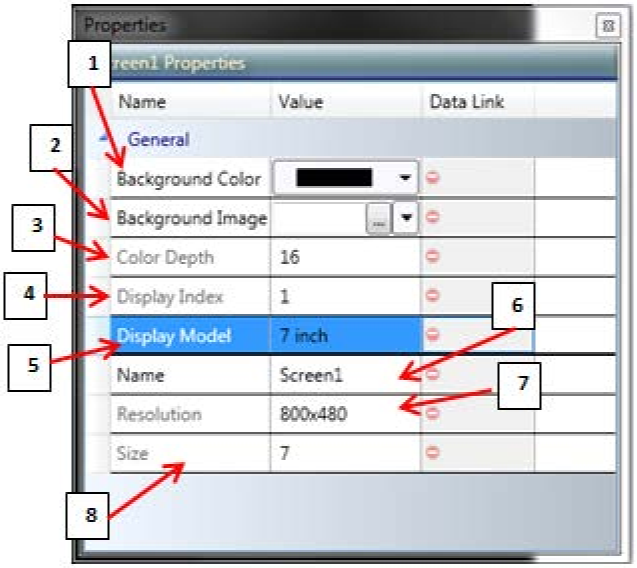
8. Size 248

Related:

Using Arranger Widgets 141
Add Widgets 322

7.2 Using Arranger™ Widgets

To begin using Arranger™, right click the “Screens” folder on the Explorer Panel and create a new screen. Upon creating a new screen, the user will see a black screen with a “Widgets” tool panel to the left of it. Highlight the newly created screen in the Explorer Panel to switch the properties panel to the properties associated with the screen.



Name	Value	Data Link
General		
Background Color	■	⊖
Background Image	...	⊖
Color Depth	16	⊖
Display Index	1	⊖
Display Model	7 inch	⊖
Name	Screen1	⊖
Resolution	800x480	⊖
Size	7	⊖

Screen Properties

Properties

1. Background Color 248
2. Background Image 248
3. Color Depth 249
4. Display Index 251
5. Display Model 251
6. Name 255
7. Resolution 257
8. Size 258

Some properties cannot be changed, but their value is useful for you, so they are displayed in a disabled state. For example, when you change the Display Model property for your screen, the values for Color Depth, Resolution, and Size change to reflect attributes of the physical hardware that you are targeting your screen for.

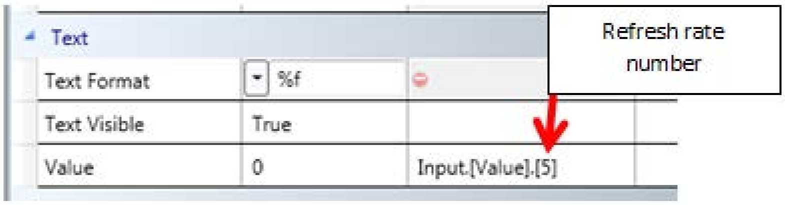
Once the properties are set to their desired values, the user can begin editing the screen by adding widgets.

Related:

Arranger Screen 139
Add Widgets 322

7.2.1 Change Update Rate

When using data items with properties, a small number next to the link represents the rate, in 100 ms increments, at which the display will request an update of that link.



Text		
Text Format	%f	
Text Visible	True	
Value	0	Input.[Value].[5]

Data Link Refresh Rate

The number can be changed by double clicking on the link, highlighting the number, and typing in a new one.

Changing this number can help with improving response time of changing values on the display, but will use more processing power.

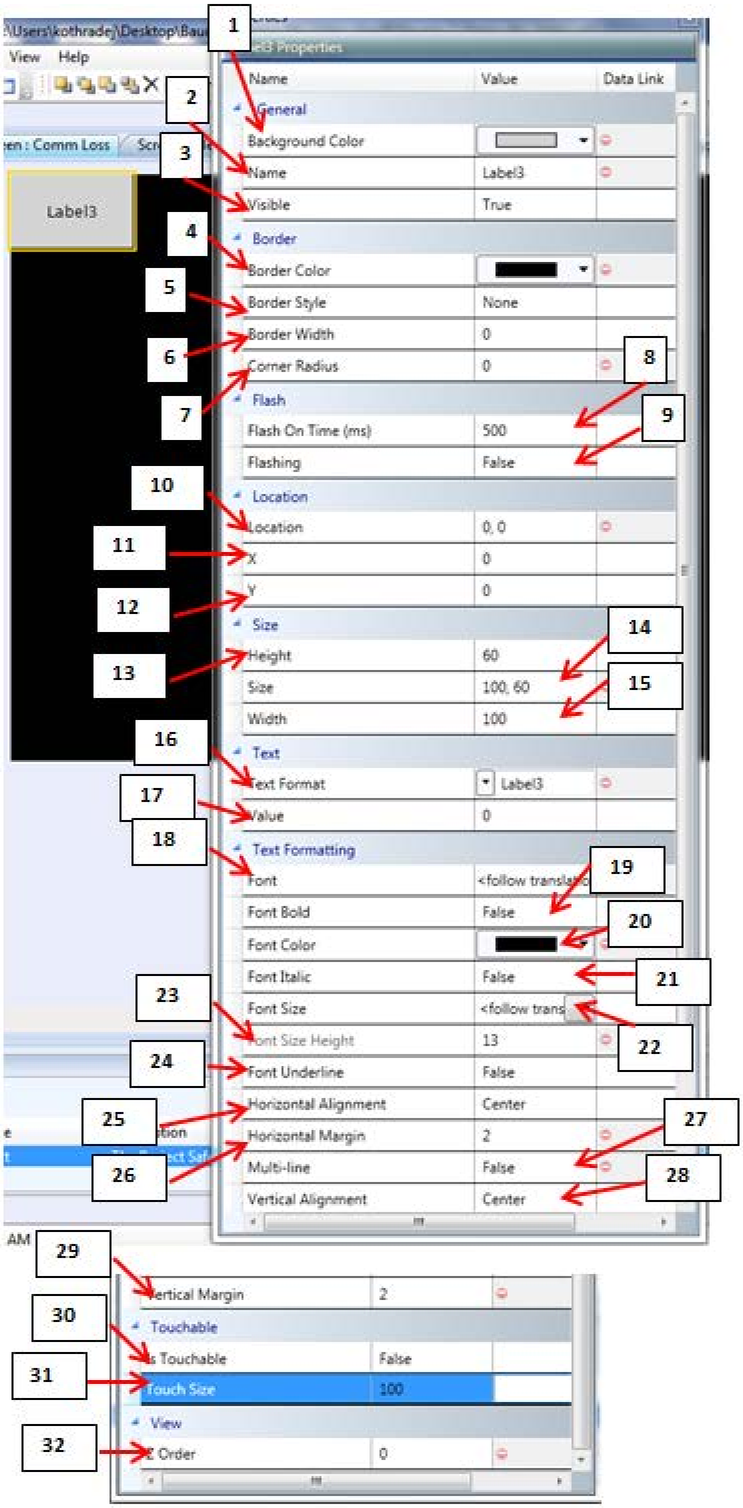
Related:

Using Arranger Widgets 141

7.3 Label

The Label widget allows the user to display an alpha numeric note or message.

Labels are fixed text, and are able to be combined with a data item value, and interchangeable language translations.



Name	Value	Data Link
General		
Background Color		
Name	Label3	
Visible	True	
Border		
Border Color		
Border Style	None	
Border Width	0	
Corner Radius	0	
Flash		
Flash On Time (ms)	500	
Flashing	False	
Location		
Location	0, 0	
X	0	
Y	0	
Size		
Height	60	
Size	100, 60	
Width	100	
Text		
Text Format	Label3	
Value	0	
Text Formatting		
Font	<follow translatio	
Font Bold	False	
Font Color		
Font Italic	False	
Font Size	<follow trans	
Font Size Height	13	
Font Underline	False	
Horizontal Alignment	Center	
Horizontal Margin	2	
Multi-line	False	
Vertical Alignment	Center	
Vertical Margin	2	
Touchable		
Is Touchable	False	
Touch Size	100	
View		
Z Order	0	

Label Properties

Properties

1. Background Color 248

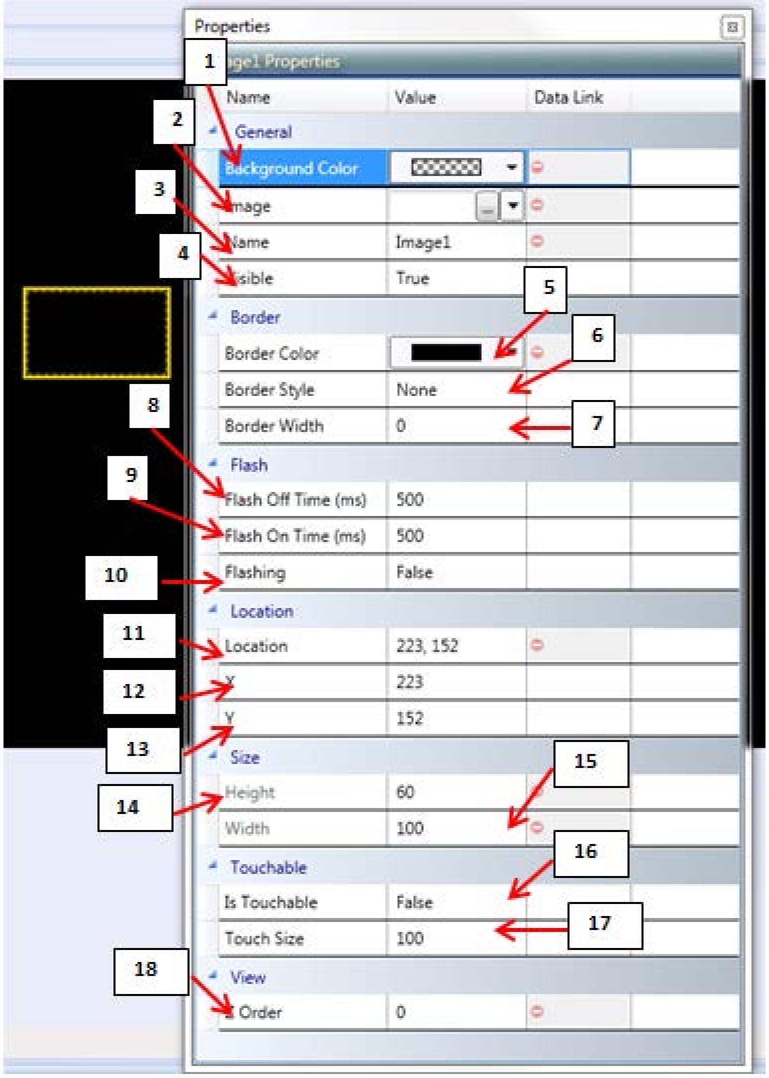
2. Name 255
3. Visible 260
4. Border Color 248
5. Border Style 248
6. Border Width 248
7. Corner Radius 249
8. Flash On Time (ms) 252
9. Flashing 252
10. Location 254
11. X 260
12. Y 260
13. Height 253
14. Size 258
15. Width 260
16. Text Format 259
17. Value 260
18. Font 252
19. Font Bold 252
20. Font Color 252
21. Font Italic 252
22. Font Size 252
23. Font Size Height 253
24. Font Underline 253
25. Horizontal Alignment 253
26. Horizontal Margin 253
27. Multi-Line 255
28. Vertical Alignment 260
29. Vertical Margin 260
30. Is Touchable 254
31. Touch Size 260
32. Z Order 260

Related:

Add Widgets 322

7.4 Image

The Image widget allows the user to use an image from a file and place it on the screen. Images are indicator icons, backgrounds, or graphical visual effects.



Properties

Image1 Properties

Name	Value	Data Link
General		
Background Color	▨	
Image		
Name	Image1	
Visible	True	
Border		
Border Color	■	
Border Style	None	
Border Width	0	
Flash		
Flash Off Time (ms)	500	
Flash On Time (ms)	500	
Flashing	False	
Location		
Location	223, 152	
X	223	
Y	152	
Size		
Height	60	
Width	100	
Touchable		
Is Touchable	False	
Touch Size	100	
View		
Z Order	0	

Image Properties

Properties

1. Background Color 248
2. Image 254
3. Name 255
4. Visible 260
5. Border Color 248
6. Border Style 248
7. Border Width 248
8. Flash Off Time (ms) 252
9. Flash On Time (ms) 252

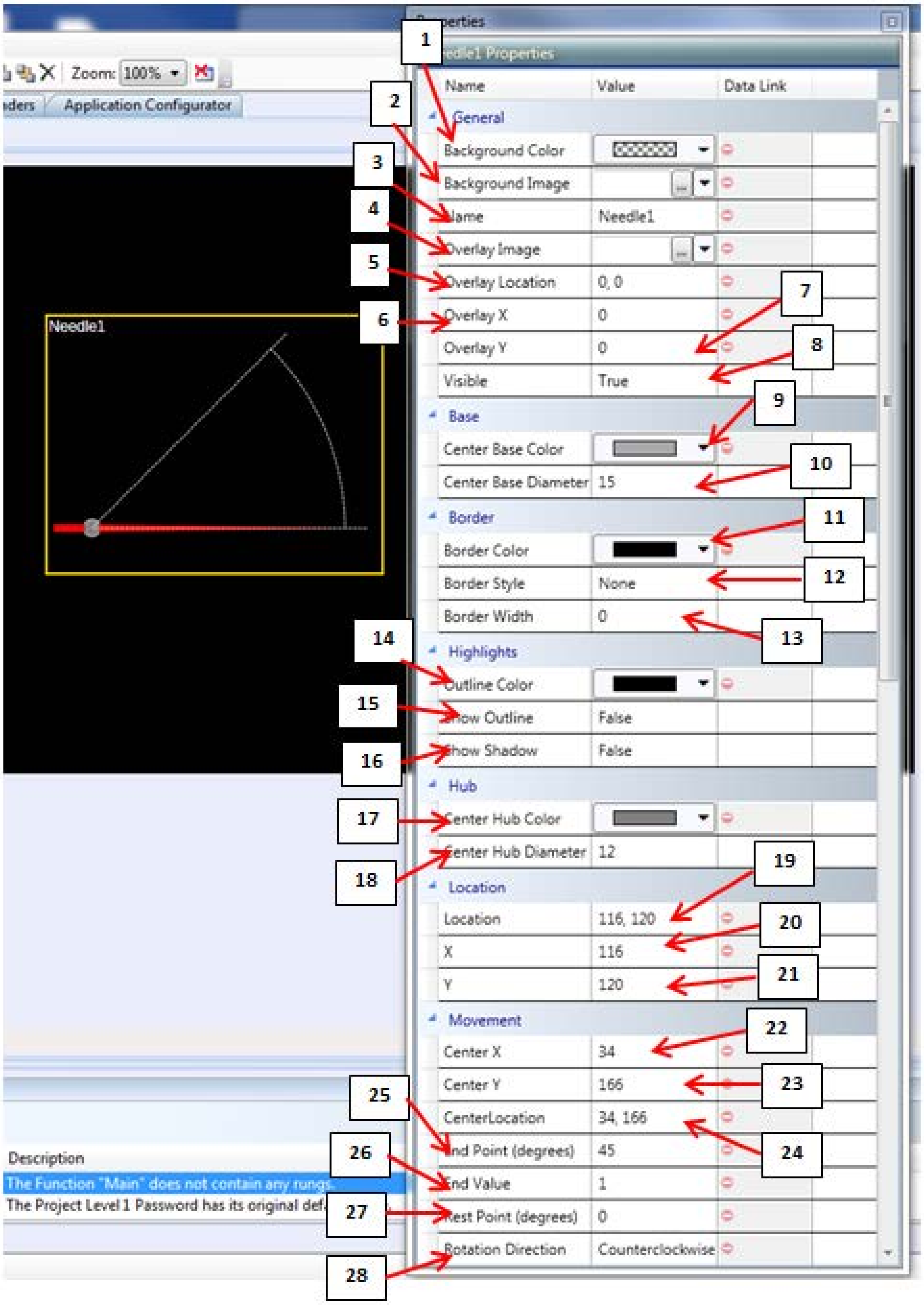
10. Flashing 252
11. Location 254
12. X 260
13. Y 260
14. Height 253
15. Width 260
16. Is Touchable 254
17. Touch Size 260
18. Z Order 260

Related:

Add Widgets 322

7.5 Needle

The Needle widget places a needle gauge on the screen that will show the progress of the item linked to it.



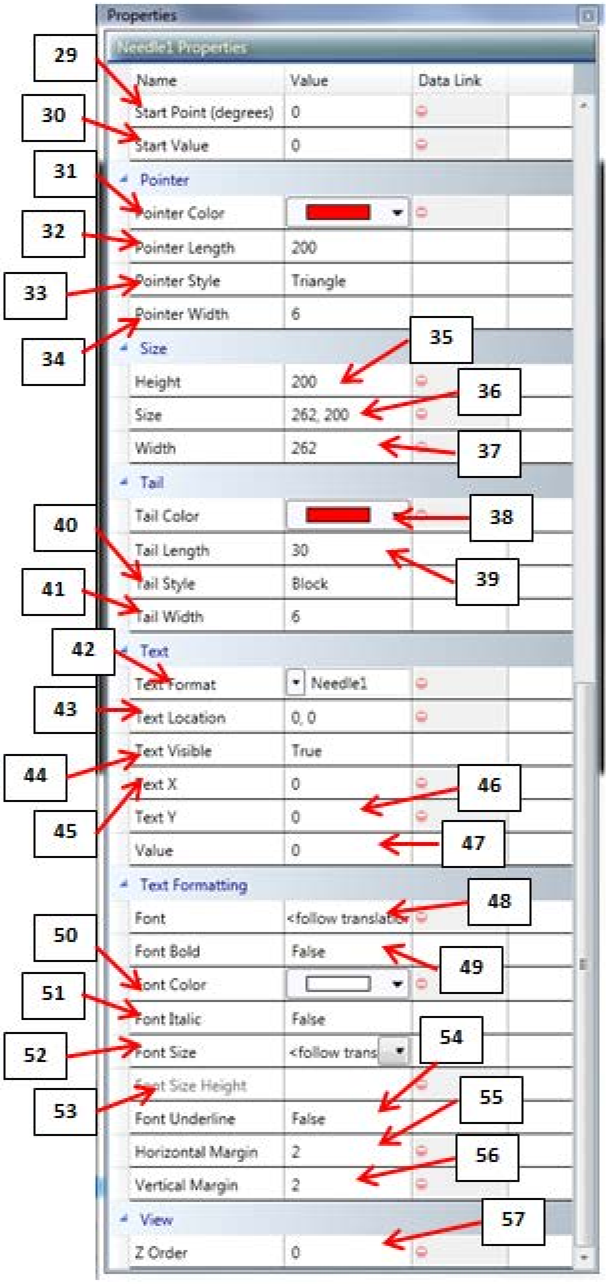
Name	Value	Data Link
General		
Background Color	▨	
Background Image		
Name	Needle1	
Overlay Image		
Overlay Location	0, 0	
Overlay X	0	
Overlay Y	0	
Visible	True	
Base		
Center Base Color	■	
Center Base Diameter	15	
Border		
Border Color	■	
Border Style	None	
Border Width	0	
Highlights		
Outline Color	■	
Show Outline	False	
Show Shadow	False	
Hub		
Center Hub Color	■	
Center Hub Diameter	12	
Location		
Location	116, 120	
X	116	
Y	120	
Movement		
Center X	34	
Center Y	166	
CenterLocation	34, 166	
End Point (degrees)	45	
End Value	1	
Rest Point (degrees)	0	
Rotation Direction	Counterclockwise	

Description

The Function "Main" does not contain any rungs.

The Project Level 1 Password has its original def

Needle Widget Properties



Name	Value	Data Link
Start Point (degrees)	0	
Start Value	0	
Pointer		
Pointer Color	🟥	
Pointer Length	200	
Pointer Style	Triangle	
Pointer Width	6	
Size		
Height	200	
Size	262, 200	
Width	262	
Tail		
Tail Color	🟥	
Tail Length	30	
Tail Style	Block	
Tail Width	6	
Text		
Text Format	Needle1	
Text Location	0, 0	
Text Visible	True	
Text X	0	
Text Y	0	
Value	0	
Text Formatting		
Font	<follow translator	
Font Bold	False	
Font Color	□	
Font Italic	False	
Font Size	<follow trans	
Font Size Height		
Font Underline	False	
Horizontal Margin	2	
Vertical Margin	2	
View		
Z Order	0	

Needle Widget Properties

Properties

1. Background Color 248
2. Background Image 248
3. Name 255
4. Overlay Image 256
5. Overlay Location 256
6. Overlay X 256
7. Overlay Y 257
8. Visible 260
9. Center Base Color 248

10. Center Base Diameter 248
11. Border Color 248
12. Border Style 248
13. Border Width 248
14. Outline Color 256
15. Show Outline 258
16. Show Shadow 258
17. Center Hub Color 248
18. Center Hub Diameter 248
19. Location 254
20. X 260
21. Y 260
22. Center X 249
23. Center Y 249
24. CenterLocation 249
25. End Point (degrees) 251
26. End Value 251
27. Rest Point (degrees) 257
28. Rotation Direction 257
29. Start Point (degrees) 258
30. Start Value 258
31. Pointer Color 257
32. Pointer Length 257
33. Pointer Style 257
34. Pointer Width 257
35. Height 253
36. Size 258
37. Width 260
38. Tail Color 259
39. Tail Length 259
40. Tail Style 259
41. Tail Width 259
42. Text Format 259
43. Text Location 259
44. Text Visible 259
45. Text X 259
46. Text Y 259
47. Value 260
48. Font 252
49. Font Bold 252
50. Font Color 252
51. Font Italic 252
52. Font Size 252
53. Font Size Height 253
54. Font Underline 253
55. Horizontal Margin 253
56. Vertical Margin 260
57. Z Order 260

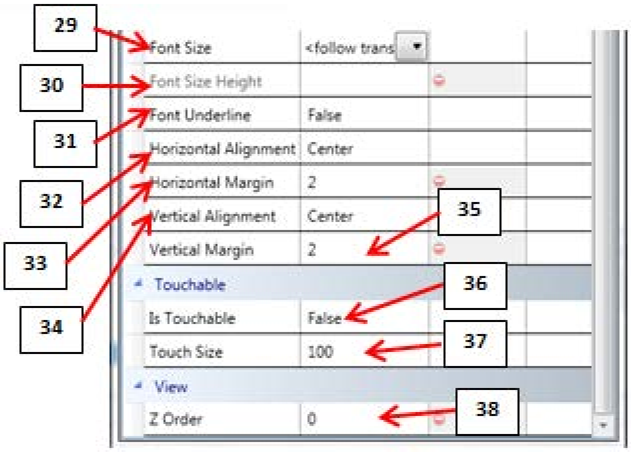
Related:

Add Widgets 322

7.6 Progress Bar

The Progress Bar widget places a progress bar on the screen that the user can link an item to, to track that item’s value graphically.

The Progress Bar is a horizontal or vertical bar that fills to a percentage of completeness or range, with optional text display of value, and optional overlay image to customize fill shape.



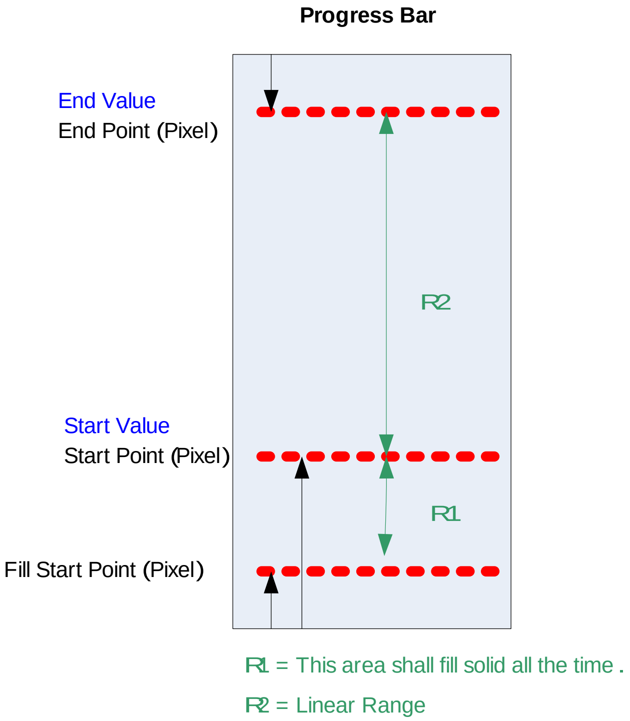
Font Size	<follow trans
Font Size Height	
Font Underline	False
Horizontal Alignment	Center
Horizontal Margin	2
Vertical Alignment	Center
Vertical Margin	2
Touchable	
Is Touchable	False
Touch Size	100
View	
Z Order	0

Progress Bar Properties

Properties

1. Background Color 248
2. Name 255
3. Overlay Image 256
4. Visible 260
5. Border Color 248
6. Border Style 248
7. Border Width 248
8. Location 254
9. X 260
10. Y 260
11. End Point (Pixel) 251
12. End Value 251
13. Fill Color 252
14. Fill Direction 252
15. Fill Start Point (Pixel) 252
16. Orientation 256
17. Start Point (Pixel) 258
18. Start Value 258
19. Height 253
20. Size 258
21. Width 260
22. Text Format 259
23. Text Visible 259

24. Value 260
25. Font 252
26. Font Bold 252
27. Font Color 252
28. Font Italic 252
29. Font Size 252
30. Font Size Height 253
31. Font Underline 253
32. Horizontal Alignment 253
33. Horizontal Margin 253
34. Vertical Alignment 260
35. Vertical Margin 260
36. Is Touchable 254
37. Touch Size 260
38. Z Order 260



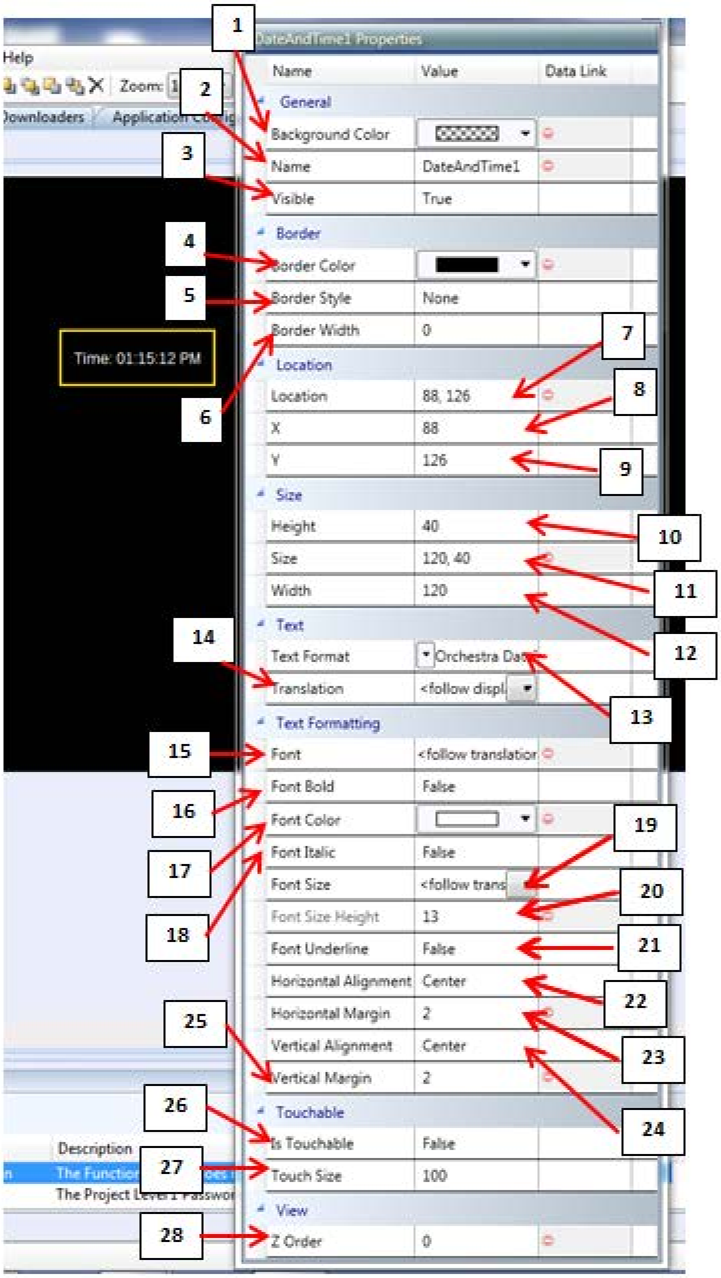
Progress Bar Start and End Point Explanation

Related:

Add Widgets 322

7.7 Time/Date

The Time/Date widget will place a label that has been preformatted to display the current date or time.



DateAndTime1 Properties

Name	Value	Data Link
General		
Background Color	▨	
Name	DateAndTime1	
Visible	True	
Border		
Border Color	■	
Border Style	None	
Border Width	0	
Location		
Location	88, 126	
X	88	
Y	126	
Size		
Height	40	
Size	120, 40	
Width	120	
Text		
Text Format	Orchestra Dat	
Translation	<follow displ	
Text Formatting		
Font	<follow translation	
Font Bold	False	
Font Color	□	
Font Italic	False	
Font Size	<follow trans	
Font Size Height	13	
Font Underline	False	
Horizontal Alignment	Center	
Horizontal Margin	2	
Vertical Alignment	Center	
Vertical Margin	2	
Touchable		
Is Touchable	False	
Touch Size	100	
View		
Z Order	0	

Date and Time Widget Properties

Properties

1. Background Color 248
2. Name 255
3. Visible 260
4. Border Color 248

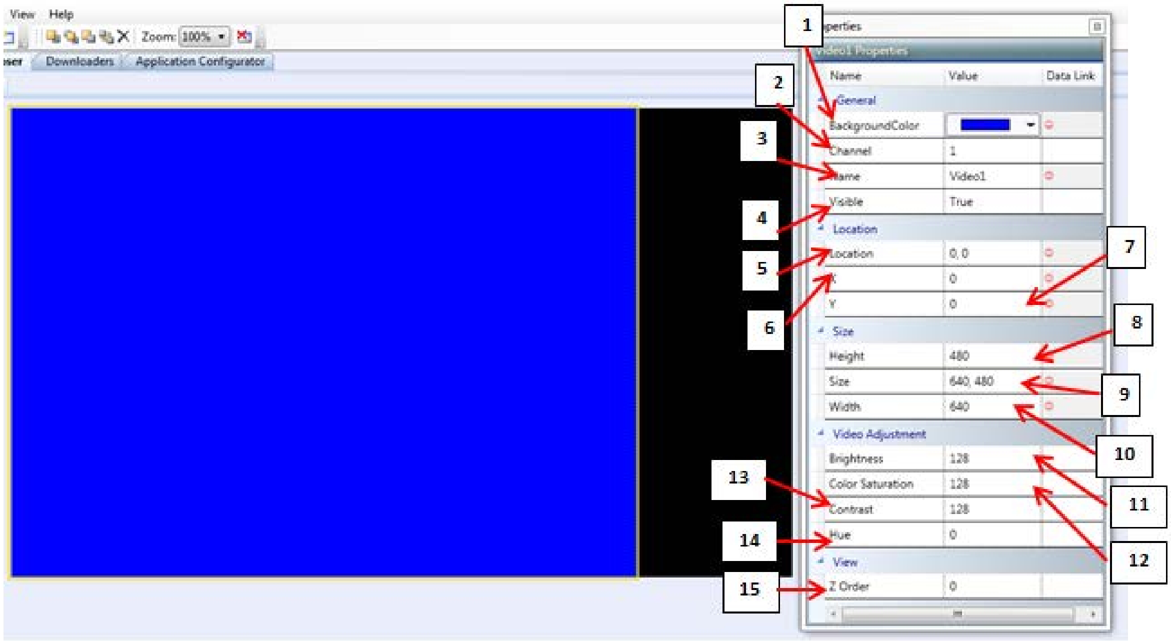
5. Border Style 248
6. Border Width 248
7. Location 254
8. X 260
9. Y 260
10. Height 253
11. Size 258
12. Width 260
13. Text Format 259
14. Translation 260
15. Font 252
16. Font Bold 252
17. Font Color 252
18. Font Italic 252
19. Font Size 252
20. Font Size Height 253
21. Font Underline 253
22. Horizontal Alignment 253
23. Horizontal Margin 253
24. Vertical Alignment 260
25. Vertical Margin 260
26. Is Touchable 254
27. Touch Size 260
28. Z Order 260

Related:

Add Widgets 322

7.8 Video

Video places a widget on the screen that will display a video feed from a camera. The video feed is in real-time, and displays in full-screen or window view.



Name	Value	Data Link
General		
BackgroundColor		
Channel	1	
Name	Video1	
Visible	True	
Location		
Location	0, 0	
X	0	
Y	0	
Size		
Height	480	
Size	640, 480	
Width	640	
Video Adjustment		
Brightness	128	
Color Saturation	128	
Contrast	128	
Hue	0	
View		
Z Order	0	

Video Widget Properties

The Video widget has a maximum size of 1024 x 1024, with both axes independent. A 7" display has a maximum size of 800x480, and a 10" display has a maximum size of 1280x800. The video widget limits the width or height appropriately to whichever is smaller based on screen size.

Properties

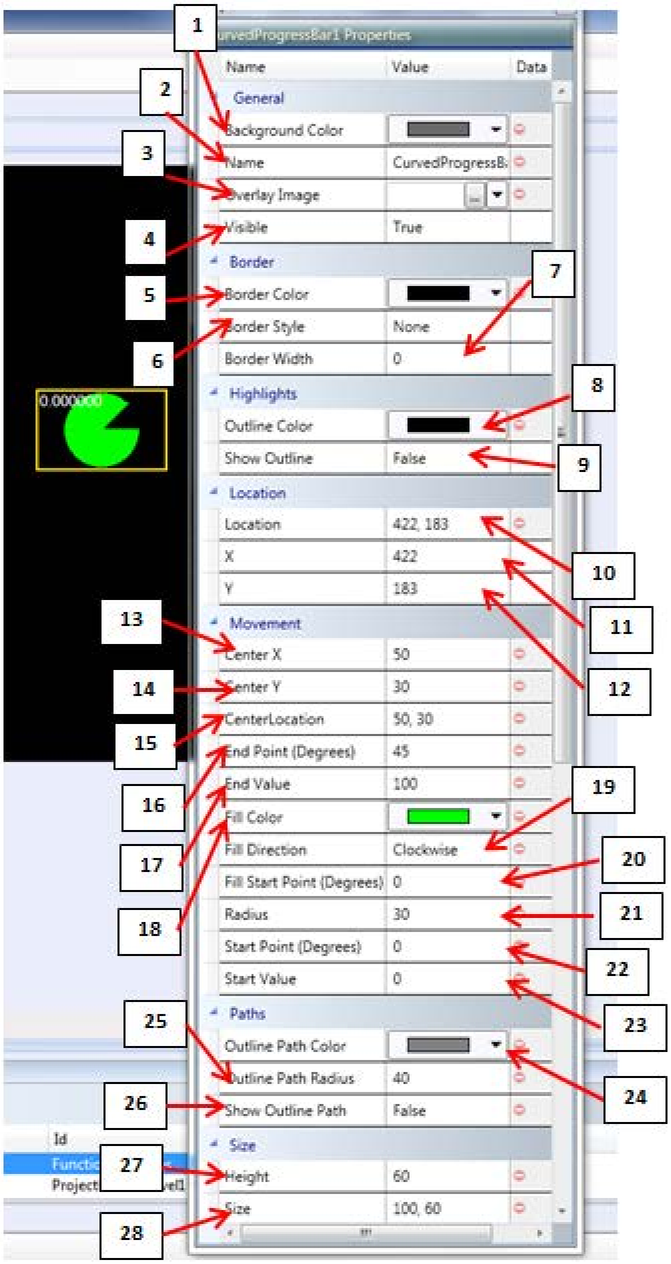
1. Background Color 248
2. Channel 249
3. Name 255
4. Visible 260
5. Location 254
6. X 260
7. Y 260
8. Height 253
9. Size 258
10. Width 260
11. Brightness 248
12. Color Saturation 249
13. Contrast 249
14. Hue 253
15. Z Order 260

Related:

Add Widgets 322

7.9 Curved Progress Bar

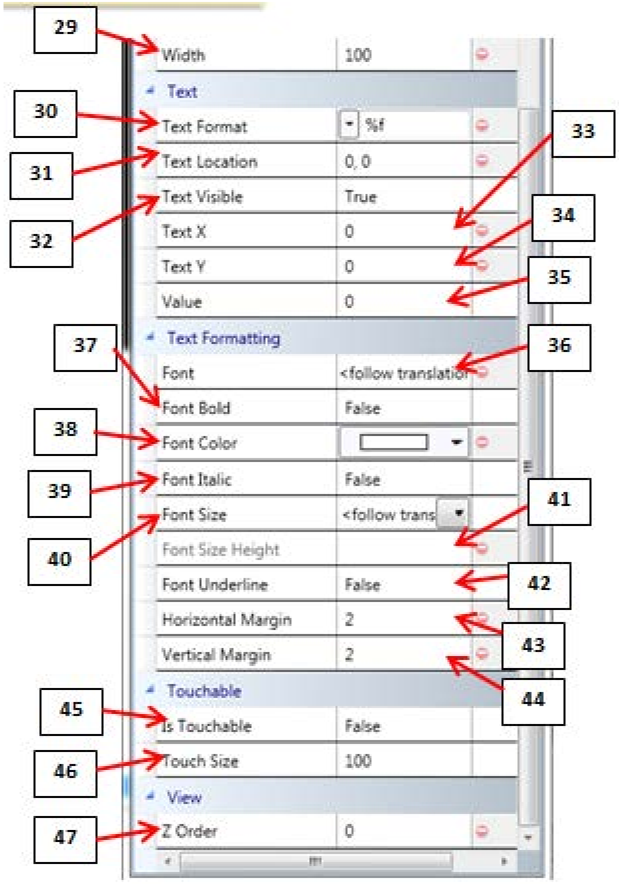
The Curved Progress Bar works similarly to the Progress bar, but instead it fills in a curved fashion versus a straight fashion.



urvedProgressBar1 Properties

Name	Value	Data
General		
Background Color	▨	
Name	CurvedProgressB	
Overlay Image		
Visible	True	
Border		
Border Color	■	
Border Style	None	
Border Width	0	
Highlights		
Outline Color	■	
Show Outline	False	
Location		
Location	422, 183	
X	422	
Y	183	
Movement		
Center X	50	
Center Y	30	
CenterLocation	50, 30	
End Point (Degrees)	45	
End Value	100	
Fill Color	🟩	
Fill Direction	Clockwise	
Fill Start Point (Degrees)	0	
Radius	30	
Start Point (Degrees)	0	
Start Value	0	
Paths		
Outline Path Color	▨	
Outline Path Radius	40	
Show Outline Path	False	
Size		
Height	60	
Size	100, 60	

Curved Progress Bar Properties



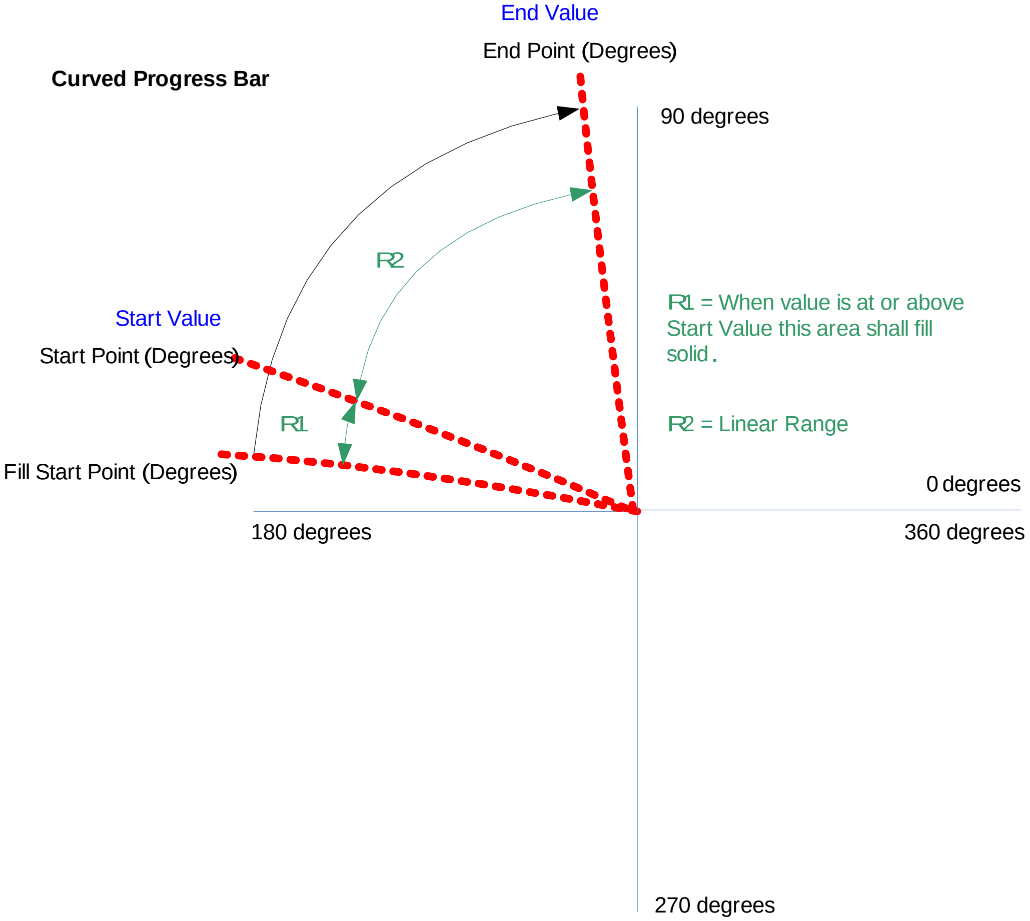
Width	100
Text	
Text Format	%f
Text Location	0, 0
Text Visible	True
Text X	0
Text Y	0
Value	0
Text Formatting	
Font	<follow translatio
Font Bold	False
Font Color	
Font Italic	False
Font Size	<follow trans
Font Size Height	
Font Underline	False
Horizontal Margin	2
Vertical Margin	2
Touchable	
Is Touchable	False
Touch Size	100
View	
Z Order	0

Additional Curved Progress Bar Properties

Properties

1. Background Color 248
2. Name 255
3. Overlay Image 256
4. Visible 260
5. Border Color 248
6. Border Style 248
7. Border Width 248
8. Outline Color 256
9. Show Outline 258
10. Location 254
11. X 260
12. Y 260
13. Center X 249
14. Center Y 249
15. CenterLocation 249
16. End Point (degrees) 251
17. End Value 251
18. Fill Color 252
19. Fill Direction 252

20. Fill Start Point (Degrees) 252
21. Radius 257
22. Start Point (degrees) 258
23. Start Value 258
24. Outline Path Color 256
25. Outline Path Radius 256
26. Show Outline Path 258
27. Height 253
28. Size 258
29. Width 260
30. Text Format 259
31. Text Location 259
32. Text Visible 259
33. Text X 259
34. Text Y 259
35. Value 260
36. Font 252
37. Font Bold 252
38. Font Color 252
39. Font Italic 252
40. Font Size 252
41. Font Size Height 253
42. Font Underline 253
43. Horizontal Margin 253
44. Vertical Margin 260
45. Is Touchable 254
46. Touch Size 260
47. Z Order 260



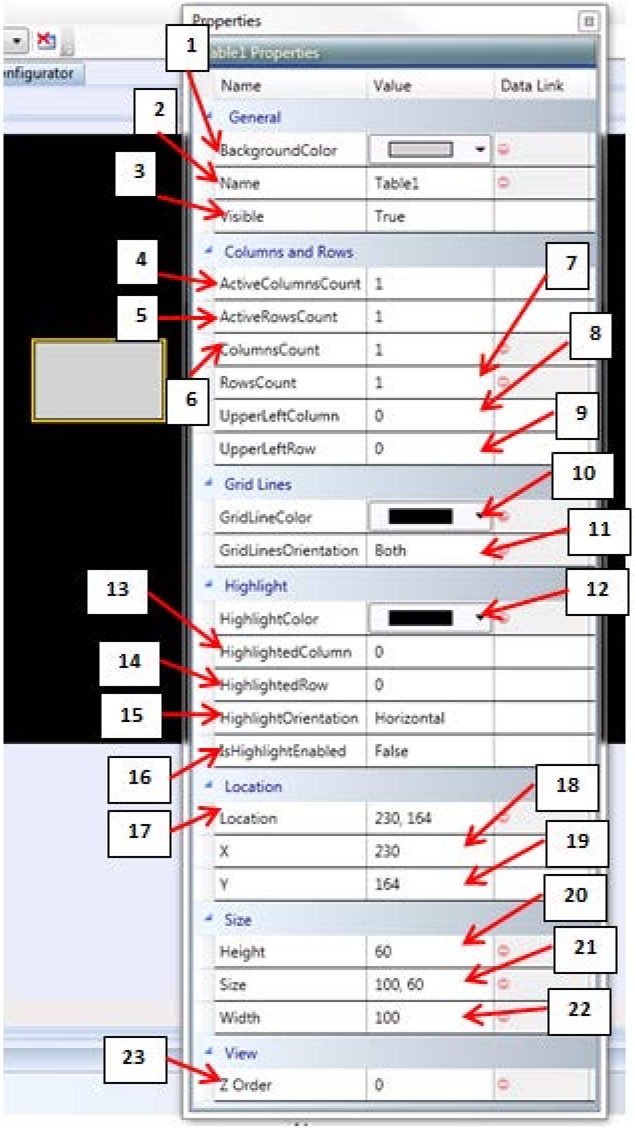
Curved Progress Bar Start and End Point Explanation

Related:

Add Widgets 322
Progress Bar 152

7.10 Table

The Table widget creates a spreadsheet data table within the screen. A small, cell sized window is shown, which the user is able to scroll through to show what is required.



Properties

ble1 Properties

Name	Value	Data Link
General		
BackgroundColor		
Name	Table1	
Visible	True	
Columns and Rows		
ActiveColumnsCount	1	
ActiveRowsCount	1	
ColumnsCount	1	
RowsCount	1	
UpperLeftColumn	0	
UpperLeftRow	0	
Grid Lines		
GridLineColor		
GridLinesOrientation	Both	
Highlight		
HighlightColor		
HighlightedColumn	0	
HighlightedRow	0	
HighlightOrientation	Horizontal	
IsHighlightEnabled	False	
Location		
Location	230, 164	
X	230	
Y	164	
Size		
Height	60	
Size	100, 60	
Width	100	
View		
Z Order	0	

Table Widget Properties

Properties

1. Background Color 248
2. Name 255
3. Visible 260
4. Active Columns Count 248
5. Active Rows Count 248
6. Columns Count 249
7. Rows Count 257

8. Upper Left Column 260
9. Upper Left Row 260
10. Grid Line Color 253
11. Grid Lines Orientation 253
12. Hightlight Color 253
13. Highlighted Column 253
14. Highlighted Row 253
15. Highlight Orientation 253
16. Is Highlight Enabled 254
17. Location 254
18. X 260
19. Y 260
20. Height 253
21. Size 258
22. Width 260
23. Z Order 260

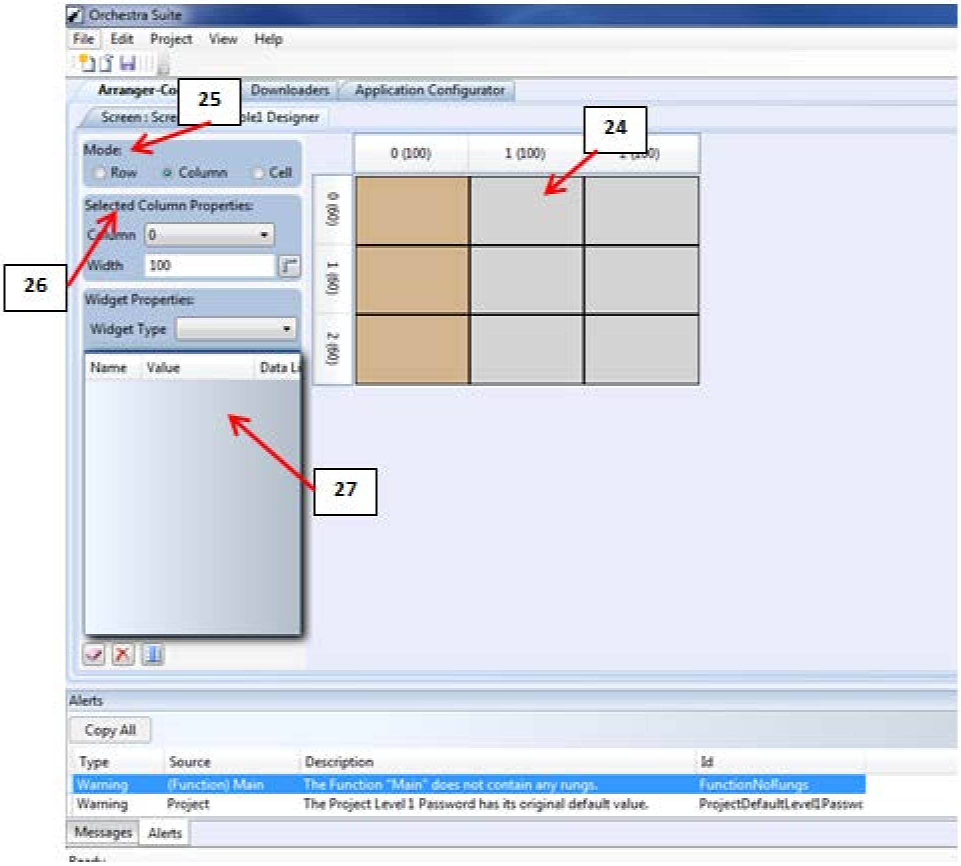
Related:

Table Designer Tab 162
Add Widgets 322

7.10.1 Table Designer Tab

Use the Table Designer tab to customize table widgets.

Double clicking on the table widget opens the Table Designer tab where the contents of the table can be edited. The tab can also be opened by right clicking the item within the Explorer pane and selecting open.



Mode: ○ Row ● Column ○ Cell

Selected Column Properties:
Column 0
Width 100

Widget Properties:
Widget Type

Name	Value	Data L

Alerts

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passw

Table Designer

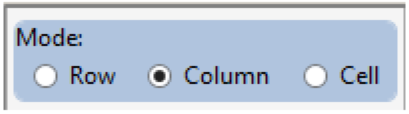
Related:
Table 161
Mode 163
Selected Properties 164
Widget Properties 164

7.10.1.1 Mode

There are three modes:

1. Row
2. Column
3. Cell

Click the radio button next to the mode to select that mode for editing.



Mode:
○ Row ● Column ○ Cell

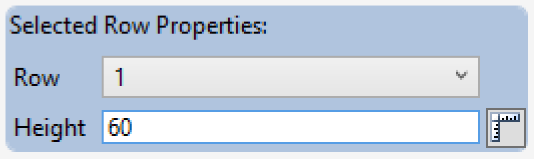
Table Designer: Mode Pane

Related:
Table 161
Table Designer Tab 162

[Selected Properties](164)
[Widget Properties](164)

7.10.1.2 Selected Properties

The selected properties change to match the selected mode. Use this pane to set row height and column width, as well as choose a cell to edit. Select a row/column from the drop-down menu.



Selected Row Properties:

Row: 1

Height: 60

Table Designer Tab: Selected Properties

Related:
[Table](161)
[Table Designer Tab](162)
[Mode](163)
[Widget Properties](164)

7.10.1.3 Widget Properties

Widget Type

Choose either Image or Label from the drop-down menu. The selection activates the properties pane.

Properties Pane

This pane is used to configure the cell, row, or column widgets inside the table. To make changes to the entire table properties, use the [main Properties pane](32).

Related:
[Table](161)
[Table Designer Tab](162)
[Mode](163)
[Selected Properties](164)

7.11 Command Widget

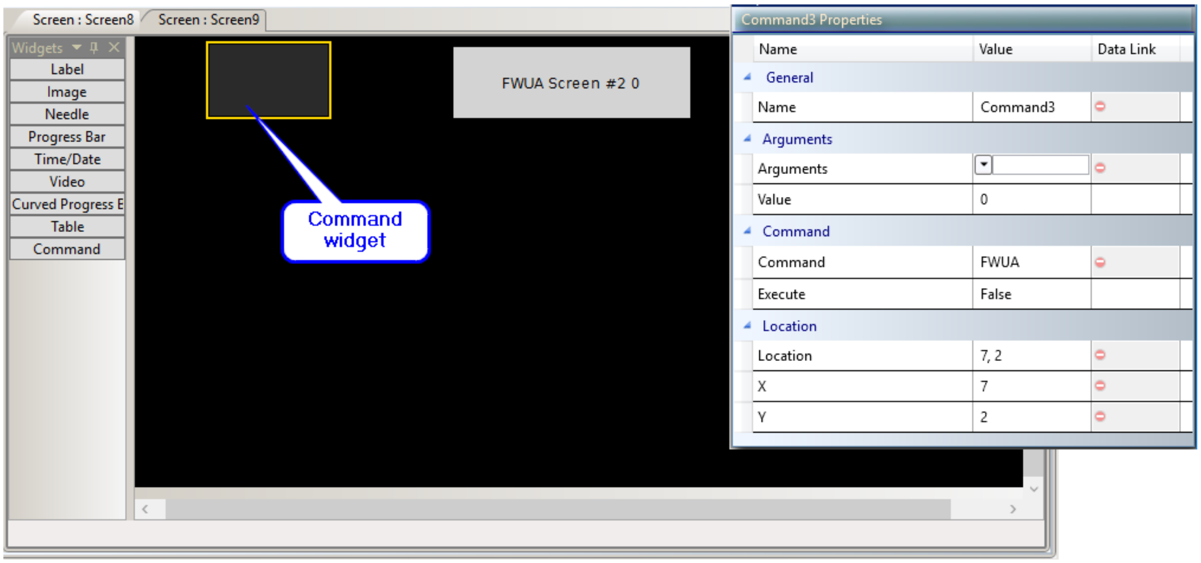
The Command Widget gives Presto™ or Rungs programmers the ability to issue a command directly to Arranger™.

Arranger™ allocates 2 status bits in the already existing ErrorBits IOMap item and fills them with Status and Success, so that the programmer will know when the command has been executed and if it succeeded. The predefined set of supported commands is contained in a static CommandList (similar to the DateTime Format list). The List Index is linkable. The programmer adds a new command widget for each command they want to execute. This is because some commands require variable parameters and some do not.

Arranger™ executes external commands verbatim.

Command Examples

- Calibrate touchscreen
- Verify touchscreen calibration
- Take or Release Window Focus
- Launch a PDF reader and display a PDF file
- Will require a (new) FileList as a parameter
- Start/Stop Serial Passthru
- Start/Stop a Reprogramming Applet
- Enable/Disable X-Server
- Start/Stop Third Party Application (Streetwise)
- Launch a shell script
- Parameter List
- Reboot



Command3 Properties

Name	Value	Data Link
General		
Name	Command3	
Arguments		
Arguments		
Value	0	
Command		
Command	FWUA	
Execute	False	
Location		
Location	7, 2	
X	7	
Y	2	

Command Widget

Properties

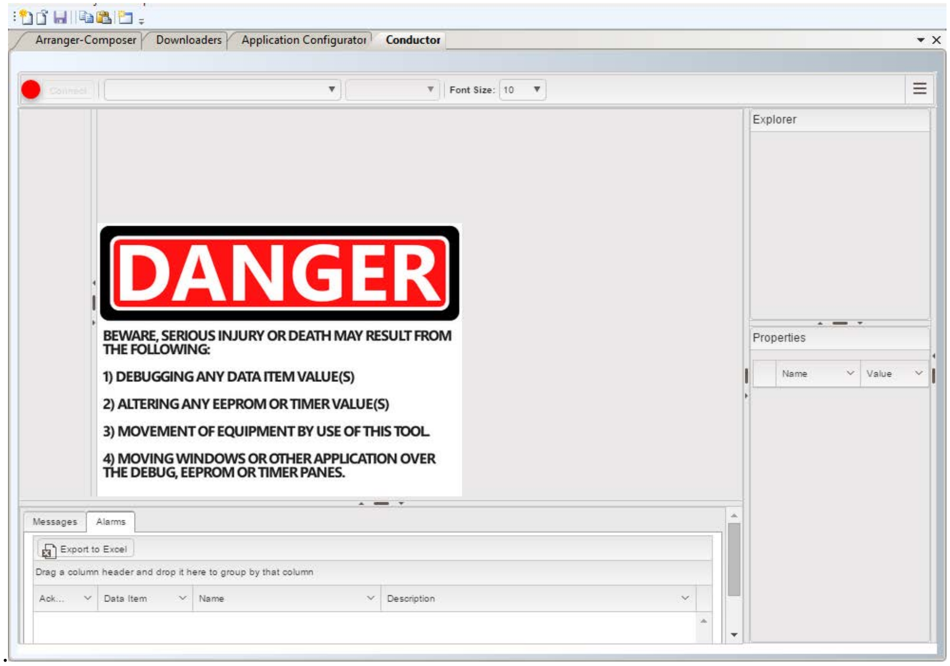
1. Name 255
2. Arguments 248--Allows links
3. Value 260
4. Command 249--Does not allow links
5. Execute 251
6. Location 254
7. X 260
8. Y 260

Related:
Add Widgets 322

Conductor™

8 Conductor™

Conductor™ is used to connect to the master module and get real time feedback for all of the data items in the system. It is also used to debug (ex: override the value to turn on an output for trouble shooting). Also, the user can update EEPROMs (ex: enable or disable features).



DANGER

BEWARE, SERIOUS INJURY OR DEATH MAY RESULT FROM THE FOLLOWING:

1) DEBUGGING ANY DATA ITEM VALUE(S)

2) ALTERING ANY EEPROM OR TIMER VALUE(S)

3) MOVEMENT OF EQUIPMENT BY USE OF THIS TOOL.

4) MOVING WINDOWS OR OTHER APPLICATION OVER THE DEBUG, EEPROM OR TIMER PANES.

Conductor™ Landing Screen

Related:

EEPROM 101

8.1 Introduction

Conductor™ is used for connecting to the master module and getting real time feedback for all of the data items in the system. It is also used to debug (ex: override the value to turn on an output for trouble shooting). Also, the user can update EEPROMs (ex: enable or disable features).

Related:

Safety Use Precautions 170
Using Conductor 306

8.2 Safety Use Precautions

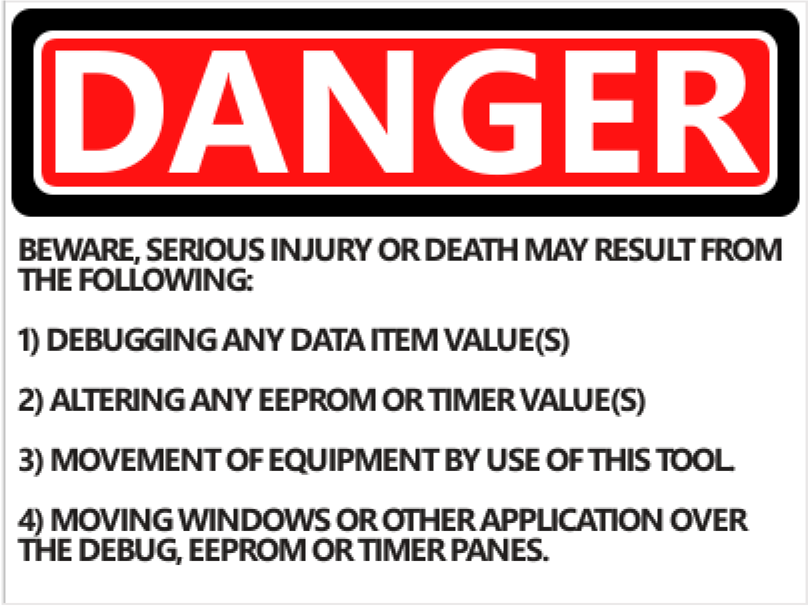
The Conductor™ software is designed to be used in diagnostic troubleshooting of HED®, Inc. installed CANLink® module systems. Because this software can alter the settings of any of these modules, extreme caution should be taken to understand how module setting adjustments will impact operation of the Original Equipment Manufacturers (OEM) product.

Users of this software should be fully authorized and trained in the use of the CANLink® Orchestra® suite of software packages. The user should also have full working knowledge of the OEM product before making any troubleshooting adjustments with the Conductor™ software.

Death or serious injury may result from debugging and/or altering any Data Item. Users should also understand that adjustments made with this software could cause unexpected movement or other undesired results of the OEMs product.

Warning

WHEN DEBUG IS ACTIVATED, YOU ARE ASSUMING CONTROL OF SELECTED DATA ITEMS DIRECTLY AND POTENTIALLY BYPASSING SAFETY INTERLOCKS AND NORMAL OPERATON. THIS MAY RESULT IN SERIOUS INJURY OR DEATH TO YOU OR OTHERS IN THE VICINITY OF THE EQUIPMENT.



DANGER

BEWARE, SERIOUS INJURY OR DEATH MAY RESULT FROM THE FOLLOWING:

1) DEBUGGING ANY DATA ITEM VALUE(S)

2) ALTERING ANY EEPROM OR TIMER VALUE(S)

3) MOVEMENT OF EQUIPMENT BY USE OF THIS TOOL.

4) MOVING WINDOWS OR OTHER APPLICATION OVER THE DEBUG, EEPROM OR TIMER PANES.

Software Safety Warning

Related:

Emergency Stop Safe Mode 171

8.2.1 Emergency Stop Safe Mode

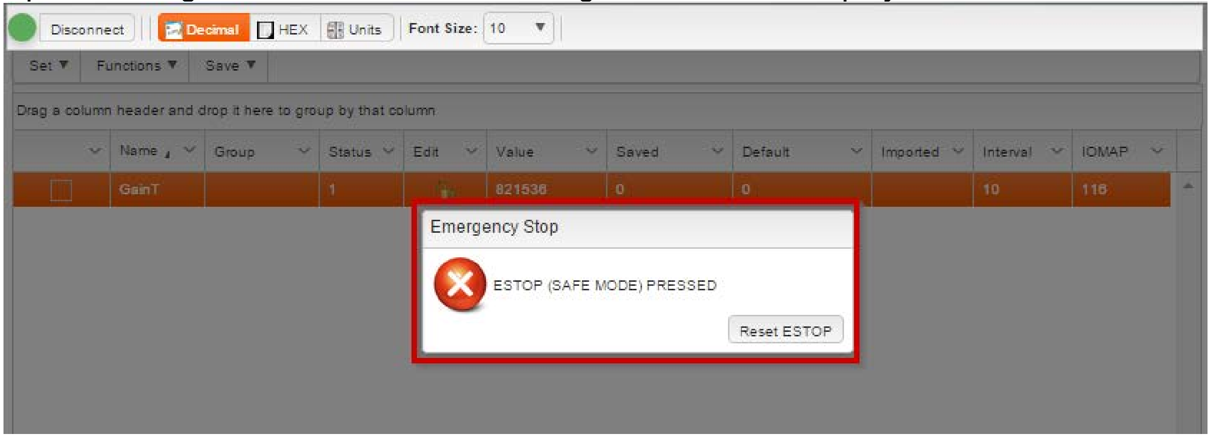
The Emergency Stop (E-Stop) - Safe Mode is a predetermined state defined by the OEM to ensure the product reacts predictably in the event of emergency. This Safe Mode state will put the OEM product into the safest configuration by turning selected Outputs either ON or OFF. It is suggested that you contact the OEM to understand how the product will behave once the Safe Mode is engaged.

Activate Emergency Stop Safe Mode

At any time that the Conductor™ software is connected to an OEM product, the Emergency Stop (E-STOP) - Safe Mode can be activated one of two ways.

- Depressing the computer keyboard spacebar
- Disconnecting the communication cable between the computer and the product

Upon entering the Safe Mode, the following screen will be displayed.



Font Size: 10

Drag a column header and drop it here to group by that column

	Name	Group	Status	Edit	Value	Saved	Default	Imported	Interval	IOMAP
□	GainT		1		821536	0	0		10	116

Emergency Stop

ESTOP (SAFE MODE) PRESSED

Safe Mode Screen

While in the Safe Mode, some Data Items configured for Safe Mode will not update. These Data Items are fixed until Safe Mode has been reset. Click RESET ESTOP to run the system back in normal mode. All Data Items will update periodically after resetting the Safe Mode.

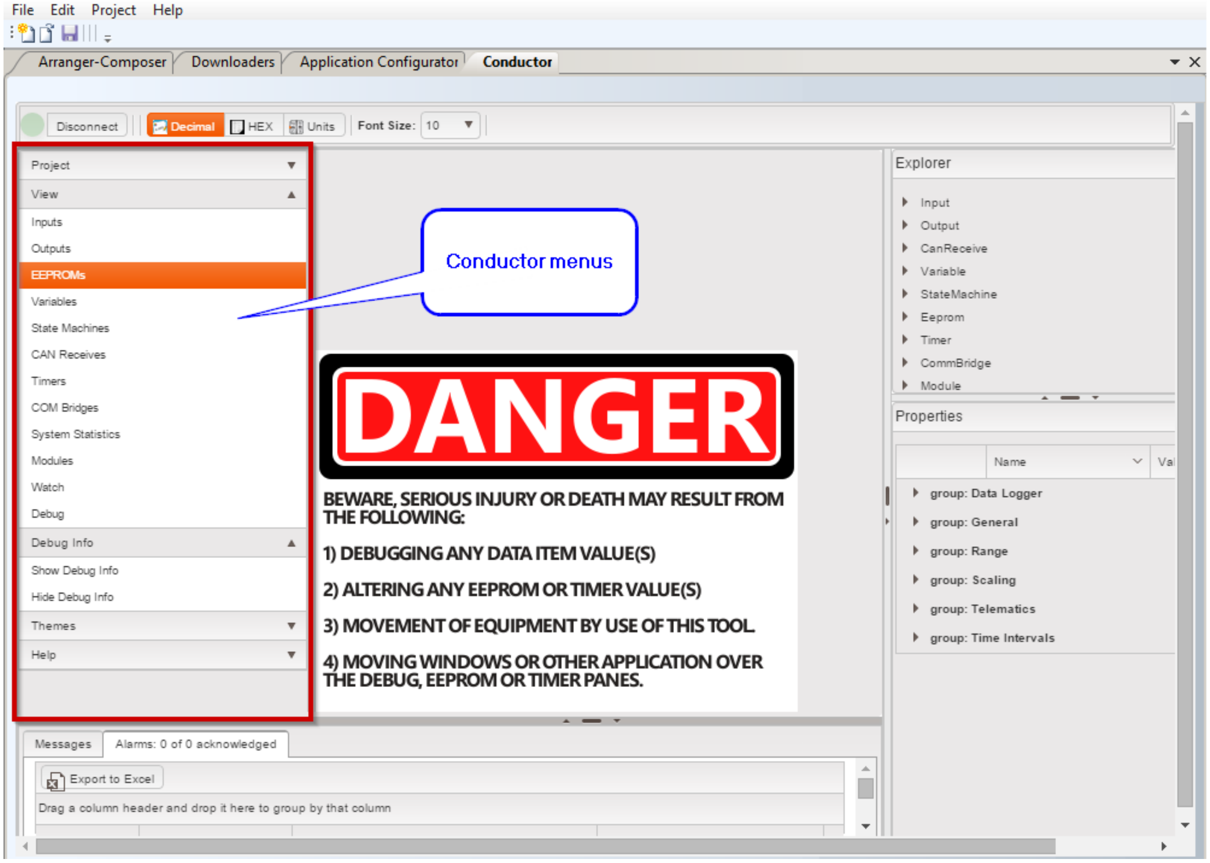
Related:

Safety Use Precautions 170
Outputs Window 183

8.3 Software Layout

8.3.1 Menus

Conductor™ menus are located on the left side of the screen.



DANGER

BEWARE, SERIOUS INJURY OR DEATH MAY RESULT FROM THE FOLLOWING:

1) DEBUGGING ANY DATA ITEM VALUE(S)

2) ALTERING ANY EEPROM OR TIMER VALUE(S)

3) MOVEMENT OF EQUIPMENT BY USE OF THIS TOOL.

4) MOVING WINDOWS OR OTHER APPLICATION OVER THE DEBUG, EEPROM OR TIMER PANES.

Drag a column header and drop it here to group by that column

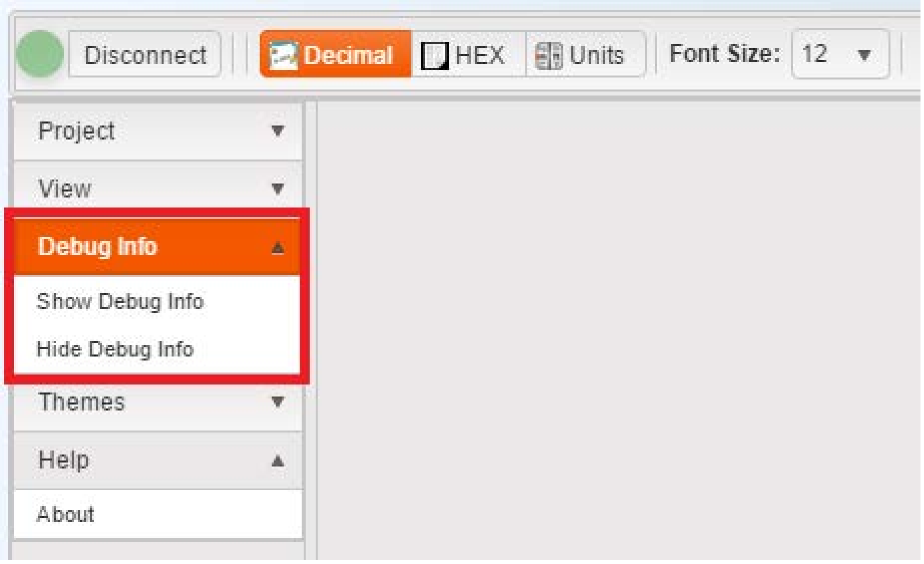
Conductor Menus

Related:

Project Menu 173
View Menu 176
Help Menu 177

8.3.1.1 Debug Info Menu

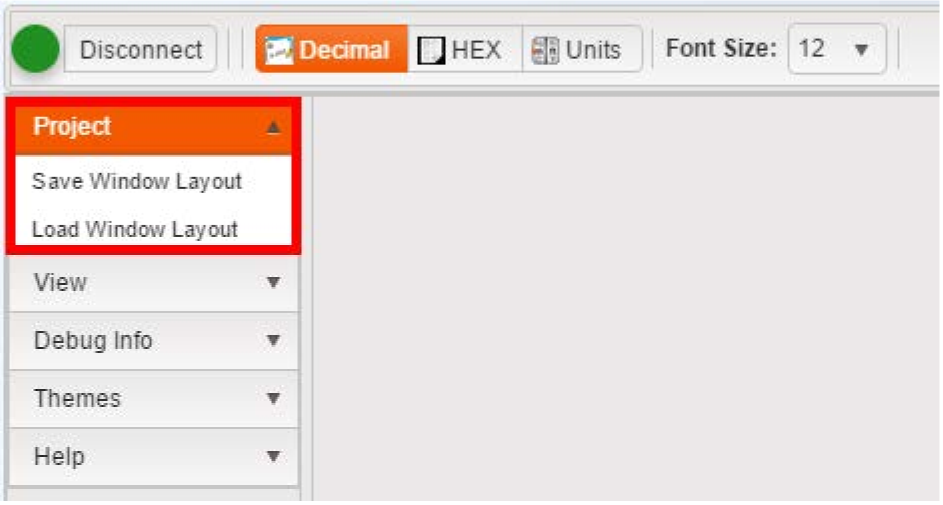
This menu is used to select whether to show or hide the Debug Info window.



Debug Info Window

8.3.1.2 Project Menu

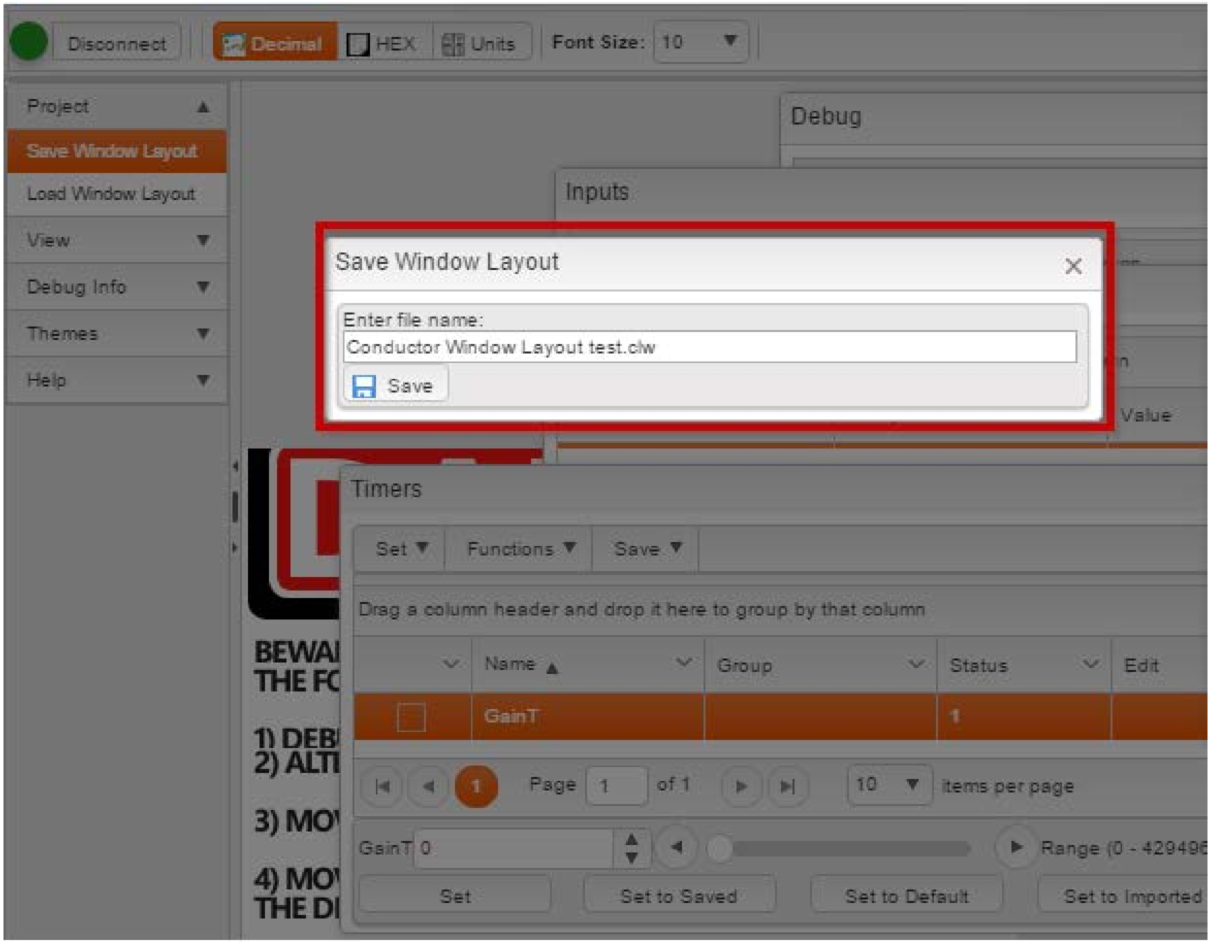
Use this function to save the window layout or load the window layout. Save a window layout while troubleshooting specific issues to come back to that same layout.



Project Menu

Save Window Layout

This feature lets the user save the arrangement of the windows on the Conductor™ screen.



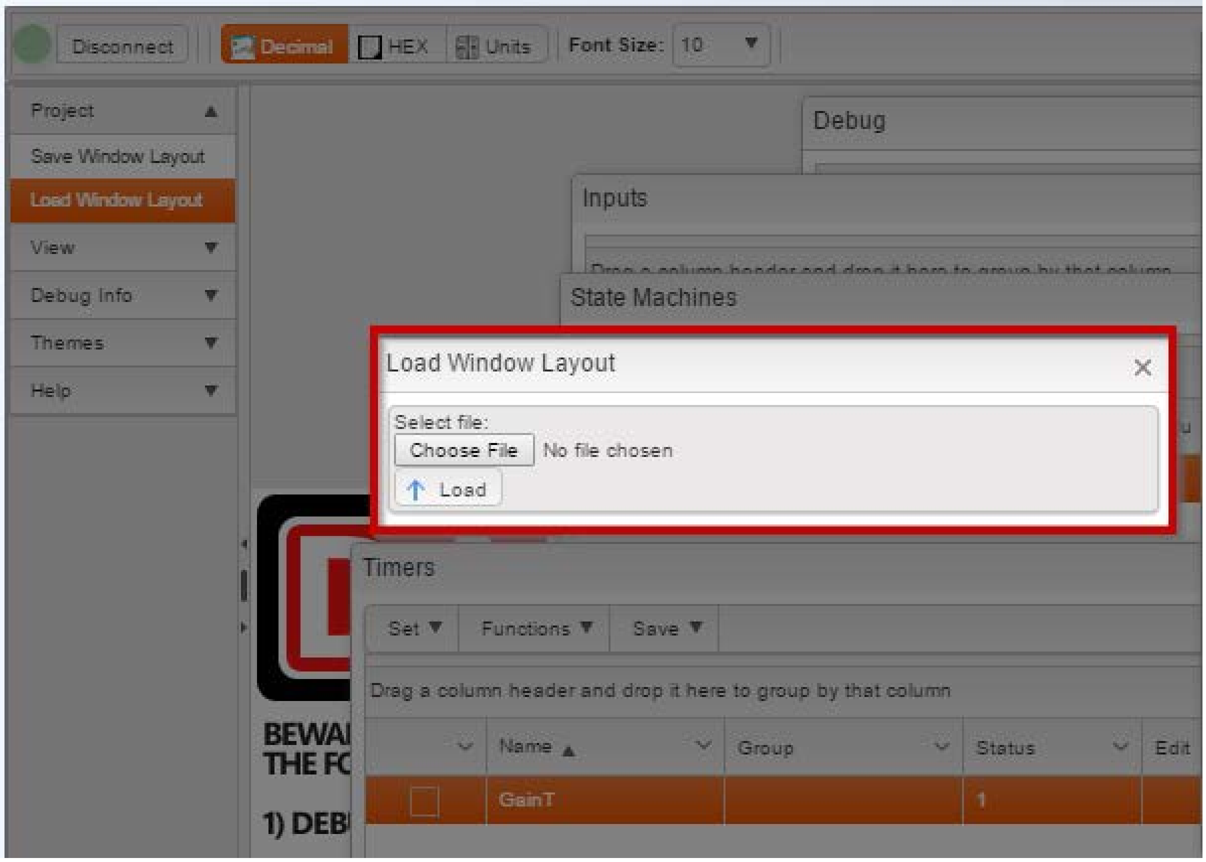
Save Window Layout

Enter file name: Conductor Window Layout test.clw

Save Window Layout

Load Window Layout

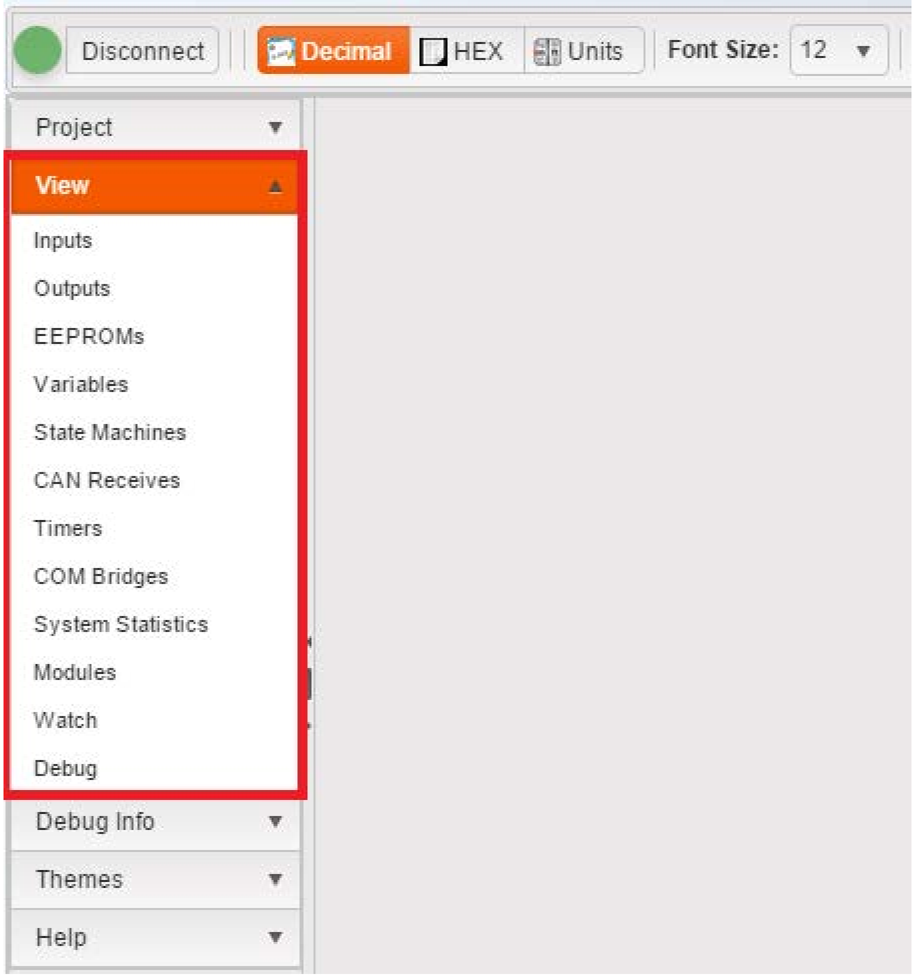
This feature lets the user load and open a saved window layout.



Load Window Layout

8.3.1.3 View Menu

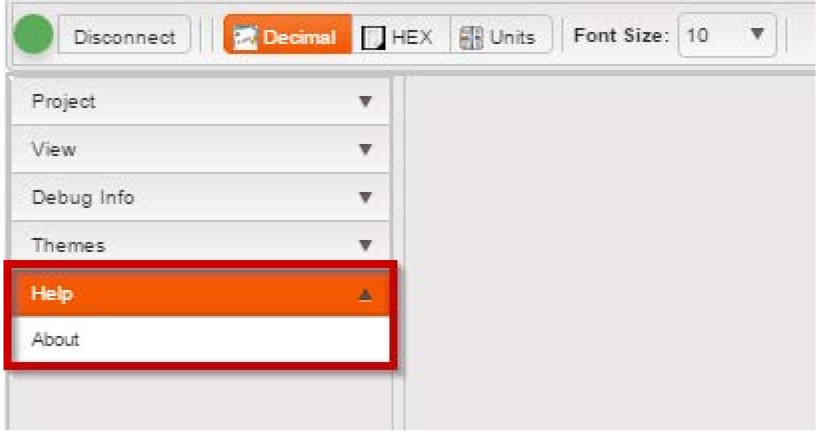
Use this function to open the following Conductor™ windows: Inputs 182, Outputs 183, EEPROM 197, Variables 186, State Machines 187, CAN Receives 188, Timers 189, Modules 181, Watch 205 or Debug 208. If a data item type is not configured for this specific project, that data items name will be grayed out.



View Menu

8.3.1.4 Help Menu

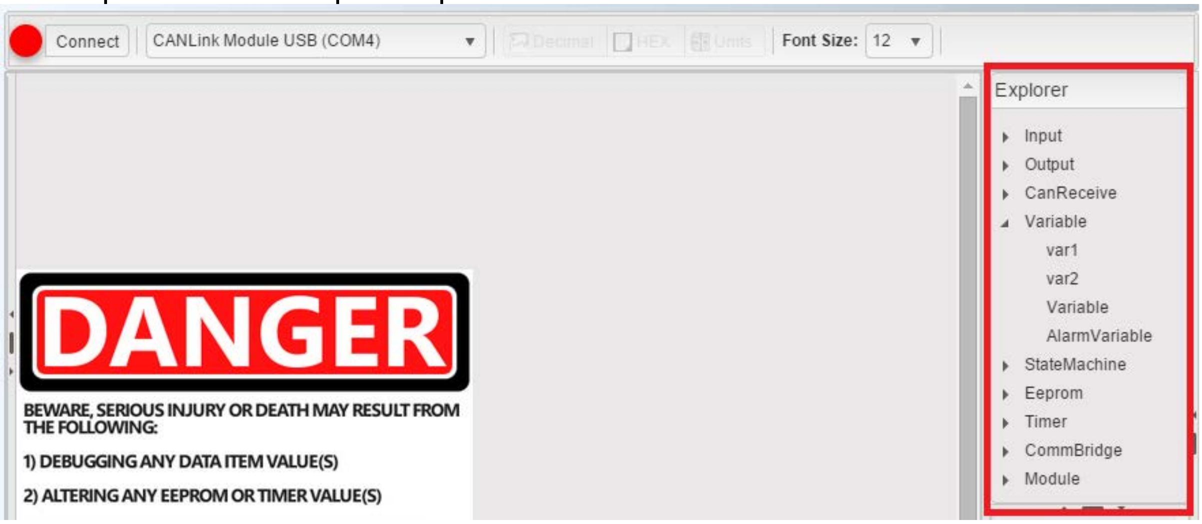
Use this feature to find out about the Conductor™ or request a temporary license file.



Help Menu

8.3.2 Explorer Pane

The Explorer pane mirrors the Orchestra® Explorer pane. Clicking one of the data items opens it in the Properties pane.


Connect
CANLink Module USB (COM4)
Decimal
HEX
Units
Font Size:
12
Explorer
Input
Output
CanReceive
Variable
var1
var2
Variable
AlarmVariable
StateMachine
Eeprom
Timer
CommBridge
Module
DANGER
BEWARE, SERIOUS INJURY OR DEATH MAY RESULT FROM THE FOLLOWING:
1) DEBUGGING ANY DATA ITEM VALUE(S)
2) ALTERING ANY EEPROM OR TIMER VALUE(S)

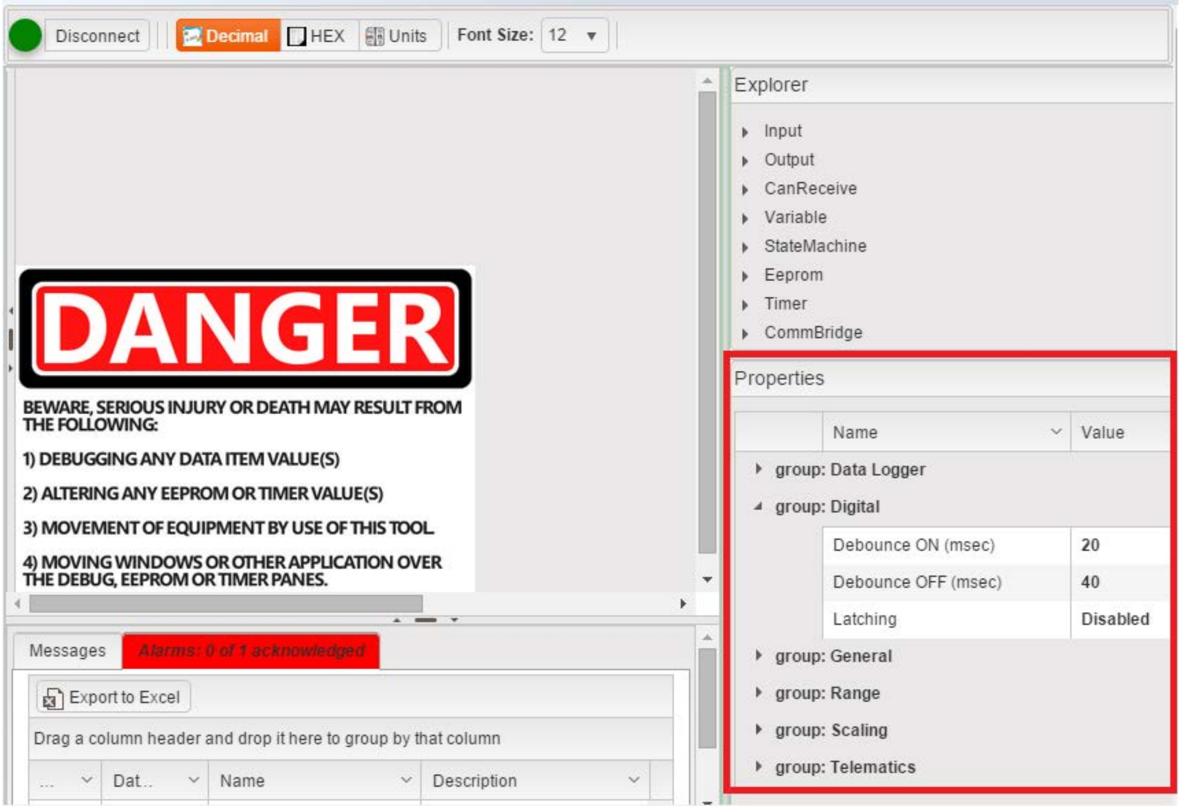
Explorer Pane

Related:

Properties Pane 178

8.3.3 Properties Pane

Data item properties display in the Properties pane. To populate this pane, click a data item in the Explorer pane 177.



DANGER

BEWARE, SERIOUS INJURY OR DEATH MAY RESULT FROM THE FOLLOWING:

1) DEBUGGING ANY DATA ITEM VALUE(S)

2) ALTERING ANY EEPROM OR TIMER VALUE(S)

3) MOVEMENT OF EQUIPMENT BY USE OF THIS TOOL

4) MOVING WINDOWS OR OTHER APPLICATION OVER THE DEBUG, EEPROM OR TIMER PANES.

Drag a column header and drop it here to group by that column

Name	Value
group: Data Logger	
group: Digital	
Debounce ON (msec)	20
Debounce OFF (msec)	40
Latching	Disabled
group: General	
group: Range	
group: Scaling	
group: Telematics	

Properties Pane

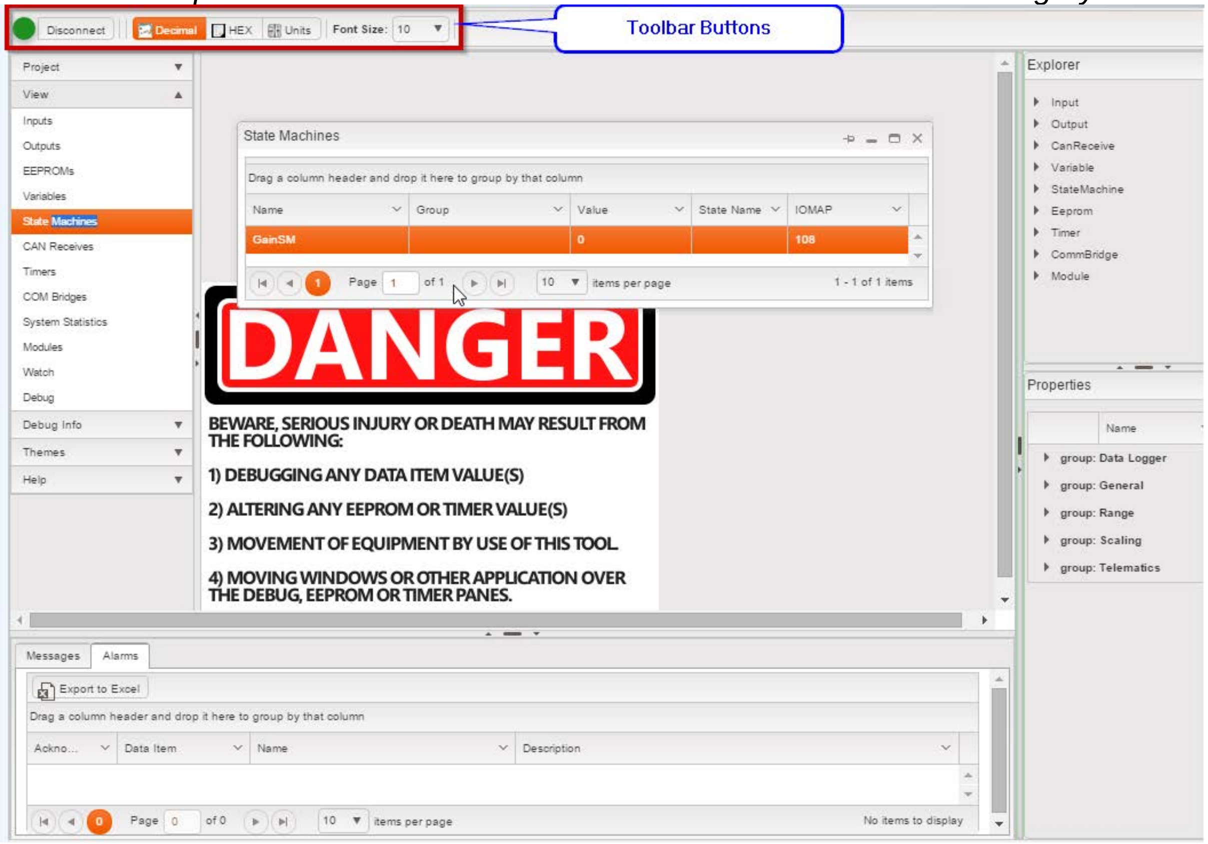
Related:

Explorer Pane 177

8.4 Toolbar Buttons

These buttons provide simple access to all major Conductor™ data windows and other Conductor™ functions.

Note: With this release of the Conductor™ software, the ViewGraph Windowbutton and ViewSnapshot Windowbutton are not active features and will remain grayed out.



Drag a column header and drop it here to group by that column

Name	Group	Value	State Name	IOMAP
GainSM		0		108

1 Page 1 of 1 10 items per page 1 - 1 of 1 items

DANGER

BEWARE, SERIOUS INJURY OR DEATH MAY RESULT FROM THE FOLLOWING:

1) DEBUGGING ANY DATA ITEM VALUE(S)

2) ALTERING ANY EEPROM OR TIMER VALUE(S)

3) MOVEMENT OF EQUIPMENT BY USE OF THIS TOOL.

4) MOVING WINDOWS OR OTHER APPLICATION OVER THE DEBUG, EEPROM OR TIMER PANES.

Drag a column header and drop it here to group by that column

Ackno...	Data Item	Name	Description

0 Page 0 of 0 10 items per page No items to display

Toolbar Buttons

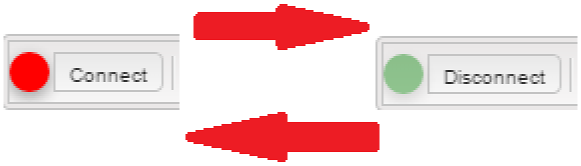
Related:

[Connect/Disconnect Buttons](180)

8.4.1 Connect/Disconnect Buttons

Clicking this button results in connection to or disconnection from the master module in the OEM product. The button appears in two ways depending upon whether there is a communications link between the product and your computer.

If the button appears as Connected, click it to disconnect. Conversely, if the button appears as Disconnected, you can click it to re-establish the connection. The button displays as follows:



Connect/Disconnect Button

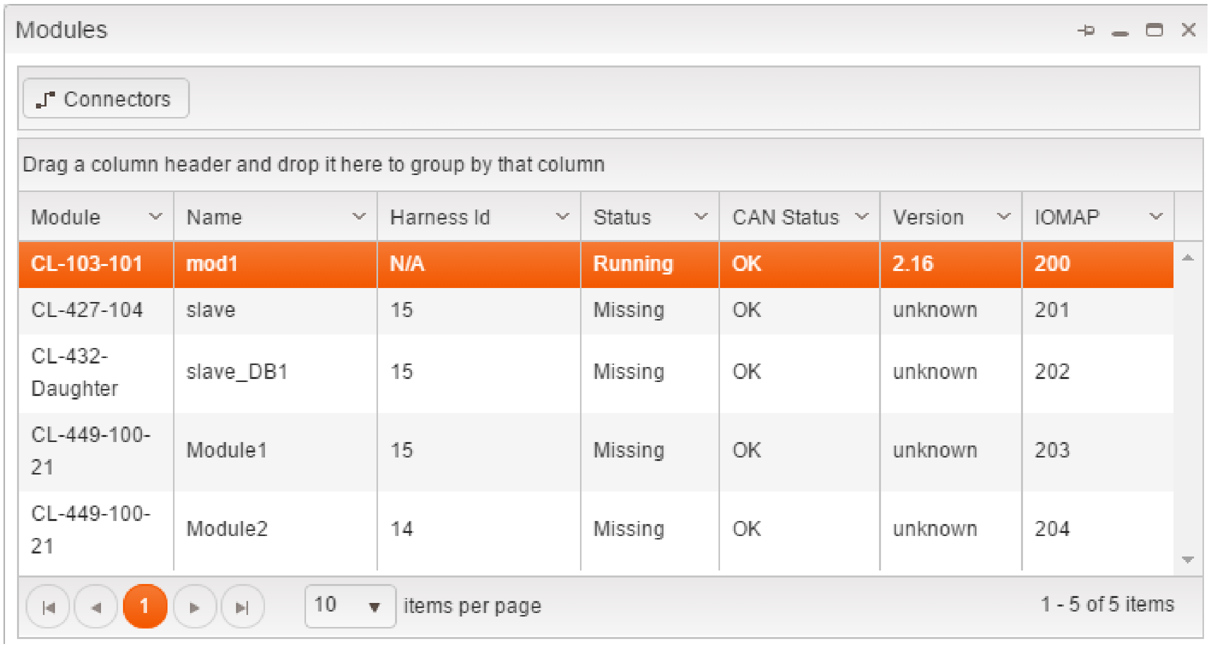
Related:

[Toolbar Buttons 179]
[Conductor Quick Start 302]
[Using Conductor 306]

8.5 Windows

8.5.1 Modules Window

The Modules window includes all CANLink® modules on your OEM product.The module, name, harness ID, status, CAN status, version, and IOMap can be observed from this window.



Drag a column header and drop it here to group by that column

Module	Name	Harness Id	Status	CAN Status	Version	IOMAP
CL-103-101	mod1	N/A	Running	OK	2.16	200
CL-427-104	slave	15	Missing	OK	unknown	201
CL-432-Daughter	slave_DB1	15	Missing	OK	unknown	202
CL-449-100-21	Module1	15	Missing	OK	unknown	203
CL-449-100-21	Module2	14	Missing	OK	unknown	204

items per page

1 - 5 of 5 items

Modules Window

Module

Module type

Name

Each module is given a name by the OEM and appears in this column.

Harness ID

A harness code is a number that is assigned to each individual module. To make it easier to provision a vehicle, harness codes are used to determine where a controller has been connected to on the vehicle.

Status

- **Running:** module is communicating correctly
- **Missing:** there is a communication problem with the module
- **Needs Config:** error, contact OEM
- **Needs Assign:** error, contact OEM
- **Wake up:** error, contact OEM
- **Comm Fail:** error, contact OEM

- **Soft Reset:** error, contact OEM
- **In Stub:** error, contact OEM
- **Unknown:** module may be Missing

Note: Viewing a modules status is the best starting point when troubleshooting a system problem. If a module is missing, all of its inputs and outputs will not be contributing to the control of the system. Ensuring proper communications between all system members is an imperative first step (check wiring and power).

CAN Status

- **OK:** system is communicating correctly
- **CAN1 Bad:** Dual CAN system only, utilizing the second CAN bus (CAN2), CAN1 is not functioning properly
- **CAN2 Bad:** Dual CAN system only, utilizing the first CAN bus (CAN1), CAN2 is not functioning properly
- **Unknown:** status unknown, module may be missing

Version

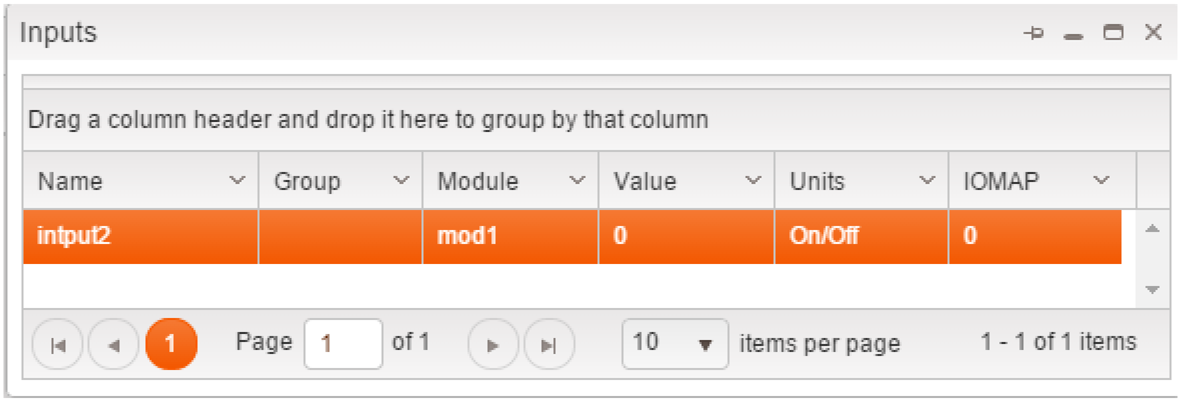
Each modules firmware is identified with a version number by the OEM and appears in this column.

IOMap

Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

8.5.2 Inputs Window

The Inputs window contains all the connected CANLink® module inputs for the OEM product. Within the Inputs window, the name, group, module, value, units, and IOMap can be observed.



Inputs

Drag a column header and drop it here to group by that column

Name	Group	Module	Value	Units	IOMAP
intput2		mod1	0	On/Off	0

Page 1 of 1 · 10 items per page · 1 - 1 of 1 items

Inputs Window

Name

Each input is given a name by the OEM and appears in this column.

Group

This is the group that the data item was assigned to in Orchestra®.

Module

This is the name given to the module.

Value

In this column, the values of all inputs will be displayed. It will either be a numeric value for analog inputs, or ON or OFF for digital inputs.

Units

Indicates how to interpret numeric values.

IOMap

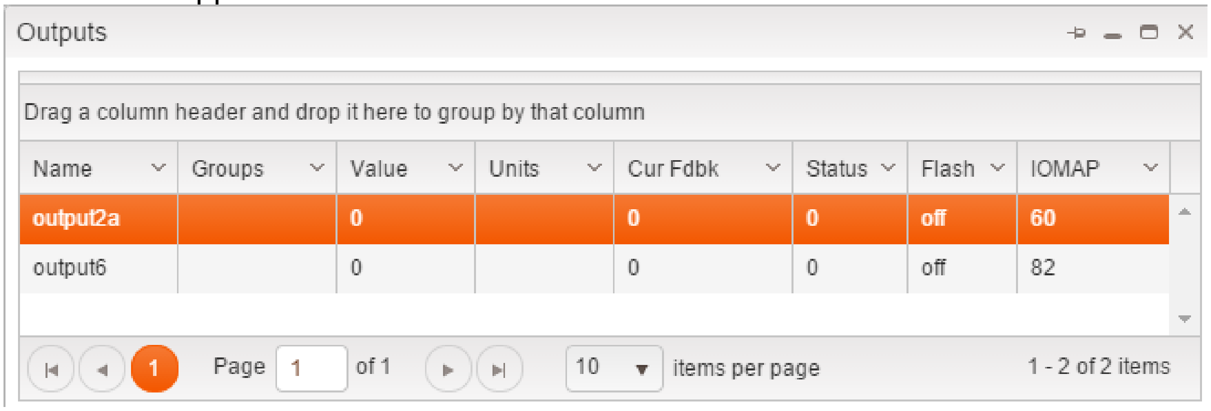
Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

Related:

Outputs Window 183

8.5.3 Outputs Window

The Outputs window has within it all the connected CANLink® module outputs for the OEM product. It does not include outputs from other devices not controlled by a CANLink® module. For all outputs listed within the Outputs window, the name, groups, value, units, current feedback, status, flash, and IOMap can be observed. In addition, current feedback and/or Status and/or Flash status may be observed if the hardware in the module supports it.



Drag a column header and drop it here to group by that column

Name	Groups	Value	Units	Cur Fdbk	Status	Flash	IOMAP
output2a		0		0	0	off	60
output6		0		0	0	off	82

Outputs

Name

Each output is given a name by the OEM and appears in this column.

Groups

This is the group that the data item was assigned to in Orchestra®.

Value

On/Off: If an output is digital, it will show either ON or OFF depending on its state.
PWM Duty Cycle: The value commanded is a percentage ranging from 0% to 100%.
Constant Current: The value commanded is a current value in mA.

Units

Used to give meaning to the values in the Values column.

Current Feedback (Cur Fdbk)

This column displays the current through the output, given in mA.

Status

The status shows how a particular output appears to the module it is on.

Normal: Output attached to normal load
Open: no load attached to output
Short GND: Output shorted to ground
Short BATT: Output shorted to battery
OverCurrent: load attached to output is drawing excess current
Grounded: servo on output is shorted to ground
STB or Open: Output is shorted to battery or open, undetermined
Under/Over Volt: voltage supplied to output is either above or below required voltage
Over Temp: Output is drawing excess current
STG or Open: Output is shorted to ground or open, undetermined

Flash

Output Flashing is a means of commanding an output to turn on and off automatically. If Flash is enabled, it will display On in this column.

IOMap

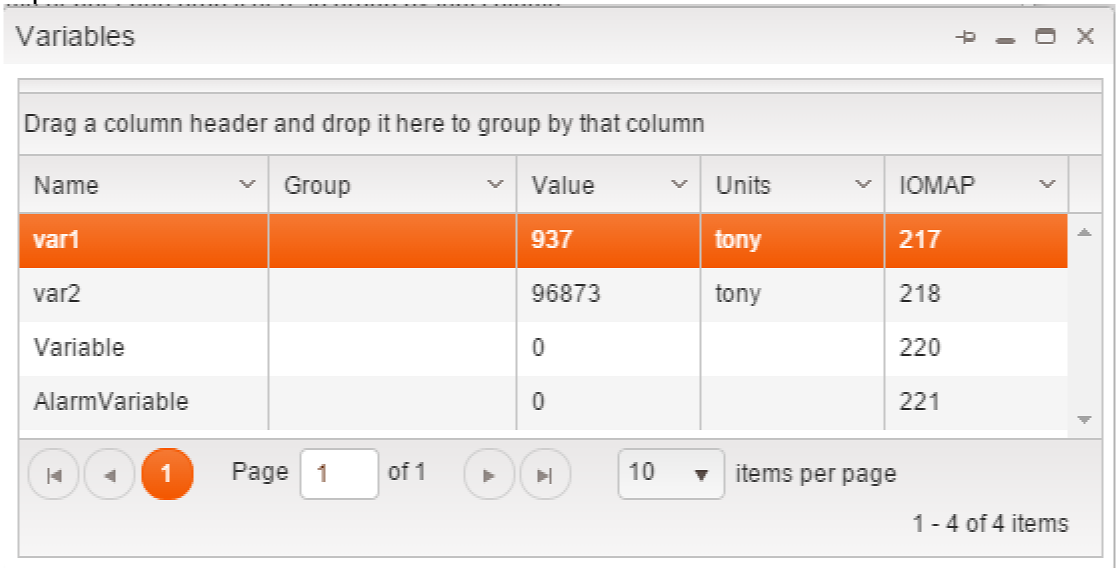
Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

Related:

Inputs Window 182

8.5.4 Variables Window

The Variables window is where all the modules variable information can be displayed. For all variables listed within the Variables window, the name, group, value, units, and IOMap can be observed.



Drag a column header and drop it here to group by that column

Name	Group	Value	Units	IOMAP
var1		937	tony	217
var2		96873	tony	218
Variable		0		220
AlarmVariable		0		221

Variable Window

Name

Each variable is given a name by the OEM and appears in this column.

Group

This is the group that the data item was assigned to in Orchestra®.

Value

Digital: ON or OFF
Other: Various numeric values

Units

Used to give meaning to the values in the Values column.

IOMap

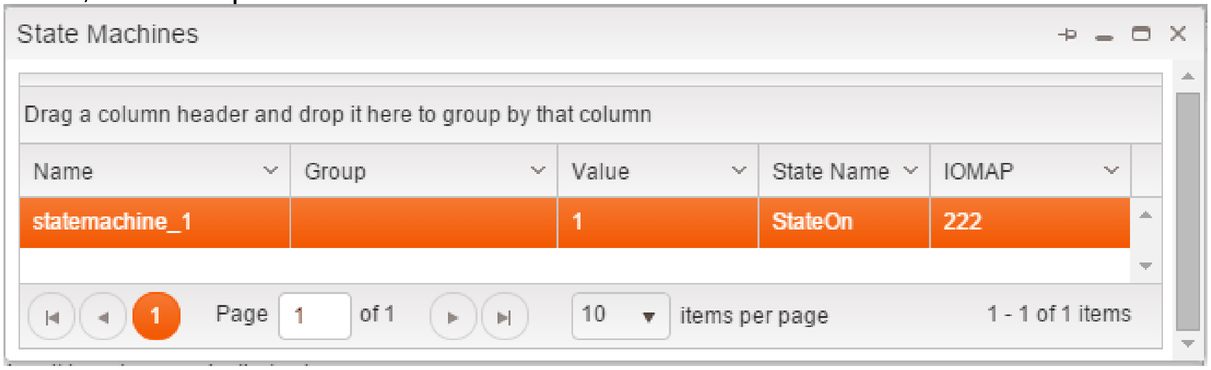
Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

Variables Information

Variables information can be obtained by double-clicking on the variable name. A separate information window will appear.

8.5.5 State Machines Window

The State Machines window displays the current value for the state machines. For all state machines listed within the State Machines window, the name, group, value, state name, and IOMap can be observed.


State Machines
Drag a column header and drop it here to group by that column
Page 1 of 1
10
items per page
1 - 1 of 1 items

Name	Group	Value	State Name	IOMAP
statemachine_1		1	StateOn	222

State Machines Window

Name

Each state machine is given a name by the OEM and appears in this column.

Group

This is the group that the data item was assigned to in Orchestra®.

Value

Number Range: from 0 - 255
Other: An enumeration of a specifically named value.

State Name

The state name is the name the developer gave the State Machine value that was created in Composer.

For example:

0 = Off

1 = On

IOMap

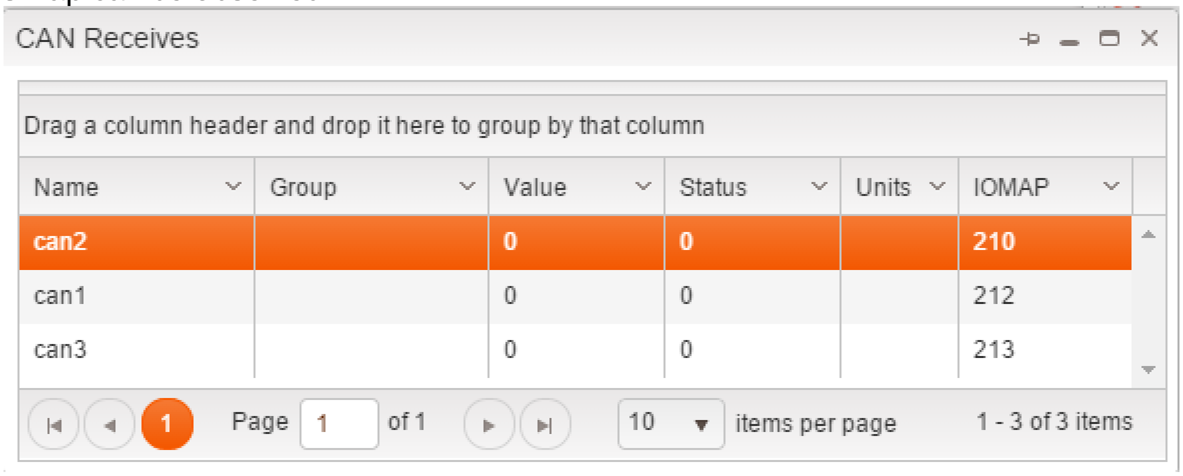
Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

State Machines Information

State Machines information can be obtained by double-clicking on the state machines name. A separate information window will appear.

8.5.6 CAN Receives Window

The CAN Receives window shows the current value received from a CAN message. For all CAN Receives listed within the window, the name, group, value, status, units, and IOMap can be observed.



CAN Receives

Drag a column header and drop it here to group by that column

Name	Group	Value	Status	Units	IOMAP
can2		0	0		210
can1		0	0		212
can3		0	0		213

Page 1 of 1 10 items per page 1 - 3 of 3 items

CAN Receives Window

Name

Each CAN Receive is given a name by the OEM and appears in this column.

Group

This is the group that the data item was assigned to in Orchestra®.

Value

Number Range from 0 - 4294967294

Units

Used to give meaning to the values in the Values column.

Status

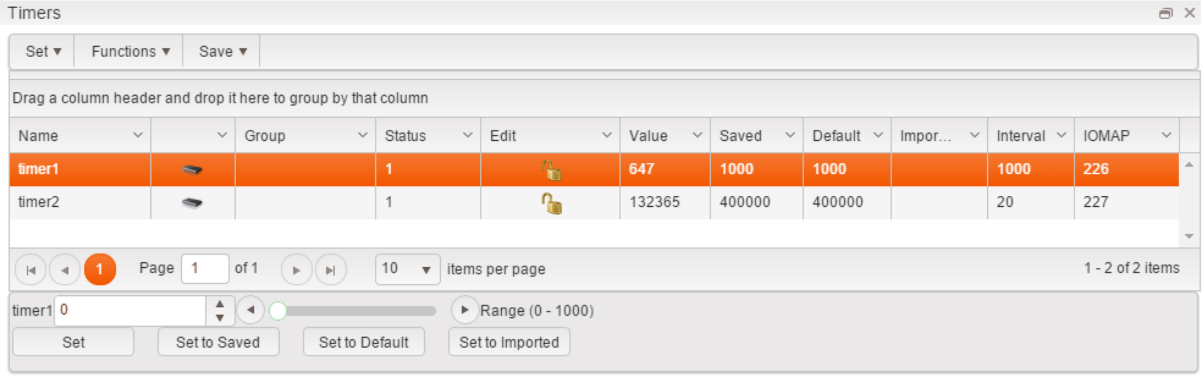
The status shows if a CAN Receives message has been cleared or received.

IOMap

Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

8.5.7 Timers Window

When the Timers function is selected, the Timers window opens.



Drag a column header and drop it here to group by that column

Name		Group	Status	Edit	Value	Saved	Default	Impor...	Interval	IOMAP
timer1	▬		1	🔓	647	1000	1000		1000	226
timer2	▬		1	🔓	132365	400000	400000		20	227

1 - 2 of 2 items

Range (0 - 1000)

Timers Window

WARNING

MASTER MODULE OVERRIDE HAZARD

Whenever Timers values are changed, the functionality of the system may be affected. Entering a wrong value may cause unpredictable OEM product behavior.

Death or serious injury can result.

Timers Window Features

This part of the Timers window lists eleven columns (Name, Save On Shutdown indication, Edit, Value, Saved, Default, Imported, Intervals, Status, and IOMAP) of timer settings installed on the OEM product. The Value column is the only column that can be edited or adjusted by the user. The users editing ability is limited by the OEM access level given for each individual timer. There are two drop-down menus to the right of the View/Sort label at the top of the Timers window, used for sorting.

Name

OEM defined name of the timer.

Edit

This column is used to display the users editing ability for each timer. If a lock icon displays in this column, your OEM-supplied user settings do not allow you to edit this timer.

Group

This is the group that the data item was assigned to in Orchestra®.

Value (Access Level Limited Only)

This column shows the actual timer values that are being read by the Conductor™. These are the values that are read from the temporary EEPROMs locations in memory. It is also the only column that can be edited or adjusted by the user. To enter a new value for the active timer, place the cursor in the Value window below and enter the new value and click the Set button (hitting the ENTER key will also place the cursor in the Value window).

Saved

This column will display all values that are currently being stored in the processors EEPROM.

Default

This column will display the OEM-defined default timer values and cannot be changed. All values in the Value column will be reset to the default values from this column when either the Set to Defaults or Set All to Defaults buttons are selected.

Warning

If you restore the OEM system to defaults, your system will return to an un-tuned state where all calibrations will be lost.

Imported

This column will display the timer data that has been imported from a separate file.

Intervals

OEM defined timed Interval that the timer uses for timing data.

Status

Displays the Status (RUN or PAUSE) of the timer.

IOMap

Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

Related:

Edit Individual Timers 191
Edit All Writable Timers 193
Timers Functions 194
Timers Information 196

8.5.7.1 Edit Individual Timers

This part of the Timers window is used to temporarily edit the value of a specific timer.

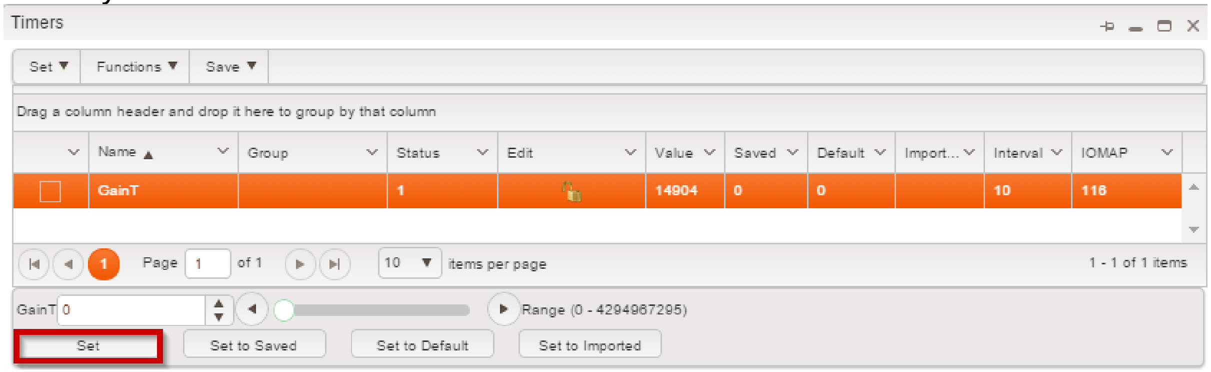
Value and Range Fields

The Value field is the only active field that the user can edit. Either type a new value in the open box or use the **Value Up (^)** or **Value Down (v)** button to change the setting in one unit increments. Click the Set button 191 to enter the new values into the active Timers Value column.

The Range label boxes are always grayed out and cannot be changed. The Range label boxes display the range of values that are allowed for the selected timers settings.

SET Button

Clicking this button will immediately enter the new value adjustment made to the Value column of the active timer selected. Clicking SET will only change the temporary memory location of this timer.



	Name ▲	Group	Status	Edit	Value	Saved	Default	Import...	Interval	IOMAP
□	GainT		1		14904	0	0		10	116

GainT 0 Range (0 - 4294967295)

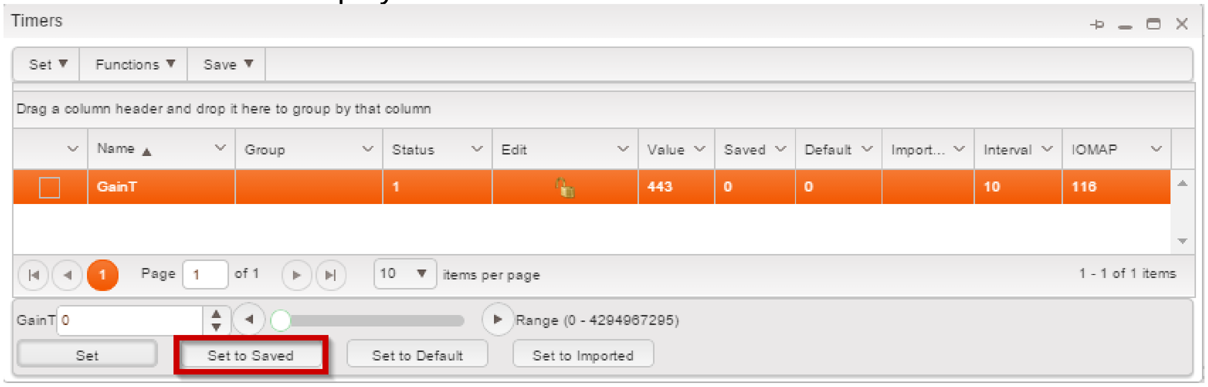
Set Button

Once a new value is entered into the Value column, it will cause the system to operate with the new value. The user can keep entering a new value and SET it into the Value column as many times as necessary to achieve the desired result. Every time a new value is entered, the system will run using the new values.

In this way, a system can be tweaked quickly without committing or saving the trial and error values. Once you are satisfied with the new value, save the new settings by clicking the **Save All** button. If the new values are not saved, the next time the modules are reset the new values will be lost and the old saved values will be reloaded. Clicking Save ALL will change the processor's EEPROM memory location.

Set to Saved Button

Clicking this button will immediately reset the Value column of the active timers selected to the values that are displayed in the Saved column.



Drag a column header and drop it here to group by that column

	Name ▲	Group	Status	Edit	Value	Saved	Default	Import...	Interval	IOMAP
□	GainT		1		443	0	0		10	116

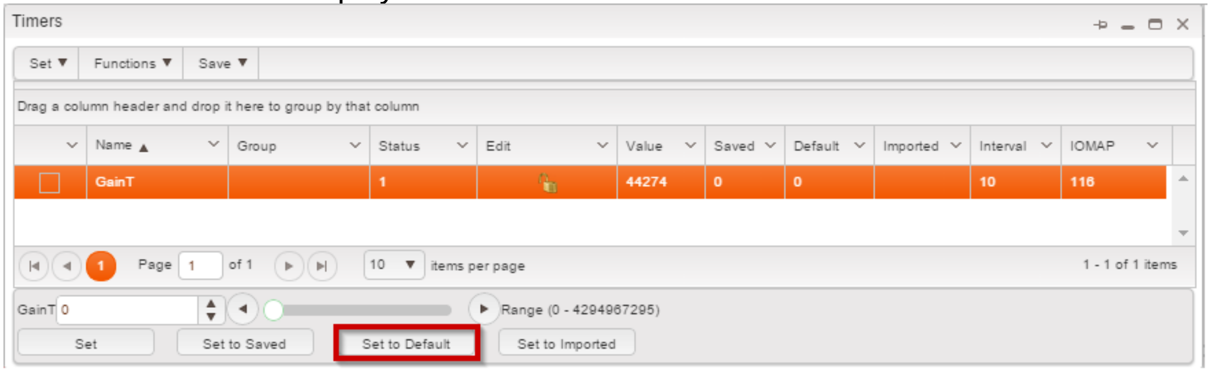
1 - 1 of 1 items

Range (0 - 4294967295)

Set to Saved Button

Set to Default Button

Clicking this button will immediately reset the Value column of the active timers selected to the values that are displayed in the Default column.



Drag a column header and drop it here to group by that column

	Name ▲	Group	Status	Edit	Value	Saved	Default	Imported	Interval	IOMAP
□	GainT		1		44274	0	0		10	116

1 - 1 of 1 items

Range (0 - 4294967295)

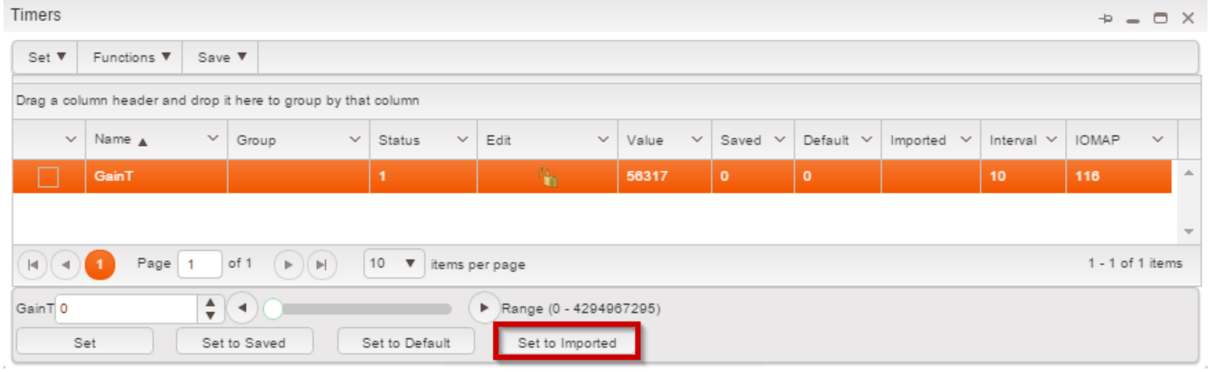
Set to Default Button

WARNING!

MASTER MODULE OVERRIDE HAZARD. Whenever Timers values are reset to their default settings, the functionality of the OEM product system may be affected and cause unpredictable behavior. Death or serious injury can result.

Set to Imported Button

Clicking this button immediately resets the Value column of the active timers selected to the values that are displayed in the Imported column.



Set to Imported Button

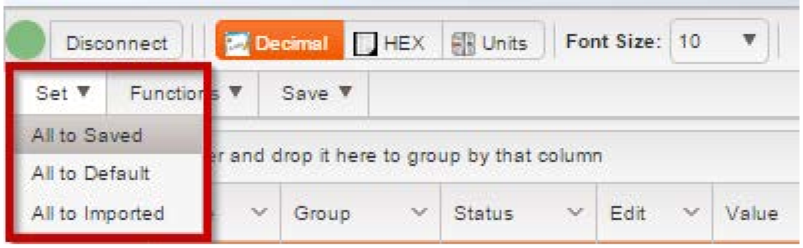
Related:

Edit All Writable Timers 193

8.5.7.2 Edit All Writable Timers

Edit All Writable Timers (Access Level Limited Only)

This part of the Timers window is used to edit the values of all active writable timers at once. If these four buttons remain grayed out after selecting a timer, the OEM-supplied user settings do not allow the user to use these features.



Set Menu

Save ALL Button

Clicking this button immediately enters the new value adjustments made into the Value column of all active timers and also changes the processors EEPROM memory location.

If the new values are not saved, the next time the modules are reset the new values will be lost and the old saved values will be reloaded.

Set ALL to Saved Button

Clicking this button immediately resets all active timer values to the values that are displayed in the Saved column.

Set ALL to Defaults Button

Clicking this button immediately resets all active timer values to the values that are displayed in the Default column.

WARNING! MASTER MODULE OVERRIDE HAZARD.

Whenever Timers values are reset to their default settings, the functionality of the OEM product system may be affected and cause unpredictable behavior. Death or serious injury can result.

Set ALL to Imported Button

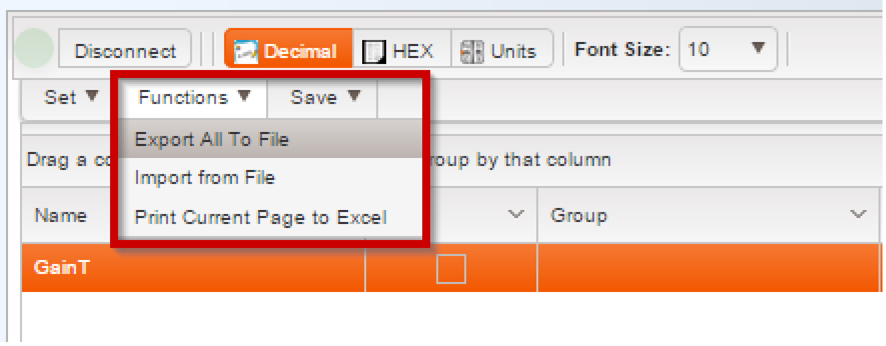
Clicking this button immediately resets all active timer values to the values that are displayed in the Imported column.

Related:

Edit Individual Timers 191

8.5.7.3 Timers Functions

This part of the Timers window is used to import, export or print the values of all active timers. If these two buttons remain grayed out after selecting a timer, the OEM-supplied user settings do not allow the user to use these features.



Timers Functions

Export to File Button

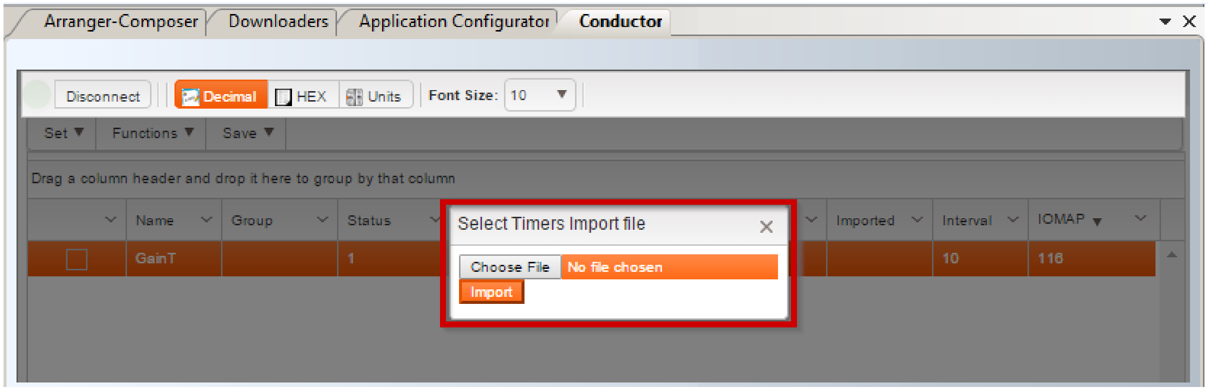
Clicking this button automatically saves the current timers values to a file that will reside in the Conductor™ software folder. This file can be used later for reference or for importing these settings back into the Conductor™.

Warning:

Exporting to a file may cause communication problems between the Conductor™ and the module. It may be necessary to reconnect to the module after an Export to File action.

Import from File Button

Clicking this button opens a dialog box prompting the user to locate and select a previously saved timers values file. Selecting this file will allow the Conductor to then read timers values from the new file.



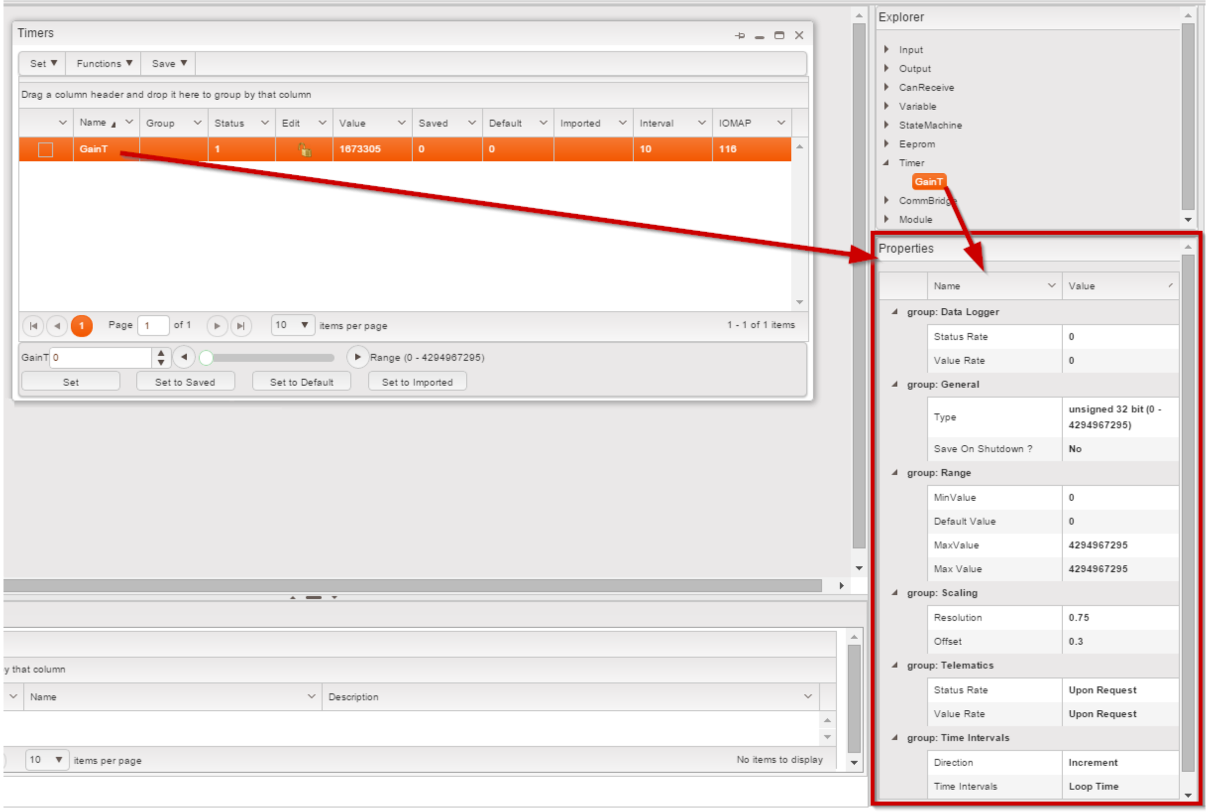
Import Timers from File

Print Current Page to Excel

This exports the current page's data to an Excel file.

8.5.7.4 Timers Properties

Timer properties information can be obtained by double-clicking on the timer name, either in the Timer window 189 or Explorer pane 177. The timer properties open in the Properties pane 178.



	Name	Group	Status	Edit	Value	Saved	Default	Imported	Interval	IOMAP
□	GainT		1		1673305	0	0		10	116

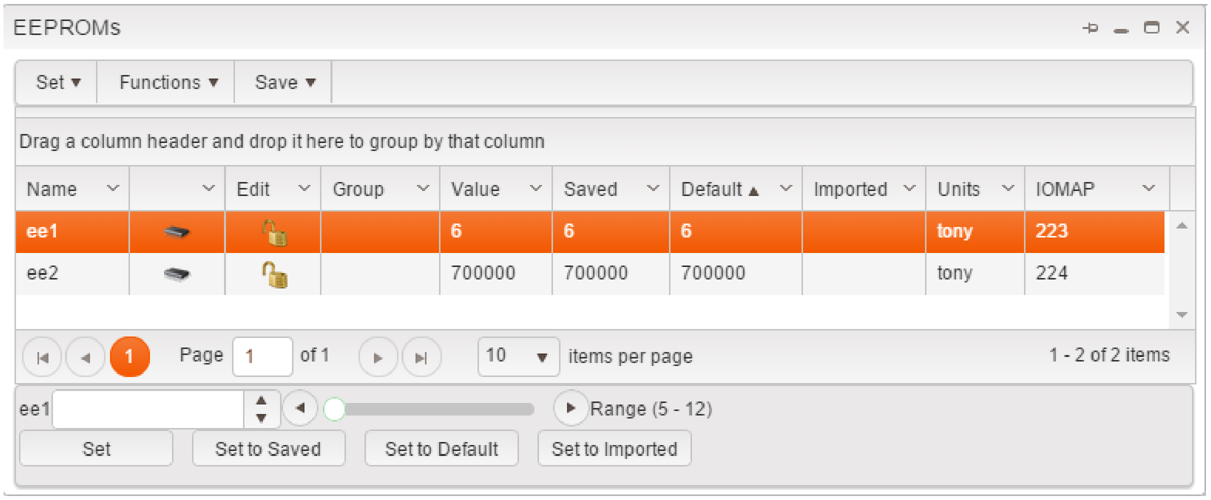
GainT 0 Range (0 - 4294967295)

Name	Value
group: Data Logger	
Status Rate	0
Value Rate	0
group: General	
Type	unsigned 32 bit (0 - 4294967295)
Save On Shutdown ?	No
group: Range	
MinValue	0
Default Value	0
MaxValue	4294967295
Max Value	4294967295
group: Scaling	
Resolution	0.75
Offset	0.3
group: Telematics	
Status Rate	Upon Request
Value Rate	Upon Request
group: Time Intervals	
Direction	Increment
Time Intervals	Loop Time

Timers Properties

8.5.8 EEPROMS Window

When the EEPROMs function is selected, the EEPROMs window opens.



Name		Edit	Group	Value	Saved	Default ▲	Imported	Units	IOMAP
ee1				6	6	6		tony	223
ee2				700000	700000	700000		tony	224

EEPROMS Window

WARNING
MASTER MODULE OVERRIDE HAZARD
Whenever EEPROMs values are changed, the functionality of the system may be affected. Entering a wrong value may cause unpredictable OEM product behavior.
Death or serious injury can result.

EEPROMS Window Features

This part of the EEPROMs window lists eight columns (Name, Save On Shutdown indication, Edit, Value, Saved, Default, Imported and Units) of EEPROM settings installed on the OEM product. The Value column is the only column that can be edited or adjusted by the user. The users editing ability is limited by their OEM access level given for each individual EEPROM. There are two drop-down menus to the right of the View/Sort label at the top of the EEPROMs window, used for sorting.

Name Column

OEM-defined Name of the EEPROMs.

Edit Column

This column is used to display the users editing ability for each EEPROM. If a lock icon displays in this column, your OEM-supplied user settings do not allow you to edit this EEPROM.

Group

This is the group that the data item was assigned to in Orchestra®.

Value Column (Access Level Limited Only)

This column shows the actual EEPROMs values that are being read by the Conductor™. These are the values that are read from the temporary EEPROMs locations in memory. It is also the only column that can be edited or adjusted by the user. To enter a new value for the active EEPROM, place the cursor in the Value window below, enter the new value and click the SET button (hitting the ENTER key will also place the cursor in the Value window).

Saved Column

This column will display all values that are currently being stored in the processors EEPROM.

Default Column

This column will display the OEM-defined default EEPROM values and cannot be changed. All values in the Value column will be reset to the default values from this column when either the **Set to Defaults** or **Set All to Defaults** buttons are selected.

IMPORTANT:

If you restore the OEM system to defaults, your system will return to an un-tuned state where all calibrations will be lost.

Imported Column

This column will display the EEPROMs data that has been imported from a separate file.

Units Column

Displays the units of the EEPROMs.

IOMap

Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

Related:

Edit Individual EEPROM Value 199
Edit All EEPROM Values that are Editable 201
EEPROM Functions 203
EEPROM Information 204

8.5.8.1 Edit Individual EEPROM Value

This part of the EEPROMs window is used to temporarily edit the value of a specific EEPROM.

Value and Range Fields

The Value field is the only active field that users can edit. Either type a new value in the open box or use the **Value Up (^)** or **Value Down (v)** button to change the setting in one unit increments. Click the **Set** button to enter the new values into the active EEPROMs Value column. The Range label boxes are always grayed out and cannot be changed.

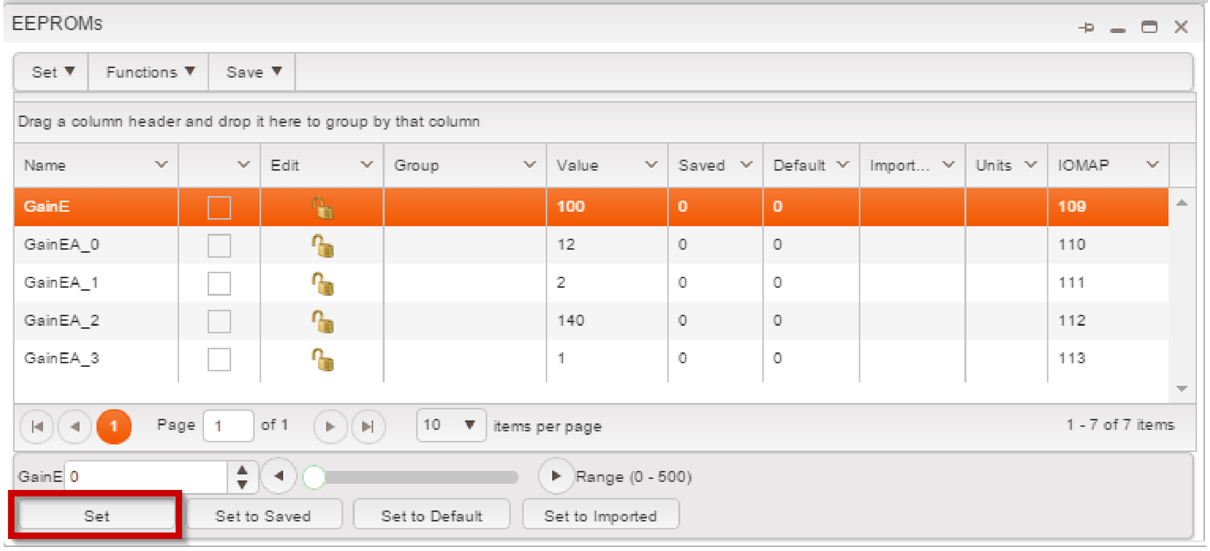
The Range label boxes display the range of values that are allowed for the selected EEPROM settings.

SET Button

Clicking this button will immediately enter the new Value adjustment made to the Value column of the active EEPROM selected. Clicking SET will only change the temporary memory location of this EEPROM.

Once a new value is entered into the Value column, it will cause the system to operate with the new Value. Users can keep entering a new Value and SET it into the Value column as many times as necessary to achieve the desired result. Every time a new value is entered, the system will run using the new values.

In this way, a system can be tweaked quickly without committing or saving the trial and error values. Once users are satisfied with the new value, save the new settings by clicking the **Save ALL** button. If the new values are not saved, the next time the modules are reset the new values will be lost and the old saved values will be reloaded. Clicking Save ALL will change the processors EEPROM memory location.



Drag a column header and drop it here to group by that column

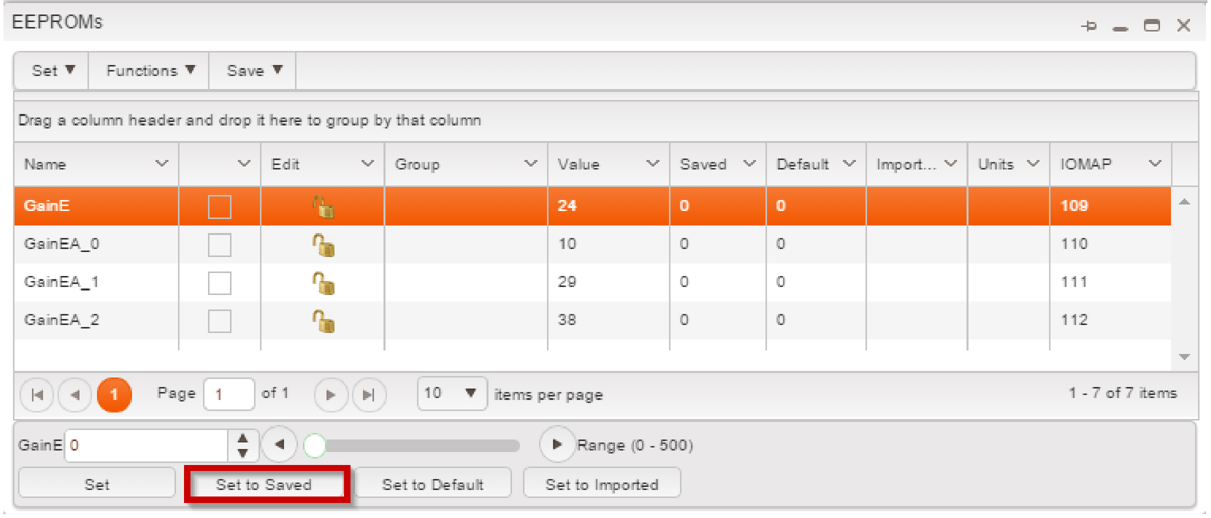
Name		Edit	Group	Value	Saved	Default	Import...	Units	IOMAP
GainE	□	🔓		100	0	0			109
GainEA_0	□	🔓		12	0	0			110
GainEA_1	□	🔓		2	0	0			111
GainEA_2	□	🔓		140	0	0			112
GainEA_3	□	🔓		1	0	0			113

GainE 0

Set Button

Set to Saved Button

Clicking this button will immediately reset the Value column, of the active EEPROMs selected to the values that are displayed in the Saved column.



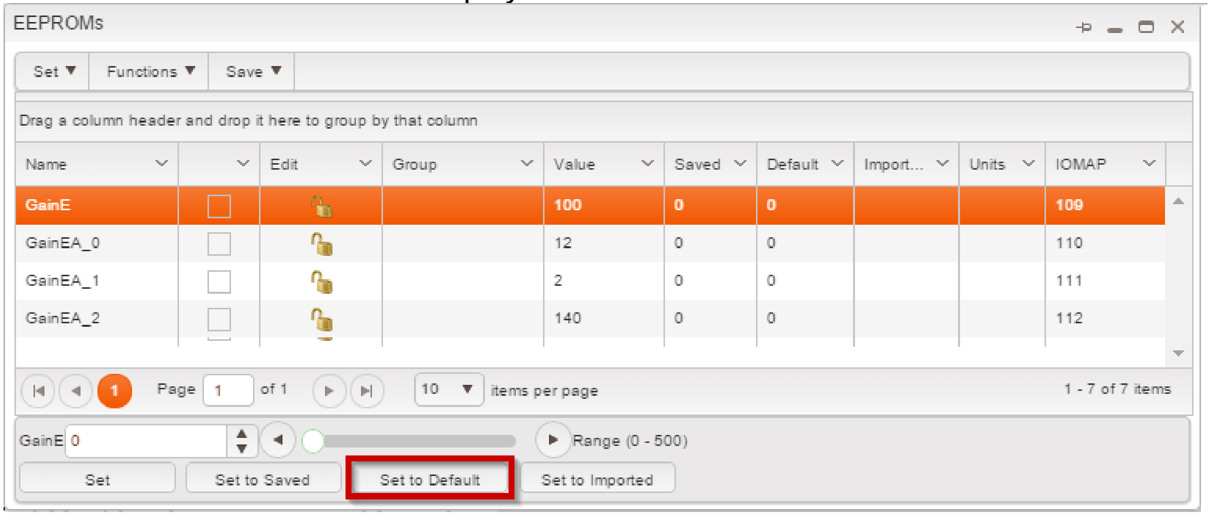
Name		Edit	Group	Value	Saved	Default	Import...	Units	IOMAP
GainE	□			24	0	0			109
GainEA_0	□			10	0	0			110
GainEA_1	□			29	0	0			111
GainEA_2	□			38	0	0			112

GainE 0 Range (0 - 500)

Set to Saved Button

Set to Default Button

Clicking this button will immediately reset the Value column of the active EEPROMs selected to the values that are displayed in the Default column.



Name		Edit	Group	Value	Saved	Default	Import...	Units	IOMAP
GainE	□			100	0	0			109
GainEA_0	□			12	0	0			110
GainEA_1	□			2	0	0			111
GainEA_2	□			140	0	0			112

GainE 0 Range (0 - 500)

Set to Default Button

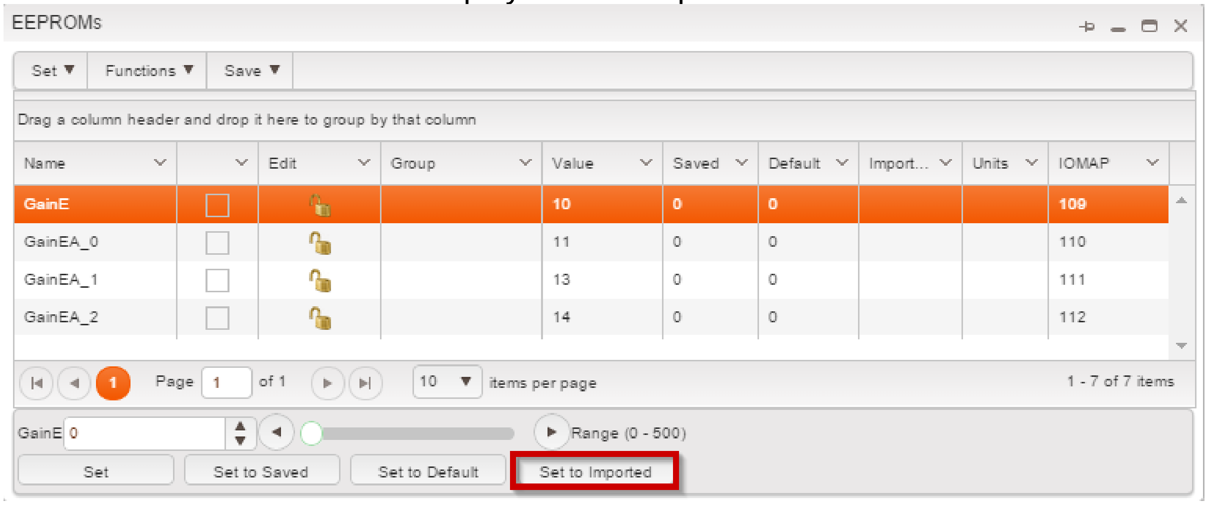
WARNING!

MASTER MODULE OVERRIDE HAZARD.

Whenever EEPROMs values are reset to their default settings, the functionality of the OEM product system may be affected and cause unpredictable behavior. Death or serious injury can result.

Set to Imported Button

Clicking this button will immediately reset the Value column of the active EEPROMs selected to the values that are displayed in the Imported column.



Name		Edit	Group	Value	Saved	Default	Import...	Units	IOMAP
GainE	□	🔓		10	0	0			109
GainEA_0	□	🔓		11	0	0			110
GainEA_1	□	🔓		13	0	0			111
GainEA_2	□	🔓		14	0	0			112

Page 1 of 1 · 10 items per page · 1 - 7 of 7 items

GainE 0 · Range (0 - 500)

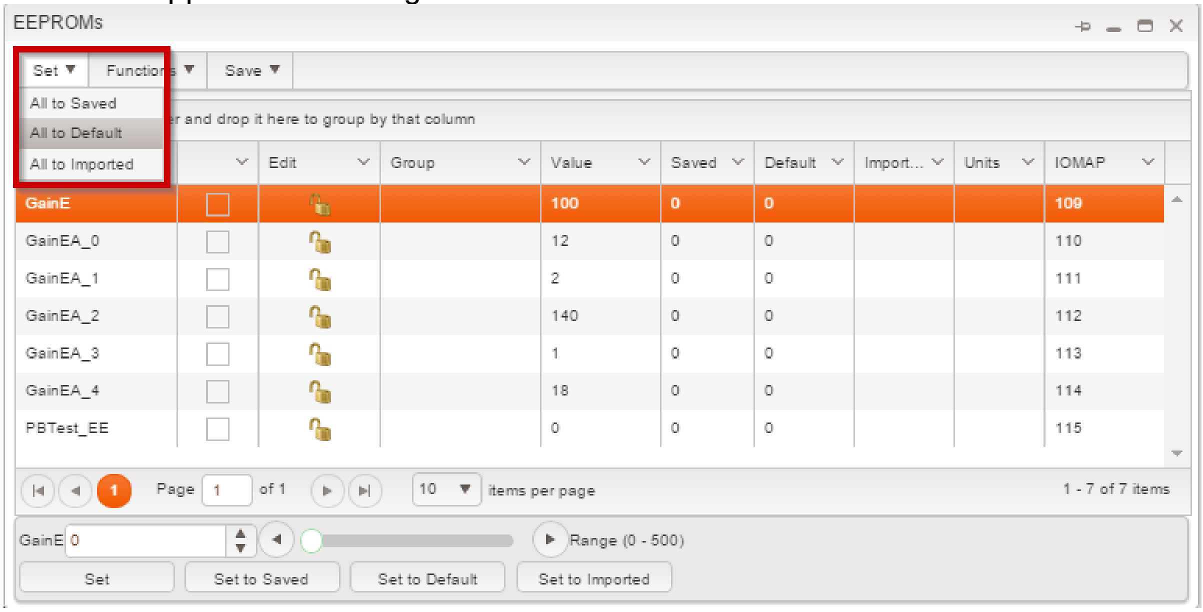
Set to Imported Button

Related:

Edit All EEPROM Values that are Editable 201

8.5.8.2 Edit All EEPROM Values that are Editable

This part of the EEPROMs window is used to edit the values of all active writable EEPROMs at once. If these four buttons remain grayed out after selecting an EEPROM, the OEM-supplied user settings do not allow the user to use these features.



		Edit	Group	Value	Saved	Default	Import...	Units	IOMAP
GainE	□	🔓		100	0	0			109
GainEA_0	□	🔓		12	0	0			110
GainEA_1	□	🔓		2	0	0			111
GainEA_2	□	🔓		140	0	0			112
GainEA_3	□	🔓		1	0	0			113
GainEA_4	□	🔓		18	0	0			114
PBTest_EE	□	🔓		0	0	0			115

Page 1 of 1 · 10 items per page · 1 - 7 of 7 items

GainE 0 · Range (0 - 500)

Set Menu

Save ALL Button

Clicking this button will immediately enter the new value adjustments made into the Value column of all active EEPROMs and also changes the processors EEPROM memory location. If the new values are not saved, the next time the modules are reset the new values will be lost and the old saved values will be reloaded.

Set ALL to Saved Button

Clicking this button will immediately reset all active EEPROM values to the values that are displayed in the Saved column.

Set ALL to Defaults Button

During system development, a set of best guess values for each EEPROM is determined and used for factory defaults. The values are nominal and are intended as a basis from which to start the tuning process. At times, should the tuning process go awry, click this button to return all values in the Value column to the default settings.

Clicking this button will immediately reset all active EEPROM values to the values that are displayed in the Default column.

WARNING! MASTER MODULE OVERRIDE HAZARD.

Whenever EEPROMs values are reset to their default settings, the functionality of the OEM product system may be affected and cause unpredictable behavior. Death or serious injury can result.

Set ALL to Imported Button

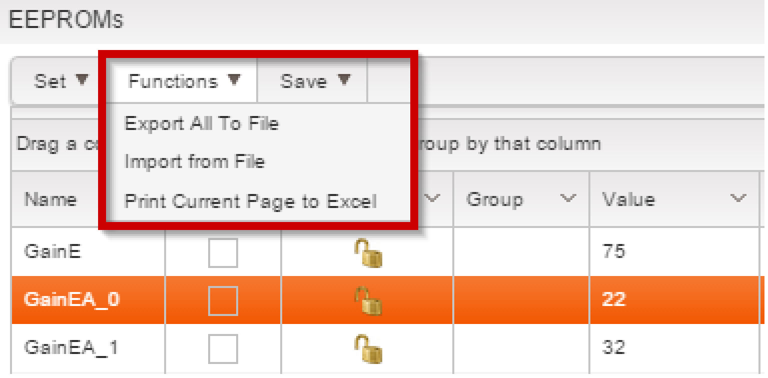
Clicking this button will immediately reset all active EEPROM values to the values that are displayed in the Imported column.

Related:

Edit Individual EEPROM Value 199

8.5.8.3 EEPROM Functions

This part of the EEPROMs window is used to import, export or print the values of all active EEPROMs. If these two buttons remain grayed out after selecting an EEPROM, the OEM-supplied user settings do not allow users to use these features.



Name			Group	Value
GainE	□			75
GainEA_0	□			22
GainEA_1	□			32

EEPROMS Functions Menu

Export to File Button

Clicking this button will automatically save the current EEPROMs values to a file that will reside in the Conductor™ software folder. This file can be used later for reference or for importing these settings back into the Conductor™.

IMPORTANT:

Exporting to a file may cause communication problems between the Conductor™ and the module. It may be necessary to reconnect to the module after an Export to File action.

Import from File Button

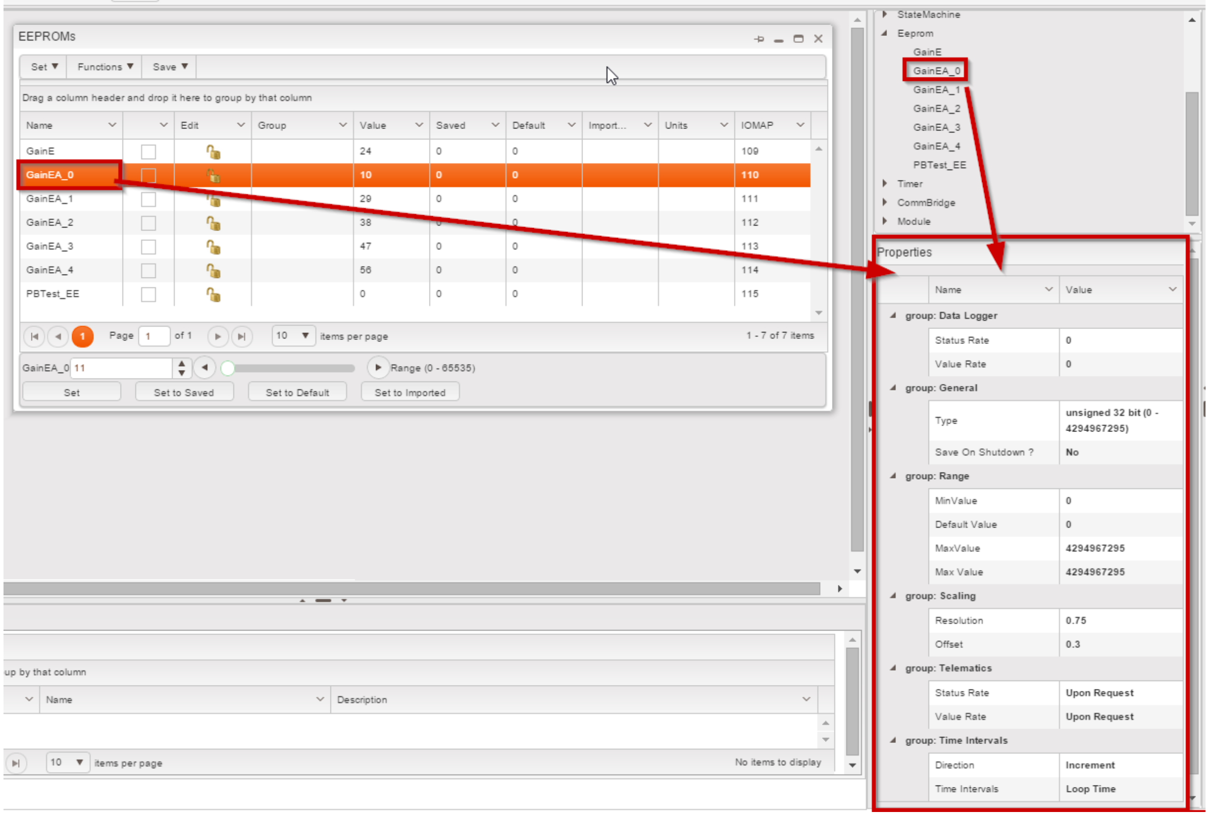
Clicking this button will open a dialog box prompting the user to locate and select a previously saved EEPROMs values file. Selecting this file will allow the Conductor to then read EEPROMs values from the new file.

Print Current Page to Excel

This exports the current page's data to an Excel file.

8.5.8.4 EEPROM Properties

EEPROM information can be obtained by double-clicking on the EEPROM name, either in the EEPROM window or the Explorer pane. The EEPROM properties open in the Properties pane 178.



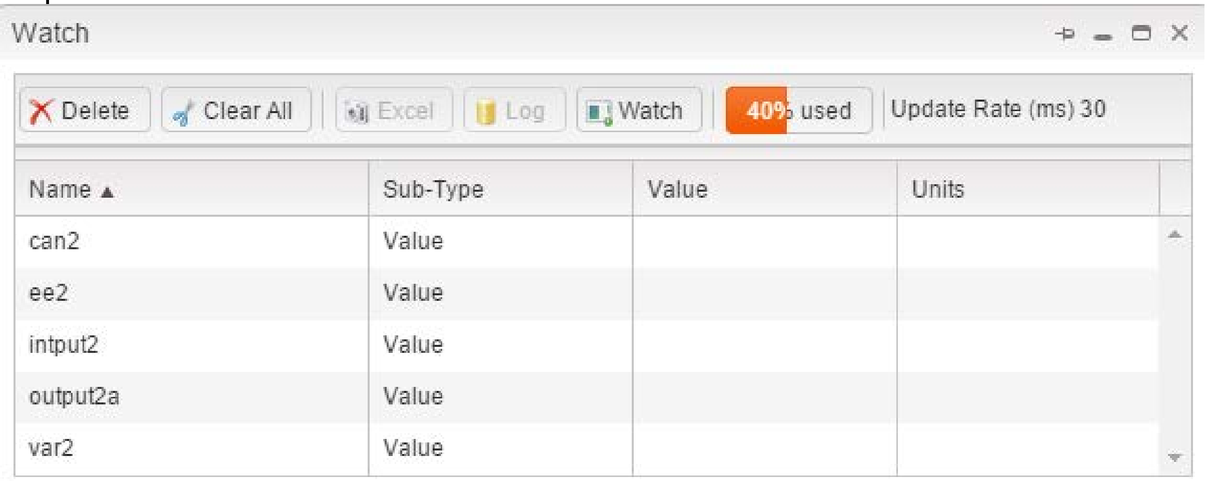
Name		Edit	Group	Value	Saved	Default	Import...	Units	IOMAP
GainE	□			24	0	0			109
GainEA_0	□			10	0	0			110
GainEA_1	□			29	0	0			111
GainEA_2	□			38	0	0			112
GainEA_3	□			47	0	0			113
GainEA_4	□			56	0	0			114
PBTest_EE	□			0	0	0			115

Name	Value
group: Data Logger	
Status Rate	0
Value Rate	0
group: General	
Type	unsigned 32 bit (0 - 4294967295)
Save On Shutdown ?	No
group: Range	
MinValue	0
Default Value	0
MaxValue	4294967295
Max Value	4294967295
group: Scaling	
Resolution	0.75
Offset	0.3
group: Telematics	
Status Rate	Upon Request
Value Rate	Upon Request
group: Time Intervals	
Direction	Increment
Time Intervals	Loop Time

EEPROM Properties

8.5.9 Watch Window

The Watch window works best for large systems with many inputs, outputs, variables, etc., where it is inconvenient to monitor several data items simultaneously within their respective data windows.



Update Rate (ms) 30

Name ▲	Sub-Type	Value	Units
can2	Value		
ee2	Value		
intput2	Value		
output2a	Value		
var2	Value		

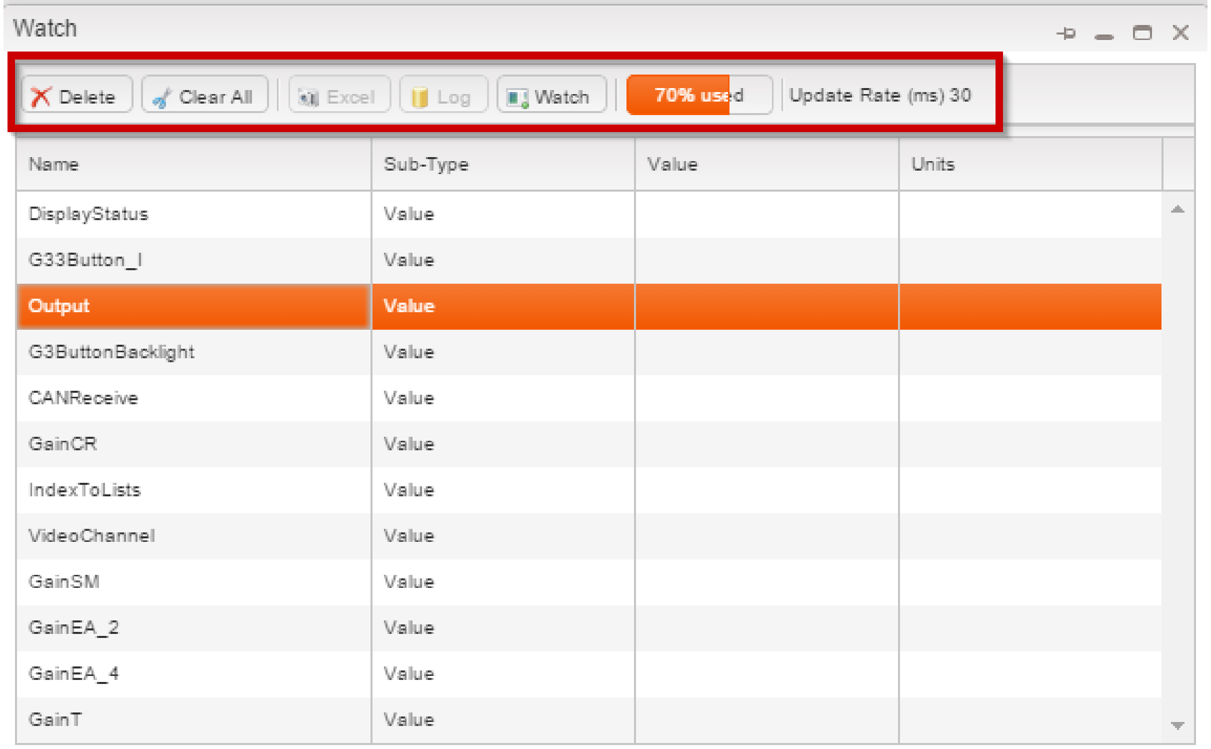
Watch Window

In the Watch window, the user will be able to select up to 20 memory locations of data items to watch. The master module that is selected for the project will determine the number of memory locations for data items that are available to watch. Data Items take one or two memory locations based on their configuration at design time. As data items are added to the Watch window, notice that the Memory Used bar indicates how much memory has been used.

View Watch Window Button

Clicking this button will open the Watch window. The Watch window can also be opened from the main window View menu by selecting Watch (Key Command Shortcut - Ctrl-W) from the drop-down menu.

Watch Window Controls



Name	Sub-Type	Value	Units
DisplayStatus	Value		
G33Button_I	Value		
Output	Value		
G3ButtonBacklight	Value		
CANReceive	Value		
GainCR	Value		
IndexToLists	Value		
VideoChannel	Value		
GainSM	Value		
GainEA_2	Value		
GainEA_4	Value		
GainT	Value		

Watch Window Controls

DELETE Button

Clicking the **Delete** button will allow the user to delete an individual data item from the Watch window list.

Clear All Button

Clicking the **Clear All** button will allow the user to remove all data items from the Watch window with one button push.

Excel Button

This button saves and exports the data as an Excel file.

Log Button

IMPORTANT:

Before you click this button, make sure you want to remove all data items from the Watch window list, as there is no undo.

Watch Button

Clicking the **Watch** button requests that the master module send data to the Conductor™ about the active data items in the Watch window list. The user will then be able to view real-time values from all active data items.

Memory Used

This displays the memory used in an orange progress bar and also lists the total number of data items that have already been added to the Watch window. The maximum number of data items that can be added to the Watch window is 20.

8.5.10 Debug Window

When the Debug function is selected, the main Debug window opens.

WARNING

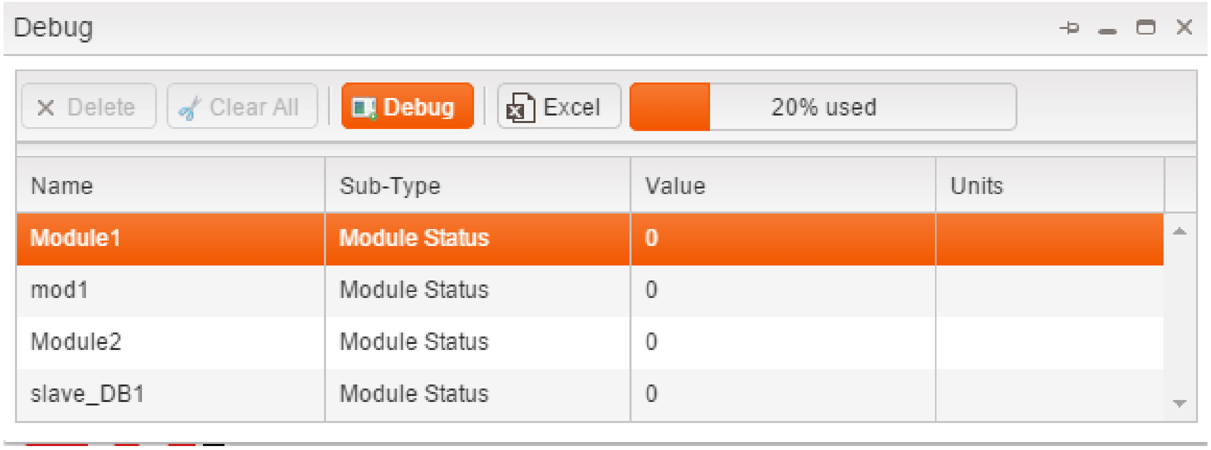
MASTER MODULE OVERRIDE HAZARD

Whenever Debug is activated, you are assuming direct control of selected Data Items value and potentially bypassing safety interlocks and normal operation.

Entering a wrong value may cause unpredictable OEM product behavior. Death or serious injury can result.

In the Debug window, the user is able to select up to 20 data items to debug. The master module that is selected for the project will determine the number of data items that are available to debug. It allows for artificial manipulation of inputs, outputs, variables, EEPROMs, timers, CAN Receives and State Machines. The window displays name, sub-type, value, and units of all data items added to the Debug window.

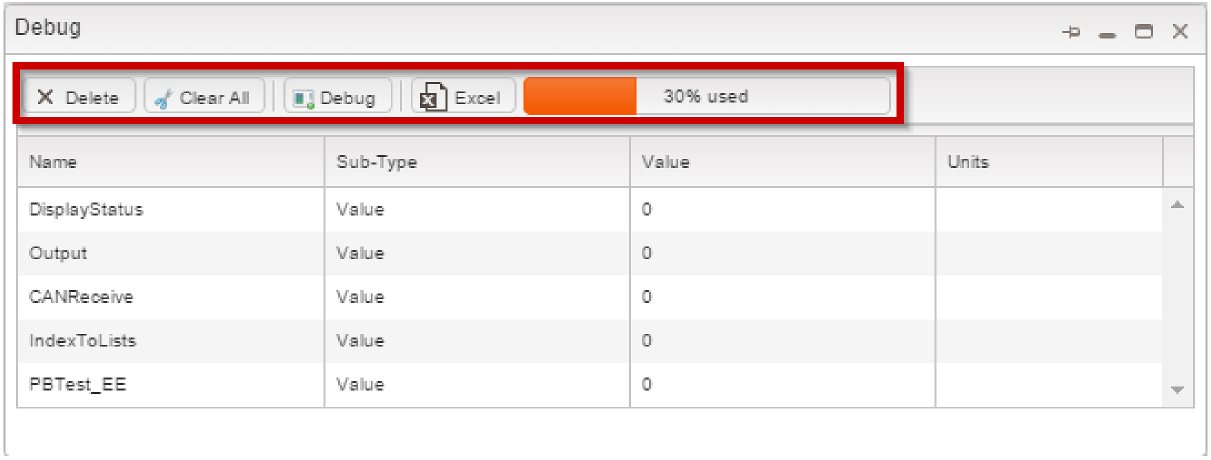
Use the Debug window to override the value of a data item to aid with troubleshooting it.



Name	Sub-Type	Value	Units
Module1	Module Status	0	
mod1	Module Status	0	
Module2	Module Status	0	
slave_DB1	Module Status	0	

Debug Window

Debug Window Controls



Name	Sub-Type	Value	Units
DisplayStatus	Value	0	
Output	Value	0	
CANReceive	Value	0	
IndexToLists	Value	0	
PBTest_EE	Value	0	

Debug Window Controls

DELETE Button

Clicking the **Delete** button will allow the user to delete an individual data item from the Debug window list.

Clear All Button

Clicking the **Clear All** button will allow the user to remove all data items from the Debug window with one button push.

Warning

Before you click this button, make sure you want to remove all Data Items from the Debug window list, as there is no undo.

Debug Button

Clicking the **Debug** button will request that the master module send current data to the Conductor™ about the active data items in the Debug window list. The user will then be able to:

- Turn specific Data Items ON or OFF
- Adjust real-time values from all active data items
- Run or Pause Timers
- Clear or Receive data from CAN bus
- Apply hypothetical conditions to outputs (Open, Short GND, Short BATT, OverCurrent, Grounded, STB or Open, Under/Over Volt, Over Temp and STG or Open)

Excel Button

This button saves and exports the data as an Excel file.

Memory Used

This displays the memory used in an orange progress bar and also lists the total number of data items that have already been added to the Debug window. The maximum number of data items that can be added to the Debug window is 20.

8.5.10.1 Debug Data Item Types

8.5.10.1.1 Input Types

Inputs typically provide run-time information to the master module of the OEM products system. The process for debugging Input type data items is essentially all the same.

Input types that are currently supported by the Conductor™ are:

- Inputs
- Variables
- State Machines
- EEPROMs
- Timers
- CAN Receives

For all input types, the Debug window displays information on the name, sub-type, value, and units.

Note: Make all initial Input Value Control adjustments before entering the Debug mode.

Related:

Output Types 210

8.5.10.1.2 Output Types

Outputs typically provide read/write value information to and from the master module of the OEM products system. The process for debugging an output type is more complicated than debugging inputs. The complexity comes into play when considering four different Sub-Type options for debugging outputs. These sub-types are Value, Status, Flash and Current .

Output Sub-Type options are selected in the Multi-View window. These Sub-Types will be based on whether the output has been set up for Output Status, Output Flash or Current Feedback, and/or your level of access privileges to the Conductor™ software. Check-boxes for any Sub-Type that are grayed out mean that the Sub-Type is not available.

Once all outputs and Sub-Type options have been selected, adjusting Value Control is basically the same as adjusting an input data item. If an output is digital, the control field will display a **Turn On** or **Turn Off** button. If it is analog, the control field will display a scroll bar. When the Scroll Bar button is moved, to change the state of the data item, the new value will be displayed in the bottom right cell of the Debug window.

Status type outputs also allow the user to apply hypothetical conditions to output data items (Open, Short GND, Short BATT, OverCurrent, Grounded, STB or Open, Under/Over Volt, Over Temp and STG or Open).

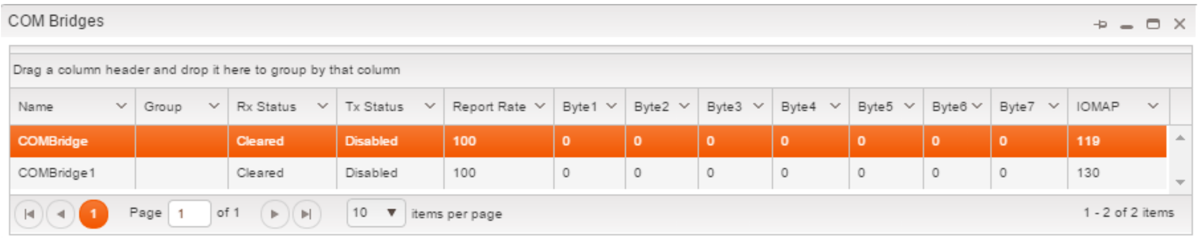
Note: Make all initial Output Value Control adjustments before entering the Debug mode.

Related:

Input Types 210

8.5.11 COM Bridges Window

The COM Bridges window shows the current values for the comm bridge data items that was configured in Orchestra® Composer™, similar to the input, output, etc. windows.



COM Bridges

Drag a column header and drop it here to group by that column

Name	Group	Rx Status	Tx Status	Report Rate	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	IOMAP
COMBridge		Cleared	Disabled	100	0	0	0	0	0	0	0	119
COMBridge1		Cleared	Disabled	100	0	0	0	0	0	0	0	130

Page 1 of 1 10 items per page 1 - 2 of 2 items

COM Bridges

Name

This is the name assigned to the COM Bridge data item in Orchestra® Composer™.

Group

This is the group that the data item was assigned to in Orchestra®.

Rx Status

This is the received status. It shows whether or not a valid message was received for the COM bus.

Tx Status

This is the data item's TX status property set in Orchestra® Composer™ The options are Pass Through, On Report Rate, and Disabled.

Set the TX Status in Orchestra Composer, by adjusting the TX Status property of the COM Bridge data item. If set to At Rate, it will transmit at report rate. If set to Pass Through, it will transmit regardless of the rate. If set to Disabled, the data will not transmit.

Report Rate

This shows the rate at which the data is being transmitted.

Byte 1

The current value of byte 1 that will be transmitted. The RX status needs to be received, the TX status needs to be enabled, and the Report Rate timer needs to be set.

Byte 2

The current value of byte 2 that will be transmitted. The RX status needs to be received, the TX status needs to be enabled, and the Report Rate timer needs to be set.

Byte 3

The current value of byte 3 that will be transmitted. The RX status needs to be received, the TX status needs to be enabled, and the Report Rate timer needs to be set.

Byte 4

The current value of byte 4 that will be transmitted. The RX status needs to be received, the TX status needs to be enabled, and the Report Rate timer needs to be set.

Byte 5

The current value of byte 5 that will be transmitted. The RX status needs to be received, the TX status needs to be enabled, and the Report Rate timer needs to be set.

Byte 6

The current value of byte 6 that will be transmitted. The RX status needs to be received, the TX status needs to be enabled, and the Report Rate timer needs to be set.

Byte 7

The current value of byte 7 that will be transmitted. The RX status needs to be received, the TX status needs to be enabled, and the Report Rate timer needs to be set.

IOMAP

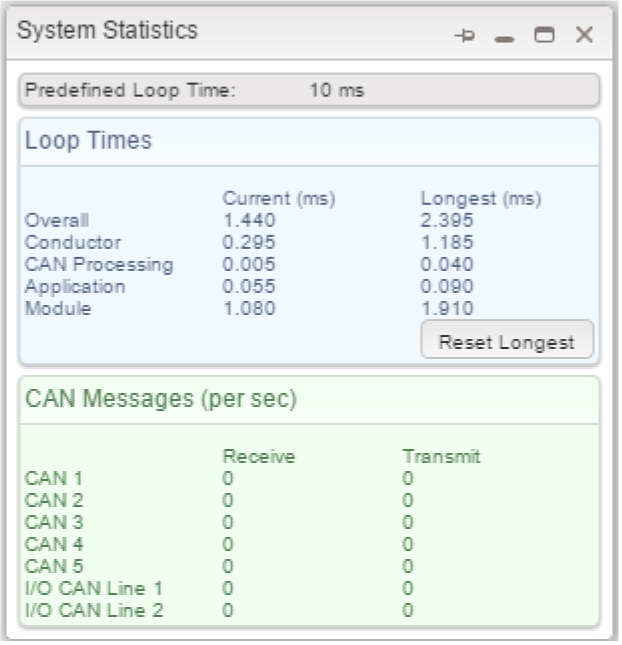
Gives the IOMap address of the data item from the Constants.h file that is used for debugging.

8.5.12 System Statistics

The System Statistics window shows the modules defined loop time that is set in Orchestra®. Each other part is a recording from the module to display the statistics.

Loop time is the amount of time it takes the firmware to run through everything it has to do once. If the module takes longer than the predefined loop time, the module gets cut off and has to start again. Some functionality will be impaired if the loop time is exceeded.

The CAN Messages per sec is showing the incoming and outgoing message averages over a period of time.



Predefined Loop Time: 10 ms

Loop Times

	Current (ms)	Longest (ms)
Overall	1.440	2.395
Conductor	0.295	1.185
CAN Processing	0.005	0.040
Application	0.055	0.090
Module	1.080	1.910

CAN Messages (per sec)

	Receive	Transmit
CAN 1	0	0
CAN 2	0	0
CAN 3	0	0
CAN 4	0	0
CAN 5	0	0
I/O CAN Line 1	0	0
I/O CAN Line 2	0	0

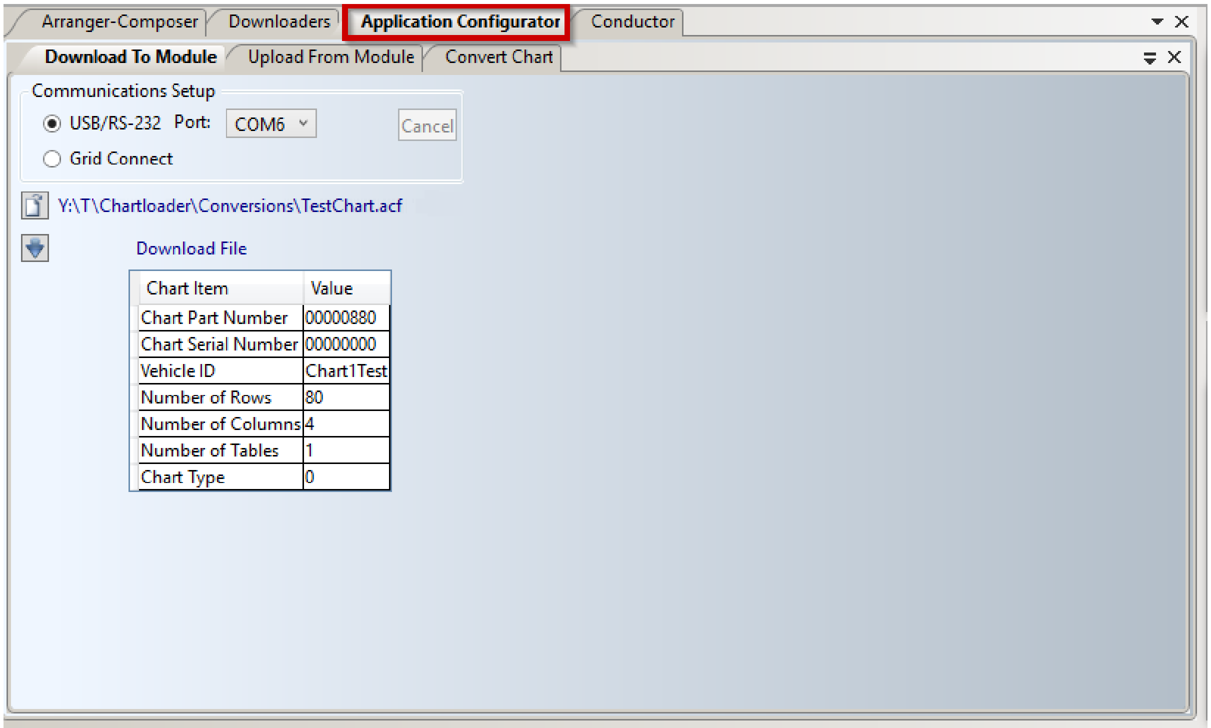
System Statistics Window

Application Configurator

9 Application Configurator

Application Configurator allows the user to download and upload configurable charts to and from master modules. It can also do a conversion to a chart to provide a kind of revision control that can prevent the given chart from being edited.

Note: This is only for master modules and is configured inside of the Presto™ code, not through Orchestra®.



Communications Setup

● USB/RS-232 Port: COM6

○ Grid Connect

Y:\T\Chartloader\Conversions\TestChart.acf

Download File

Chart Item	Value
Chart Part Number	00000880
Chart Serial Number	00000000
Vehicle ID	Chart1Test
Number of Rows	80
Number of Columns	4
Number of Tables	1
Chart Type	0

Application Configurator

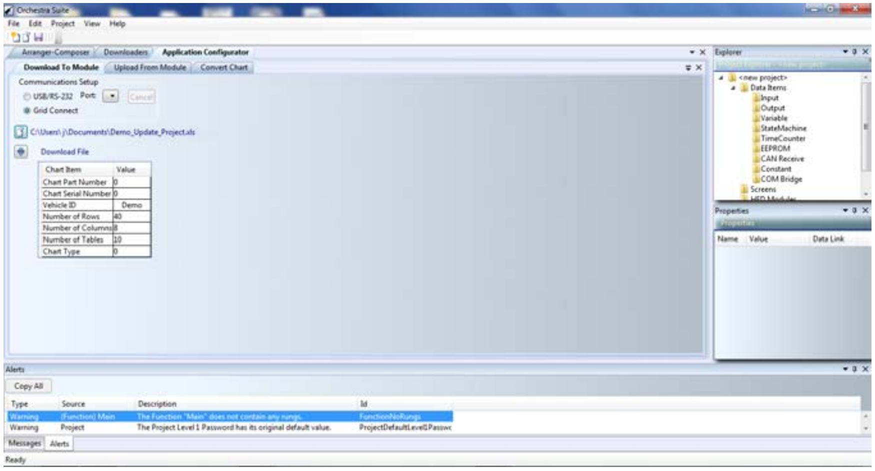
Related:

[Download To Module](217)
[Upload From Module](218)
[Convert Chart](219)
[Using Presto](334)

9.1 Download To Module

To download a chart to a module, follow the steps below.

1. Open Orchestra®.
2. Click the Application Configurator Tab.
3. Within the Application Configurator tab, click on the Download To Module tab.
4. Choose the connection method to be used, either through CAN using Grid Connect or through USB.
5. Click the small button below the communication set up to navigate to the chart that is to be downloaded.
6. Once the chart is selected and loaded within Orchestra® click the download file button.
7. After the download completes it is safe to disconnect the module.



Communications Setup

○ USB/RS-232 Port: Cancel

● Grid Connect

C:\Users\j\Documents\Demo_Update_Project.xls

Download File

Chart Item	Value
Chart Part Number	0
Chart Serial Number	0
Vehicle ID	Demo
Number of Rows	40
Number of Columns	8
Number of Tables	10
Chart Type	0

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc

Ready

Download To Module

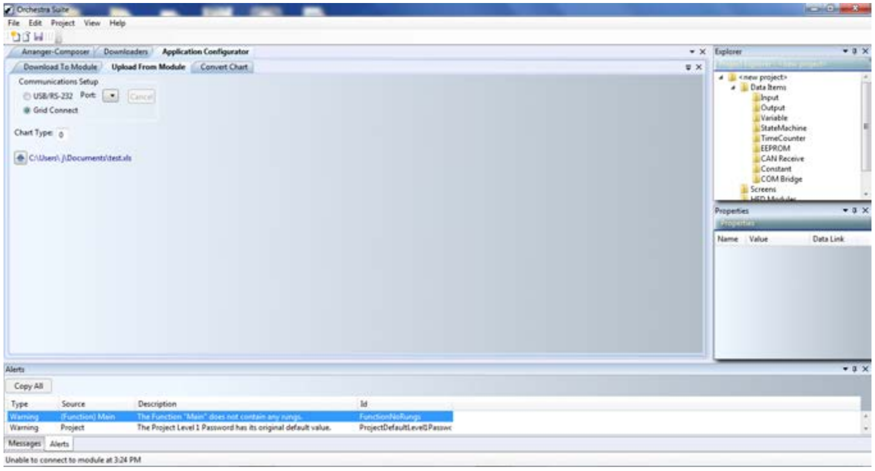
Related:

[Upload From Module] 218
[Application Configurator] 216
[Firmware Download] 225
[Application Download] 227
[Display Download] 229

9.2 Upload From Module

To upload a chart from a module, follow the steps below.

1. Open Orchestra®.
2. Click the Application Configurator Tab.
3. Within the Application Configurator tab, click on the Upload From Module tab.
4. Choose the connection method to be used, either through CAN using Grid Connect or through USB.
5. Choose the chart type that will be uploaded from the module.
6. Select a file for the chart to be uploaded into, usually an Excel file. After choosing, the Configurator will attempt to upload the chart.



○ USB/RS-232
● Grid Connect

Chart Type: 0

C:\Users\ .\Documents\test.xls

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc

Unable to connect to module at 3:24 PM

Upload From Module

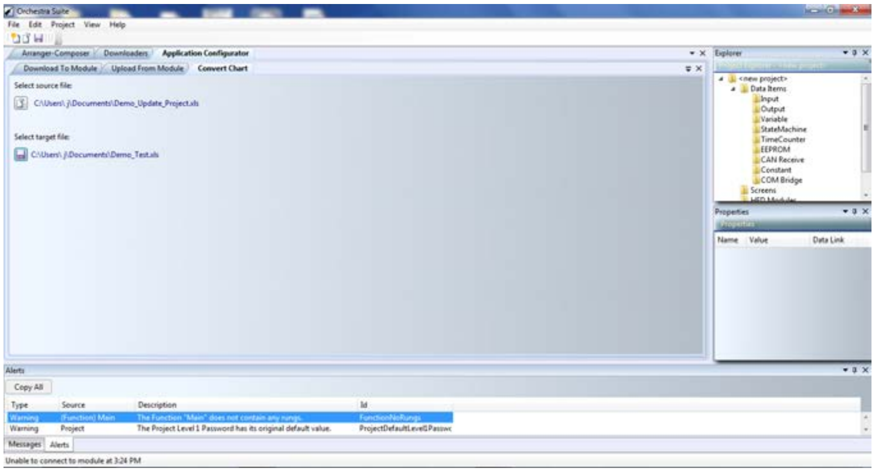
Related:

[Download To Module](217)
[Application Configurator](216)
[Firmware Download](225)
[Application Download](227)
[Display Download](229)

9.3 Convert Chart

Follow the steps below to convert a chart.

1. Open Orchestra®.
2. Click the Application Configurator Tab.
3. Within the Application Configurator tab, click on the Convert Chart tab.
4. Click the button to browse to the file that is to be converted.
5. Select the file for conversion. The Configurator prepares the file to be converted.
6. Once the file to be converted has been chosen and prepared, another button will appear. Click it and browse to the location where the newly created file will be placed, choose the format for the file to be converted to, and give the new file a name.



Select source file:
C:\Users\ j\Documents\Demo_Update_Project.xls

Select target file:
C:\Users\ j\Documents\Demo_Test.xls

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc

Unable to connect to module at 3:24 PM

Convert Chart

Related:

[Application Configurator](#) 216

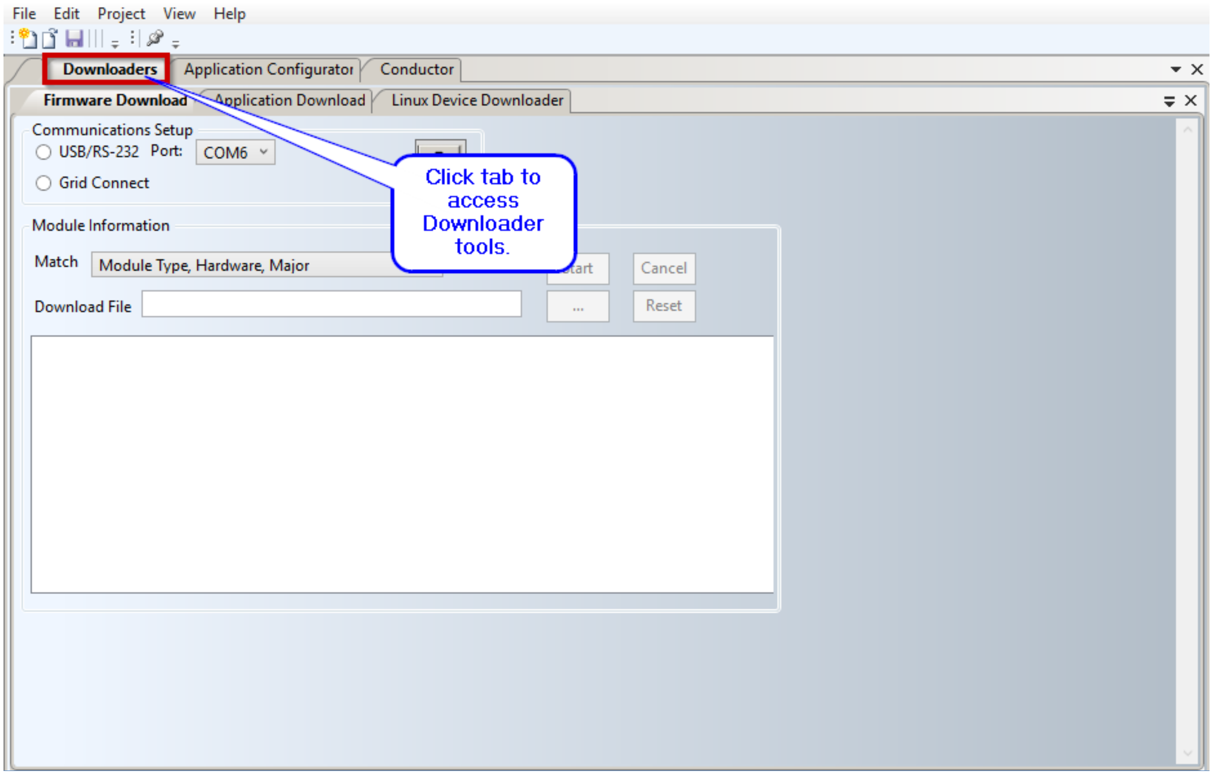
Orchestra® Downloader Guide

10 Orchestra® Downloader Guide

10.1 Introduction

There are three subtabs: Firmware Download 225, Application Download 227, and Linux Device Downloader. They each download specific software to a specific device. Before any downloading can be done, the hardware and tools must be connected and configured properly.

To access the Downloader tool, left click on the top middle tab that says "Download Project" within the main Orchestra® screen.



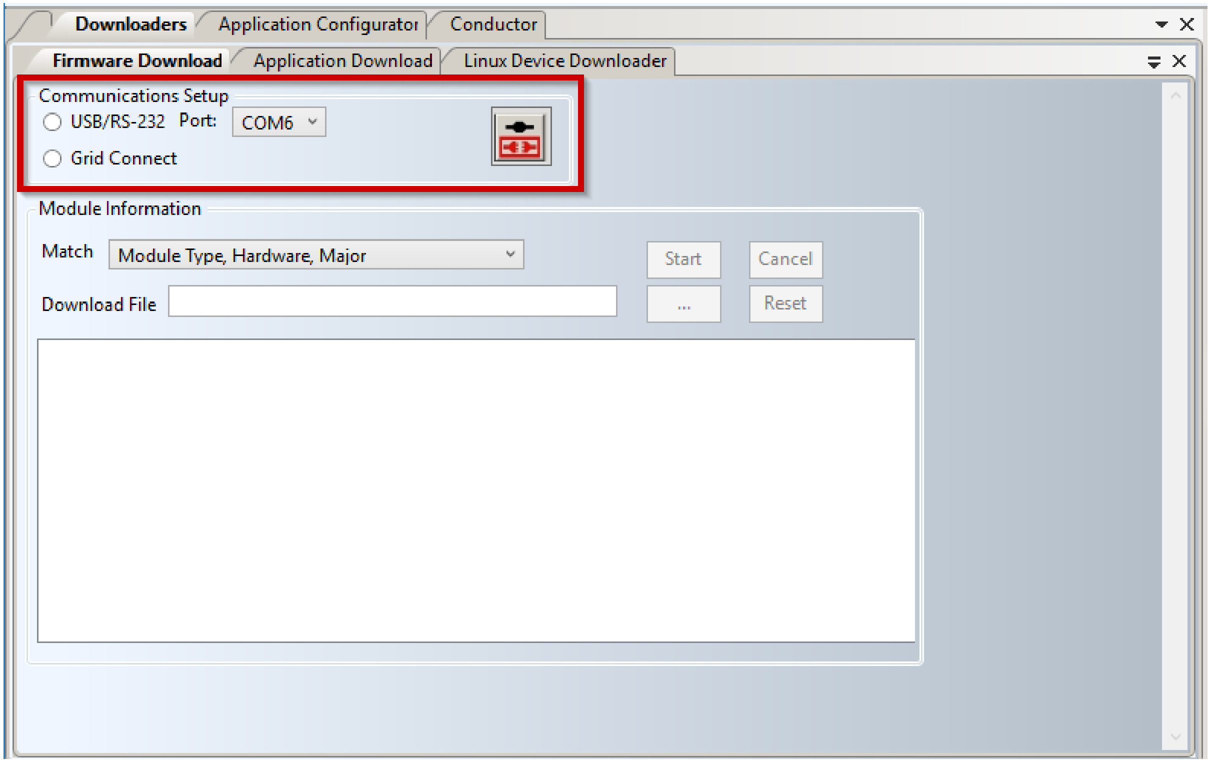
Downloader

Related:
How to Download to a Module 223

10.1.1 How to Download to a Module

There are two ways to download to a module:

1. Through a USB/RS-232 connection if the module has the capability.
2. Through the CAN using a USB to CAN adapter such as Grid Connect.



Communications Setup: USB/RS-232 or Grid Connect

Note: If the module is USB capable, Grid Connect can still be used instead, if the user wishes to download over CAN.

Materials

For the USB connection, a USB to USB cable is needed. After the cable is connected, the user must choose the COM port that the module is connected to via the drop-down menu.

To connect to the Grid Connect device, the CAN line needs to be broken out to a RS-232 connector with the standard CAN pin configuration to plug into the device.

Note: All of the downloaders can be used whether or not a project is loaded. If no project is loaded then the user will need to browse to and load the correct files.

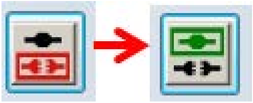
Related:
Firmware Download 225
Application Download 227

[Display Download](229)

10.1.1.1 Connect/Disconnect Button

Clicking this button connects to or disconnects from the Master Module in the OEM product. The button appears in two ways, depending upon whether there is a communications link between the product and the computer.

If the button appears as “Connected”, click it to disconnect. Conversely, if the button appears as “Disconnected”, you can click it to re-establish the connection.



Connect-Disconnect Button

Related:

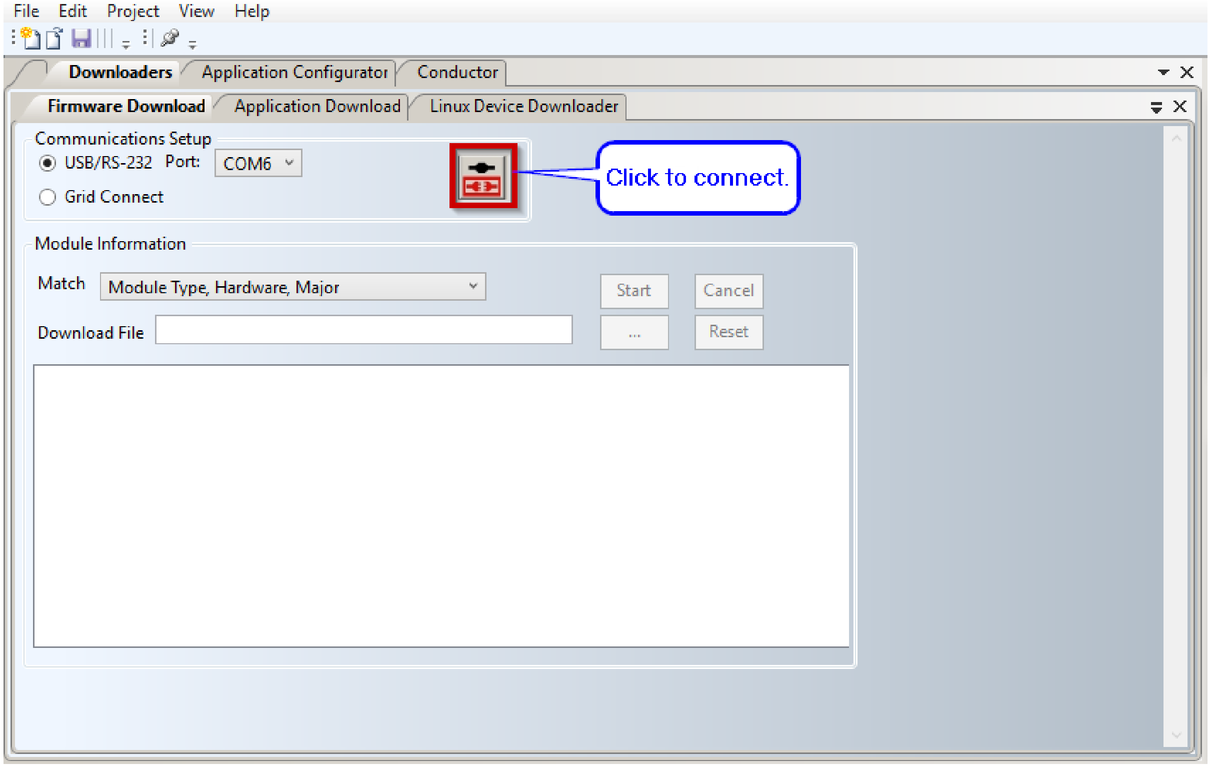
[Firmware Download](225)

10.2 Firmware Download

The Firmware Download is used to update the firmware of the master and client modules. When used with Presto™, this would also update the master module application.

1. Power up the module(s).
2. Plug in the desired download interface method.
3. Once powered and connected, click the "Download Project" tab within Orchestra®.
4. Click the "Firmware Download" tab.
5. Choose the communication method being used, either USB/RS-232 or Grid Connect.
6. Click the connect icon to the right.

Note: If using the USB/RS-232 option, the user needs to also select the port that is being used to communicate to the module(s).

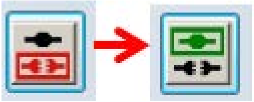


● USB/RS-232 Port: COM6

Match: Module Type, Hardware, Major

Click to connect.

Firmware Downloader



Disconnected to Connected Icon Change

Once connected, the icon will turn green and information within the “Module Information” field will be populated with all of the modules detected within the system.

Next to each found module is a check box, and clicking the check box marks that module for the firmware download. When marked, the “Download File” field will be auto-populated with the latest firmware file that is available with that version of Orchestra®. Click **Start** to begin the download, and Orchestra® will notify the user of completion.

Related:

How to Download to a Module 223
Application Download 227
Display Download 229

10.3 Application Download

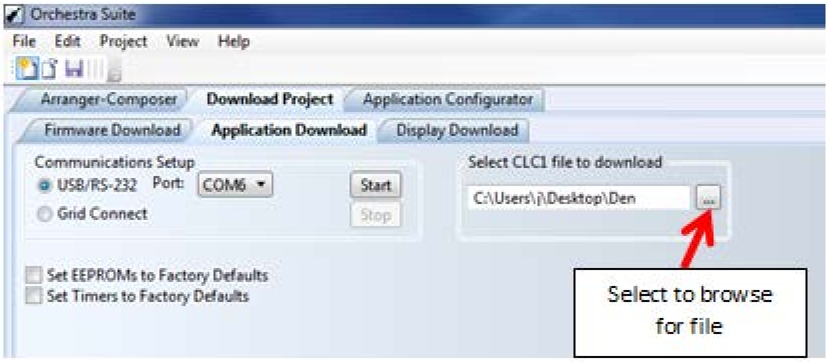
This is how to load the application if using Rungs.

1. Power up the module.

2. Plug in the interface desired to download over.

3. Click on the “Application Download” to access that downloader.

4. Depending on if a project loaded or not, there may or may not be a window to select the CLC1 file to download. If there is an open project, that window will not be there and Orchestra® will automatically select the open project’s CLC1 file for download. If there is no project open, the user will have to manually select the file to download.



● USB/RS-232 Port: COM6

C:\Users\j\Desktop\Den

Select to browse for file

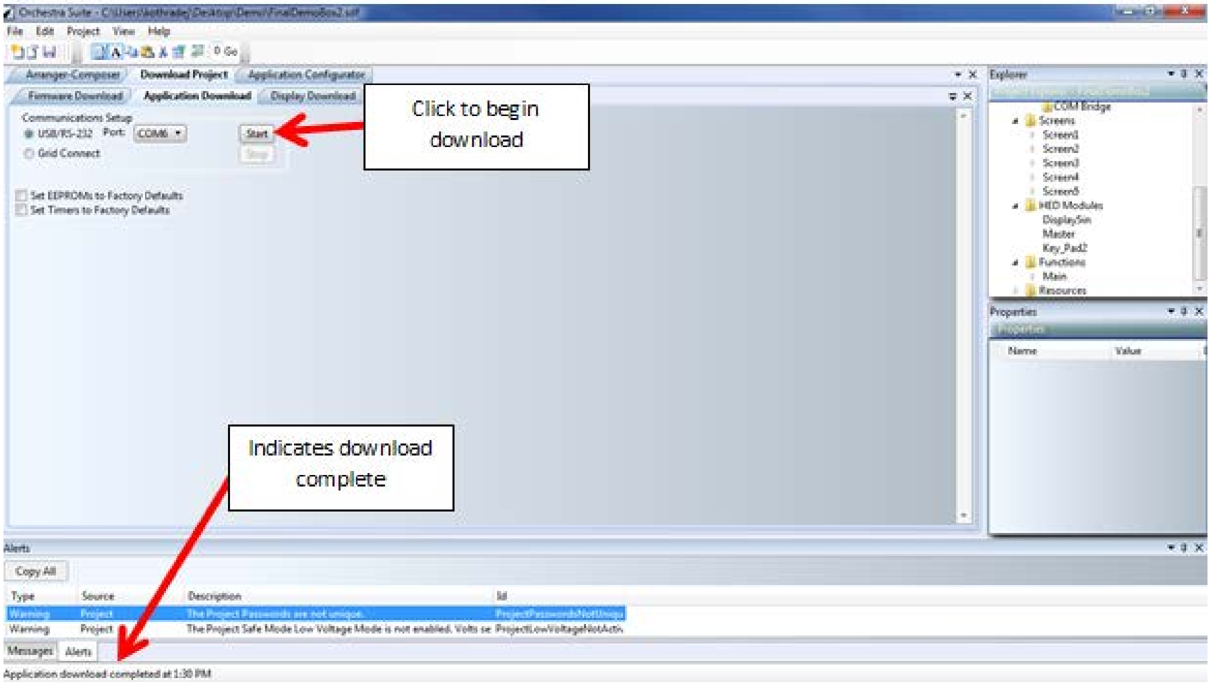
Application Downloader Manual File Selection

5. Once the desired CLC1 file is selected, choose the communication method and click the Start button.

6. Track the download progress via a progress bar in the lower right hand corner of the screen.

7. Orchestra® indicates to the user that the download is complete via text in the lower left corner of the screen.

There are two check boxes; “Set EEPROMS to Factory Defaults” and “Set Timers to Factory Defaults” that the user can check if they desire to reset the EEPROMs 101 and Timers to their factory default values.



Communications Setup
● USB/RS-232 Port: COM6
○ Grid Connect
Start
Stop
□ Set EEPROMs to Factory Defaults
□ Set Timers to Factory Defaults

Click to begin download

Indicates download complete

Type	Source	Description	Id
Warning	Project	The Project Passwords are not unique.	ProjectPasswordsNotUniqu
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActi\

Application download completed at 1:30 PM

Application Downloader

Related:

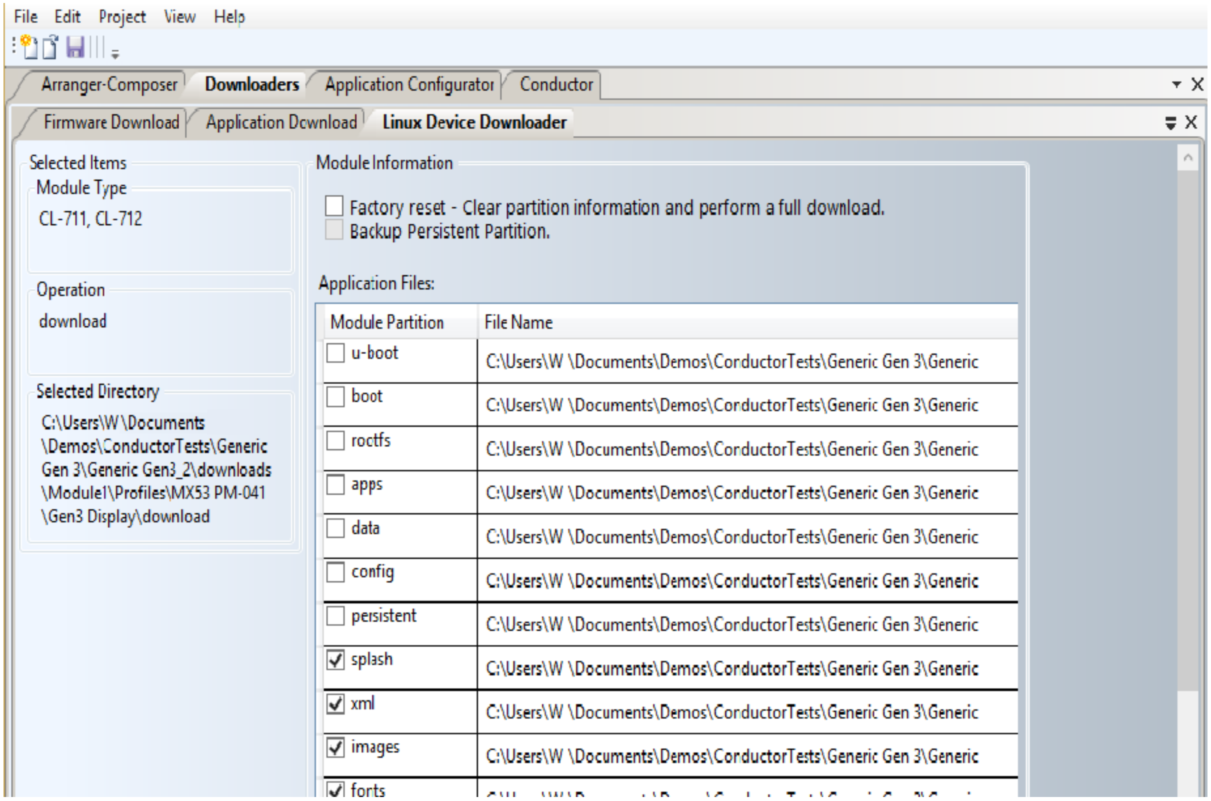
[How to Download to a Module](223) 223
[Firmware Download](225) 225
[Display Download](229) 229

10.4 Linux Device Downloader

The Linux Device Downloader programs the firmware and screen information into the display.

If the display is a master module, there is a built in USB CAN pass-through internal to the display that allows the display to be programmed with the application software and the display files using the same connection.

If the display is an I/O module, it cannot be downloaded through the master module and will require its own USB connection.



Selected Items

Module Type: CL-711, CL-712

Operation: download

Selected Directory: C:\Users\W \Documents \Demos\ConductorTests\Generic Gen 3\Generic Gen3_2\downloads \Module1\Profiles\MX53 PM-041 \Gen3 Display\download

Module Information

□ Factory reset - Clear partition information and perform a full download.
□ Backup Persistent Partition.

Application Files:

Module Partition	File Name
□ u-boot	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
□ boot	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
□ roctfs	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
□ apps	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
□ data	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
□ config	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
□ persistent	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
☑ splash	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
☑ xml	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
☑ images	C:\Users\W \Documents\Demos\CorductorTests\Generic Gen 3\Generic
☑ fonts	[illegible]

Linux Device Downloader

Related:

Using USB Cables 230
Display Information 231
How to Download to a Module 223
Firmware Download 225
Application Download 227

10.4.1 Using USB Cables

There is a specific order in which the USB cable is plugged in, in order to either program the display files or the application files.

Flash Application Software

If the display is a Master and the user desires to flash the application software, the USB cable must be plugged in after the display has already been powered up.

Flash Display Files

If the user wants to flash the display files, the USB must be plugged in prior to powering up the display regardless of being a Master or I/O Module.

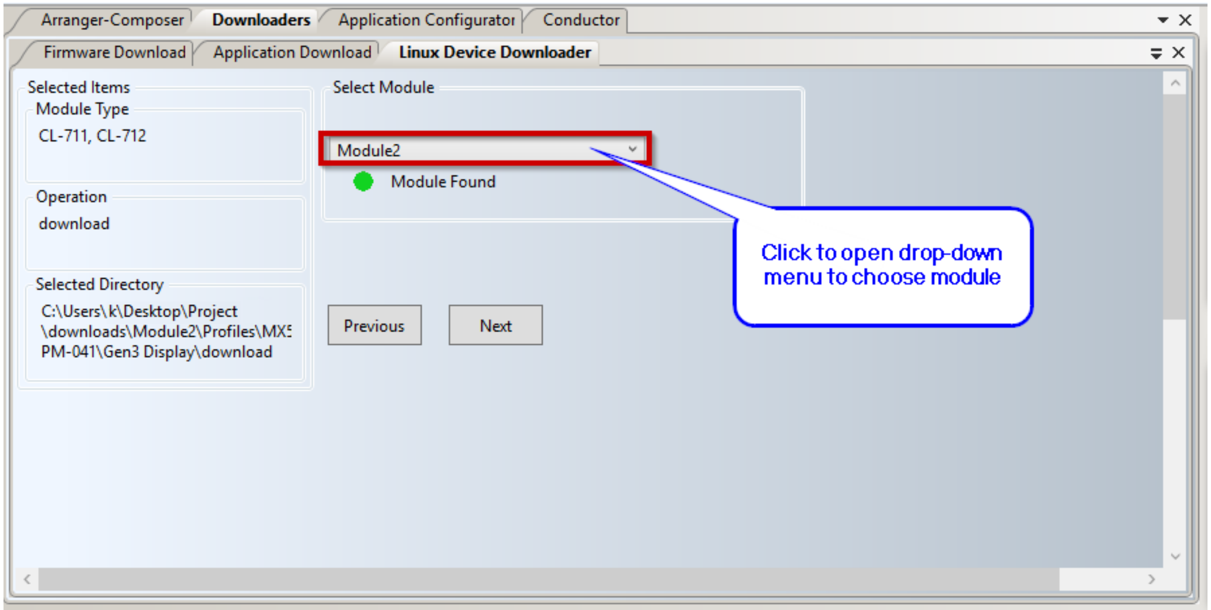
Related:

Display Download 229

10.4.2 Linux Device Information

Select Module

On the Select Module page, there is a drop-down menu with the option to choose which display to download to if there are multiple displays connected, otherwise the only option will be the single display.



Selected Items

Module Type
CL-711, CL-712

Operation
download

Selected Directory
C:\Users\k\Desktop\Project
\downloads\Module2\Profiles\MX5
PM-041\Gen3 Display\download

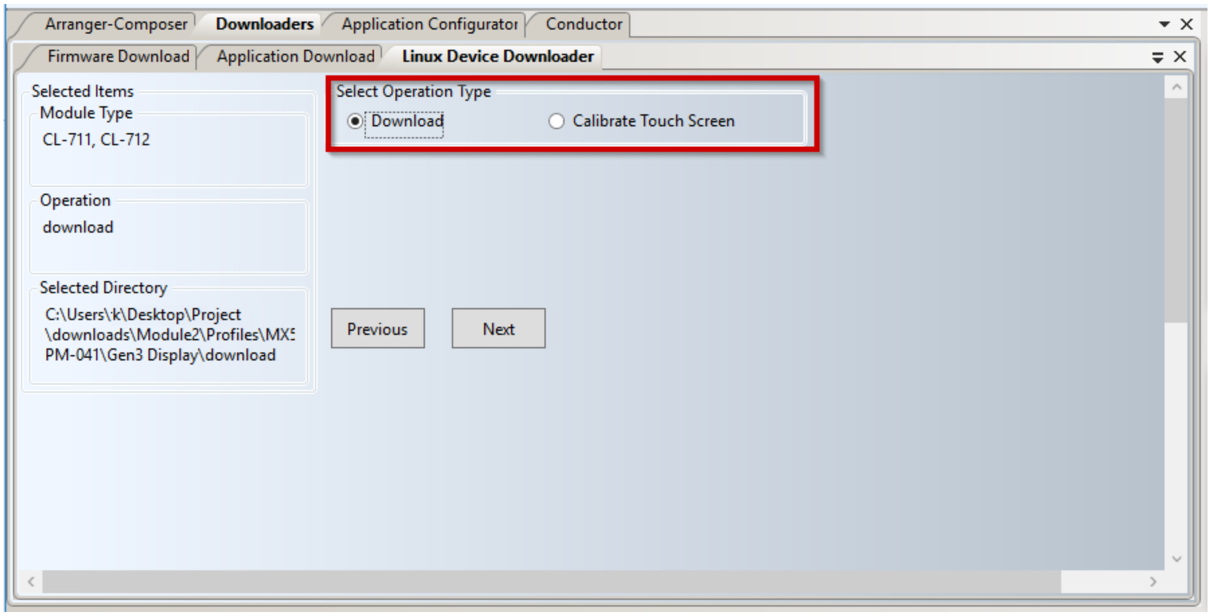
Select Module
Module2
● Module Found

Click to open drop-down menu to choose module

Select Module

Select Operation Type

Choose Download or Calibrate Touch Screen.



Selected Items

Module Type
CL-711, CL-712

Operation
download

Selected Directory
C:\Users\k\Desktop\Project
\downloads\Module2\Profiles\MX5
PM-041\Gen3 Display\download

Select Operation Type
● Download ○ Calibrate Touch Screen

Select Operation Type

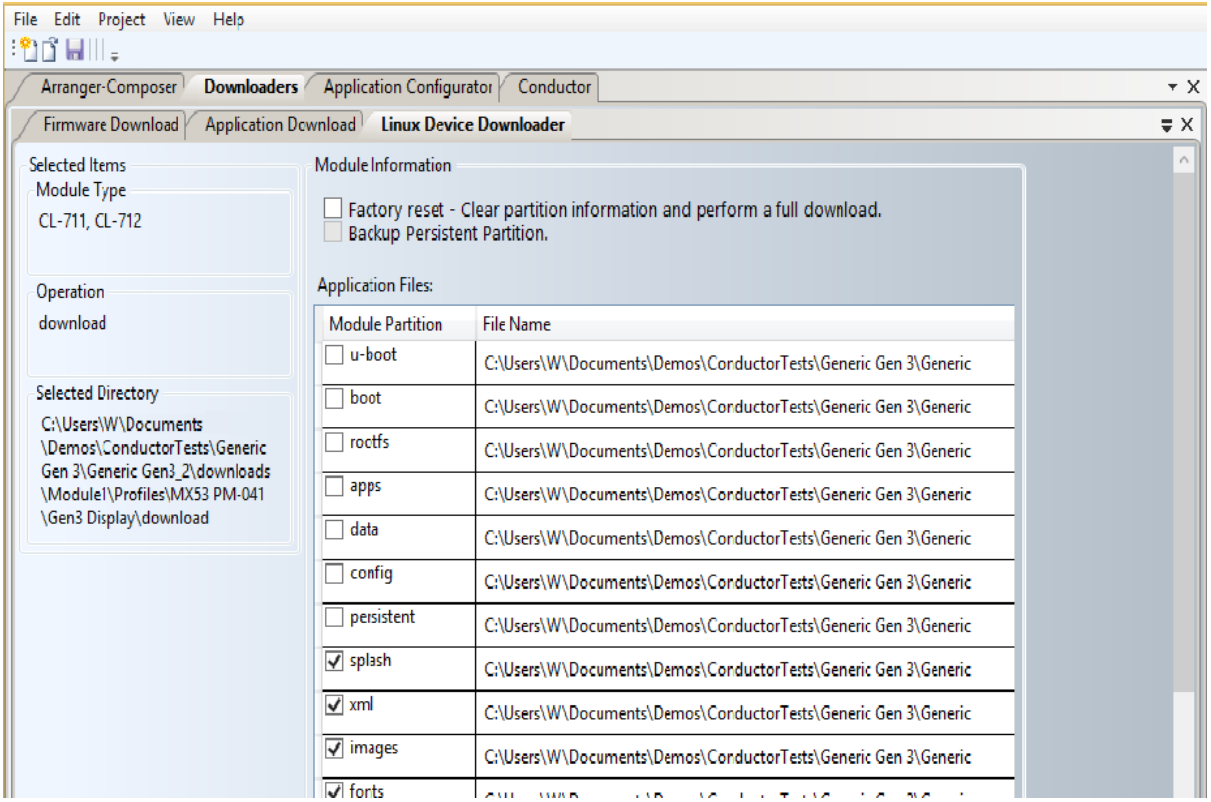
Module Information

The factory reset check box is used to clear previously downloaded files from the module.

The Backup Persistent Partition is used to back up and restore specific items in the persistent partition after a Factory reset.

The application files include a logo, xml, images, and fonts selection. Each of those is their own separate file and the paths to them should be automatically filled. The user can choose which files to load by clicking the check box next to each one. The xml file is the main file for the display, which is the equivalent of the clc1 file that a Master Module needs that holds the actual application code.

Note: the Orchestra® project must be compiled before the application files can be downloaded.



Selected Items

Module Type: CL-711, CL-712

Operation: download

Selected Directory: C:\Users\W\Documents \Demos\ConductorTests\Generic Gen 3\Generic Gen3_2\downloads \Module1\Profiles\MX53 PM-041 \Gen3 Display\download

Module Information

□ Factory reset - Clear partition information and perform a full download.
□ Backup Persistent Partition.

Application Files:

Module Partition	File Name
□ u-boot	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
□ boot	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
□ roctfs	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
□ apps	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
□ data	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
□ config	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
□ persistent	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
☑ splash	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
☑ xml	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
☑ images	C:\Users\W\Documents\Demos\ConductorTests\Generic Gen 3\Generic
☑ fonts	

Module Information

Related:

Display Download 229

10.5 Downloader Wizard Packet

The Downloader Wizard Packet is a way to fully package and download only the files needed for that given project, which also provides a form of revision control. This feature creates the packet.

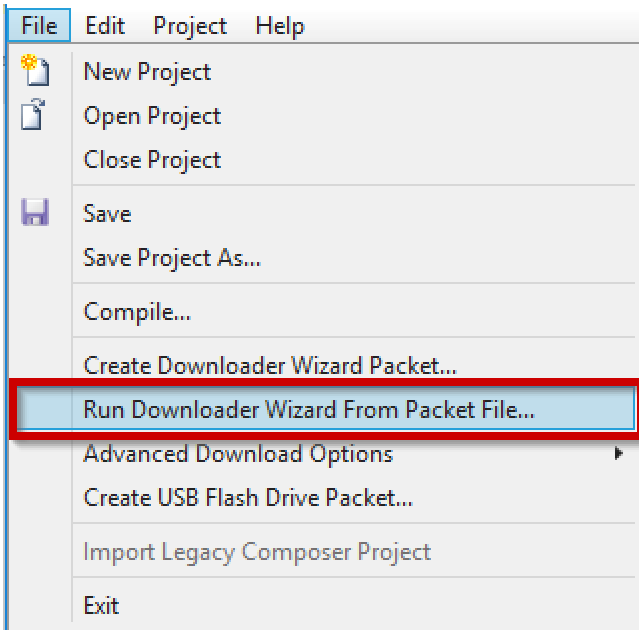
Orchestra® takes the user through the setup process. Use the Run Downloader Wizard from Packet File command to use Orchestra® to install the packet.

Related:
Downloading from a Packet 233

10.5.1 Downloading from a Packet

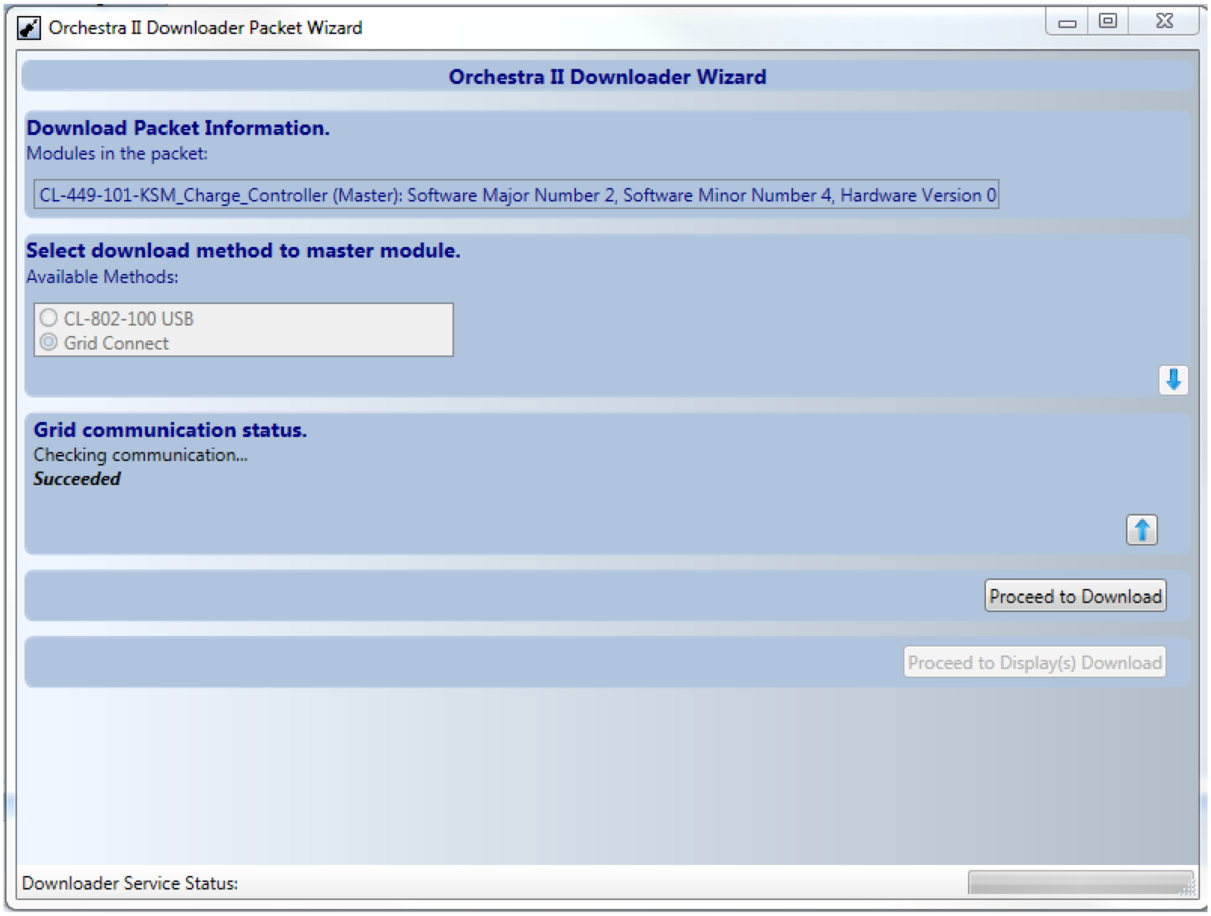
To download from a Wizard packet, follow the steps outlined below.

1. Open a new instance of Orchestra®.
2. Select the file drop-down menu and click on “Run Downloader Wizard from Packet File…”.


File
Edit
Project
Help
New Project
Open Project
Close Project
Save
Save Project As...
Compile...
Create Downloader Wizard Packet...
Run Downloader Wizard From Packet File...
Advanced Download Options
Create USB Flash Drive Packet...
Import Legacy Composer Project
Exit

Begin Download From Packet

3. Navigate to the .dwp file within the computer and select it.
4. The Downloader Packet Wizard will pop up, showing the module information that the packet contains. Choose the download method from the available options.
5. Click the blue arrow once a selection has been made, and the wizard will check for the communication.



Orchestra II Downloader Packet Wizard

Orchestra II Downloader Wizard

Download Packet Information.
Modules in the packet:

CL-449-101-KSM_Charge_Controller (Master): Software Major Number 2, Software Minor Number 4, Hardware Version 0

Select download method to master module.
Available Methods:

○ CL-802-100 USB
● Grid Connect

Grid communication status.
Checking communication...
Succeeded

Downloader Service Status:

Downloader Packet Wizard

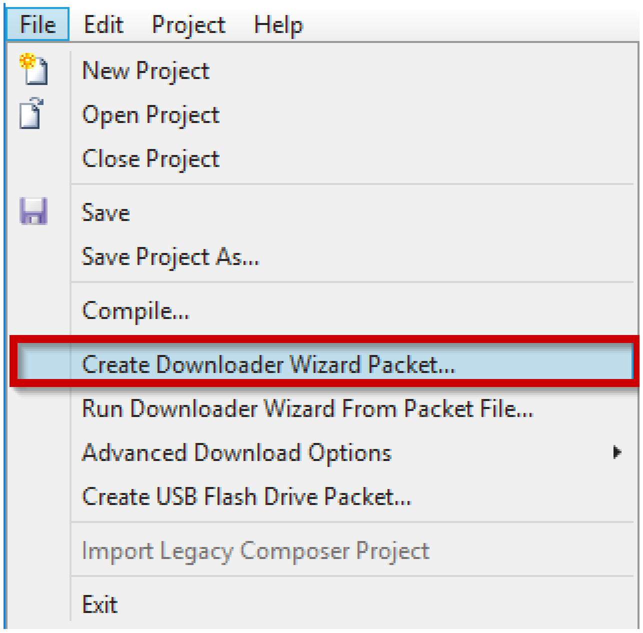
6. Once connected, click Proceed to Download and the download process will begin. The progress can be viewed with the status bar in the lower right hand corner.
7. Once the application is loaded, if there are also files to download to a display, the user will be prompted to do so.

Related:
Downloader Wizard Packet 233
Creating a Packet 235

10.5.1.1 Creating a Packet

To create a Downloader Wizard Packet, follow the steps below.

1. Open the project within Orchestra®.
2. Go to the file drop-down and select “Create Downloader Wizard Packet…”.


File
Edit
Project
Help
New Project
Open Project
Close Project
Save
Save Project As...
Compile...
Create Downloader Wizard Packet...
Run Downloader Wizard From Packet File...
Advanced Download Options
Create USB Flash Drive Packet...
Import Legacy Composer Project
Exit

Begin Download Packet Creation

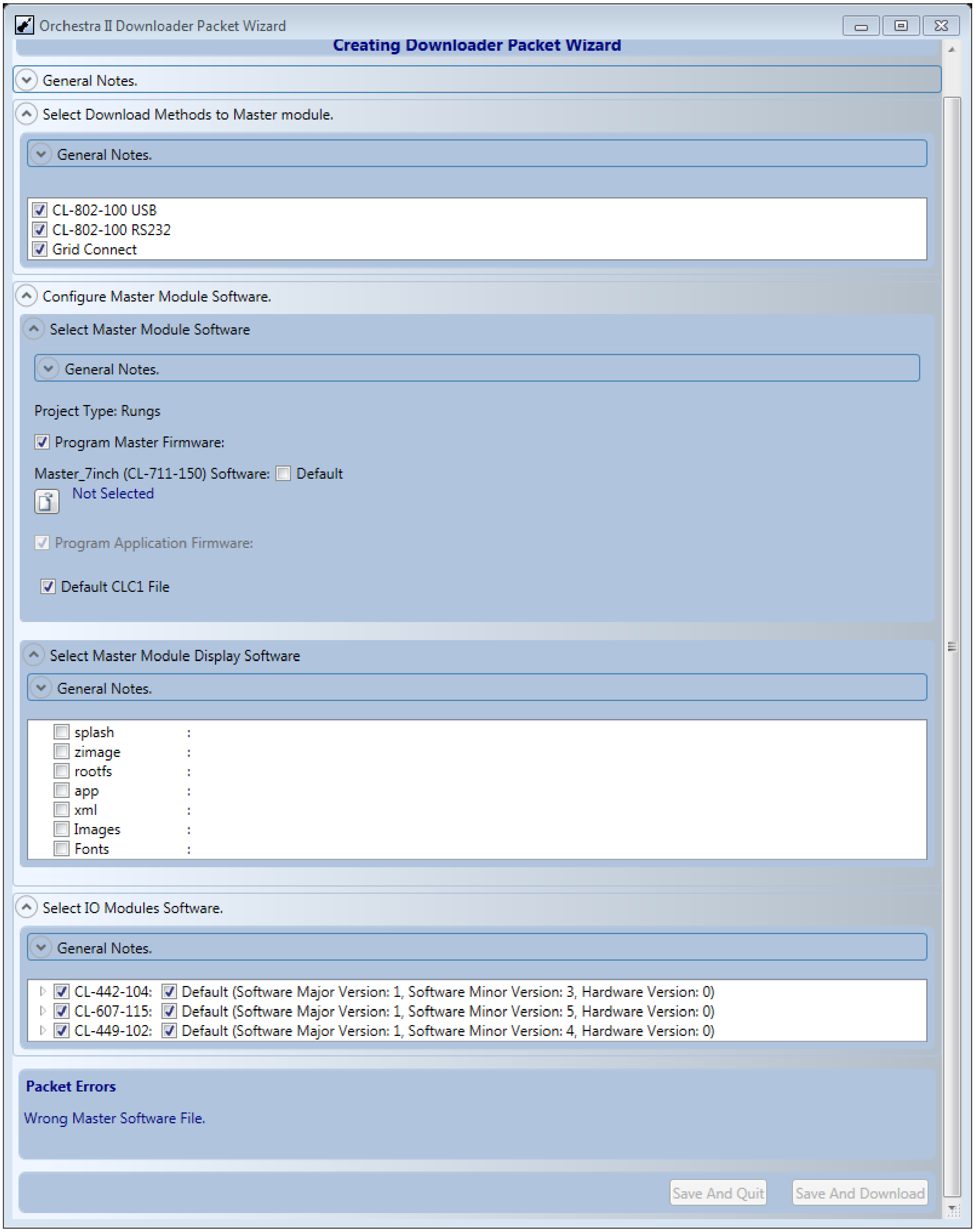
3. Select which download methods will be available to choose from when downloading.
4. Select the firmware to load into the master module, either the default or a specific file specified by the user. If this is selected when the download wizard is run it will update the firmware of the master module if necessary.

Note: if you do not select the firmware to be downloaded, there is a chance that the master module will not support the .clc1 format of the application. It is suggested that you use the default software, as this is the latest known firmware for the module and will be compatible with the .clc1 file generated with this version of Orchestra®.

5. If Rungs, select the .clc1 file to download to the module. The default .clc1 file is the one generated from the open Orchestra® project.
6. If a display is part of the project, select all of the files related to the display to be included with the download packet.
7. If desired, select I/O Module software to have the I/O Modules firmware updated. The file to be loaded to the modules can be specifically chosen if the default is unchecked and a new file is selected.

8. Click Save and Quit to create the .dwp file and close the Downloader Packet Wizard, or click Save and Download to save the .dwp file and immediately download the packet.
9. If there are any required fields that are empty or improperly filled, there will be an error box giving a brief description of the issue.

Note: When creating a .dwp for a Presto™ project Step 4 is where the file to be loaded is selected instead of the firmware file.



Orchestra II Downloader Packet Wizard
Creating Downloader Packet Wizard
General Notes.
Select Download Methods to Master module.
General Notes.

☑ CL-802-100 USB
☑ CL-802-100 RS232
☑ Grid Connect

Configure Master Module Software.
Select Master Module Software
General Notes.

Project Type: Rungs

☑ Program Master Firmware:

Master_7inch (CL-711-150) Software: □ Default

Not Selected

☑ Program Application Firmware:

☑ Default CLC1 File

Select Master Module Display Software
General Notes.

□ splash :
□ zimage :
□ rootfs :
□ app :
□ xml :
□ Images :
□ Fonts :

Select IO Modules Software.
General Notes.

☑ CL-442-104: ☑ Default (Software Major Version: 1, Software Minor Version: 3, Hardware Version: 0)
☑ CL-607-115: ☑ Default (Software Major Version: 1, Software Minor Version: 5, Hardware Version: 0)
☑ CL-449-102: ☑ Default (Software Major Version: 1, Software Minor Version: 4, Hardware Version: 0)

Packet Errors

Wrong Master Software File.

Downloader Packet Wizard Packet Creation

Related:

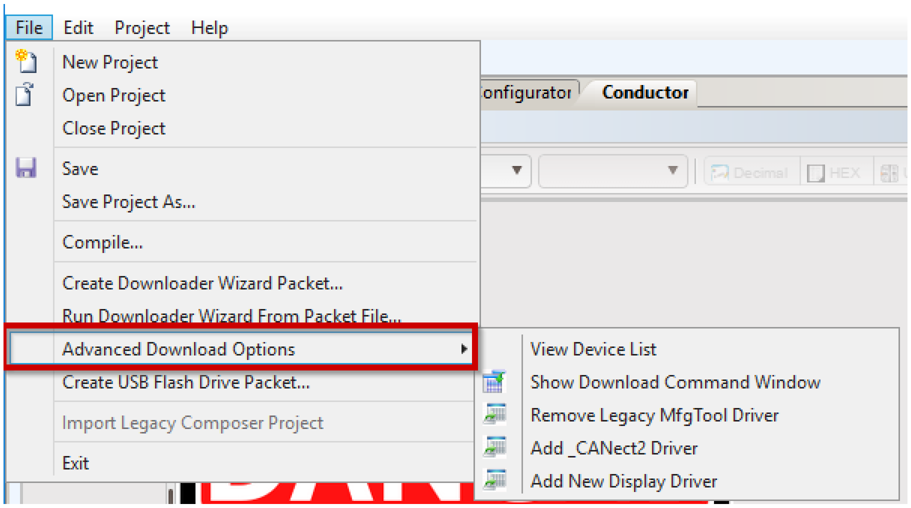
[Downloader Wizard Packet 233]

[Downloading from a Packet 233]

10.6 Advanced Download Options

The Advanced Download Options turn on the additional Linux Display Downloader 229 features. Use this menu to access View Device List, Show Download Command Window, Remove Legacy Mfg Tool Driver, Add_CANect2 Driver, and Add New Display Driver.

Note: The advanced download options for Orchestra® are implemented as part of the change to the new Downloader implementation for displays. Their functions match what is implemented in the Linux Device Downloader advanced menu.


File
Edit
Project
Help
New Project
Open Project
Close Project
Save
Save Project As...
Compile...
Create Downloader Wizard Packet...
Run Downloader Wizard From Packet File...
Advanced Download Options
Create USB Flash Drive Packet...
Import Legacy Composer Project
Exit
onfigurator
Conductor
Decimal
HEX
View Device List
Show Download Command Window
Remove Legacy MfgTool Driver
Add _CANect2 Driver
Add New Display Driver

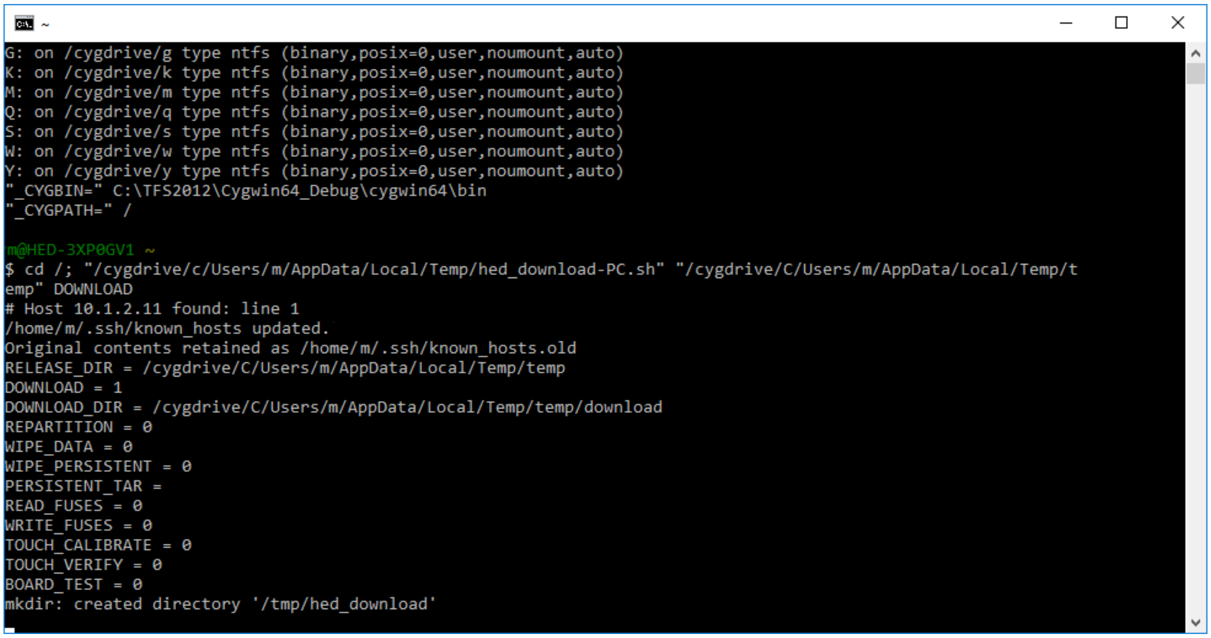
Advance Download Options

View Device List

This gives the user the option to show the list of download devices that are connected to the PC.

Show Download Command Window

This gives the user the option for the download status to be shown in a separate command window rather than in the status bar.



```
G: on /cygdrive/g type ntfs (binary,posix=0,user,noumount,auto)
K: on /cygdrive/k type ntfs (binary,posix=0,user,noumount,auto)
M: on /cygdrive/m type ntfs (binary,posix=0,user,noumount,auto)
Q: on /cygdrive/q type ntfs (binary,posix=0,user,noumount,auto)
S: on /cygdrive/s type ntfs (binary,posix=0,user,noumount,auto)
W: on /cygdrive/w type ntfs (binary,posix=0,user,noumount,auto)
Y: on /cygdrive/y type ntfs (binary,posix=0,user,noumount,auto)
"_CYGBIN=" C:\TFS2012\Cygwin64_Debug\cygwin64\bin
"_CYGPATH=" /

m@HED-3XP0GV1 ~
$ cd /; "/cygdrive/c/Users/m/AppData/Local/Temp/hed_download-PC.sh" "/cygdrive/C/Users/m/AppData/Local/Temp/t
emp" DOWNLOAD
# Host 10.1.2.11 found: line 1
/home/m/.ssh/known_hosts updated.
Original contents retained as /home/m/.ssh/known_hosts.old
RELEASE_DIR = /cygdrive/C/Users/m/AppData/Local/Temp/temp
DOWNLOAD = 1
DOWNLOAD_DIR = /cygdrive/C/Users/m/AppData/Local/Temp/temp/download
REPARTITION = 0
WIPE_DATA = 0
WIPE_PERSISTENT = 0
PERSISTENT_TAR =
READ_FUSES = 0
WRITE_FUSES = 0
TOUCH_CALIBRATE = 0
TOUCH_VERIFY = 0
BOARD_TEST = 0
mkdir: created directory '/tmp/hed_download'
```

Download Command Window

Remove Legacy Mfg Tool Driver

This removes the old driver used with the old download method: the Freescale driver used as part of the manufacturing tool.

Add CANect2 Driver

This adds the SE Blank 6UL driver.

Add New Display Driver

This adds the SE Blank Rita driver if it hasn't been added already.

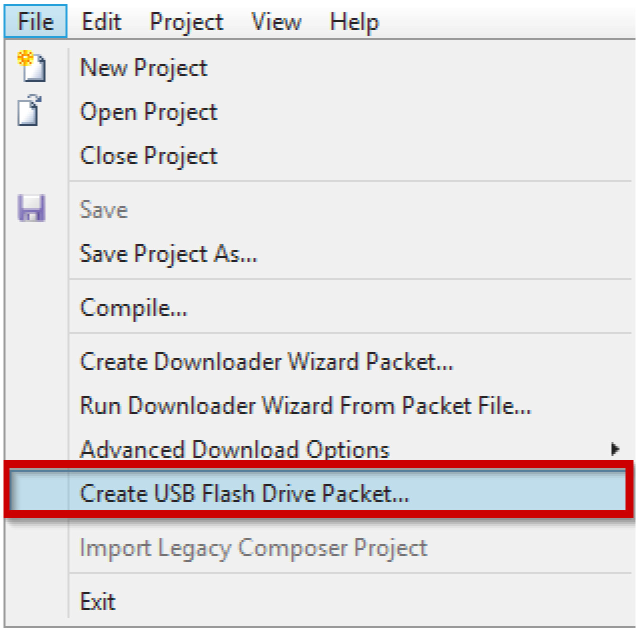
Related:
How to Download to a Module 223

10.7 Create USB Flash Drive Packet

The Create USB Flash Drive Packet command is used to generate an update package for the Gen III display (pre-SPU). This is used when a user has finished making updates to the project and wants to program a display.

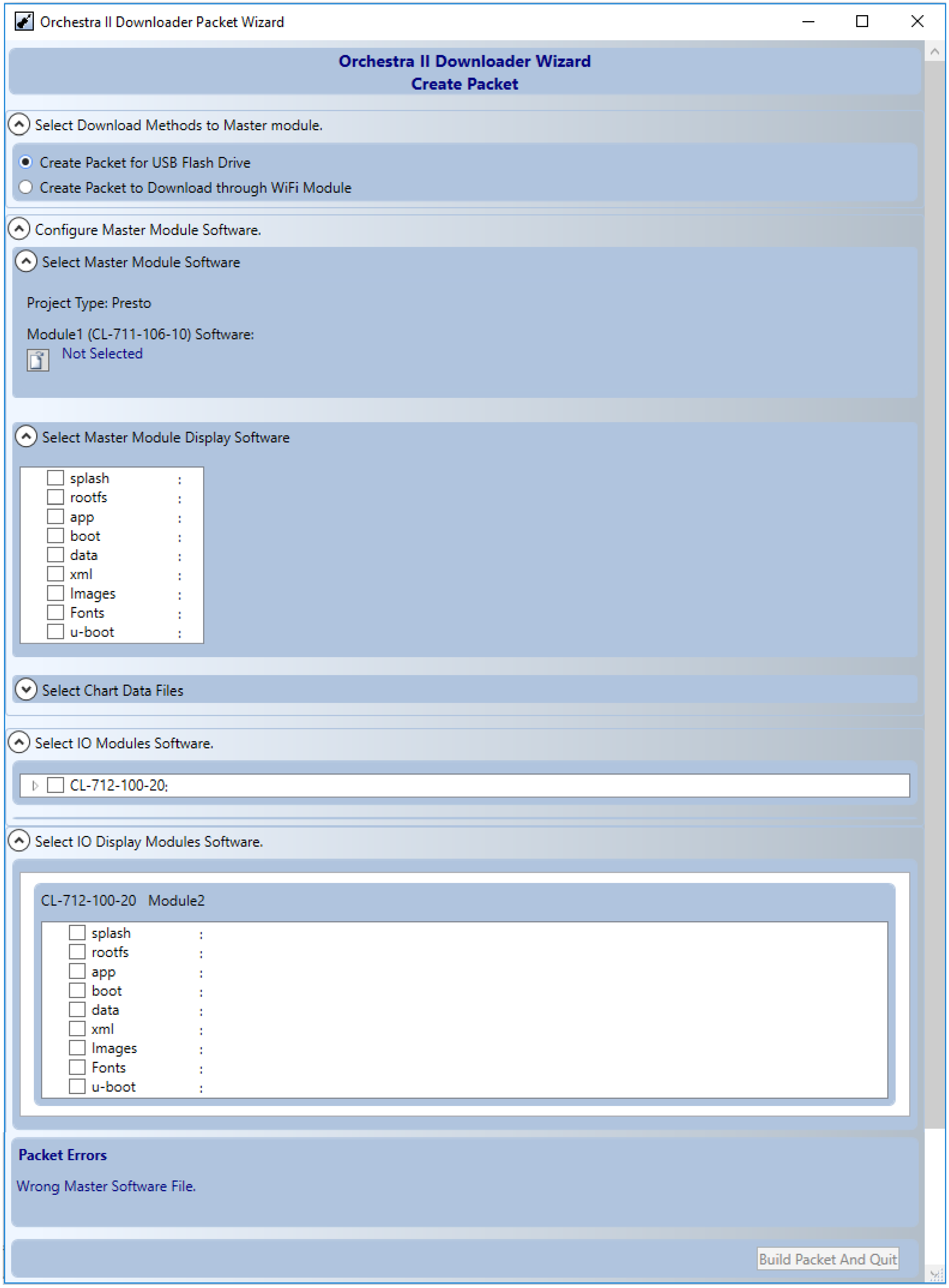
The project compile option 35 must be set to Presto™ to activate this feature.

This feature is activated once a Gen III display project is configured, saved, and compiled.


File
Edit
Project
View
Help
New Project
Open Project
Close Project
Save
Save Project As...
Compile...
Create Downloader Wizard Packet...
Run Downloader Wizard From Packet File...
Advanced Download Options
Create USB Flash Drive Packet...
Import Legacy Composer Project
Exit

File: Create USB Flash Drive Packet

Clicking the **Create USB Flash Drive Packet** command opens the **Orchestra II Downloader Wizard: Create Packet** screen.



Orchestra II Downloader Packet Wizard

Orchestra II Downloader Wizard
Create Packet

Select Download Methods to Master module.

● Create Packet for USB Flash Drive
○ Create Packet to Download through WiFi Module

Configure Master Module Software.

Select Master Module Software

Project Type: Presto

Module1 (CL-711-106-10) Software:
Not Selected

Select Master Module Display Software

□ splash :
□ rootfs :
□ app :
□ boot :
□ data :
□ xml :
□ Images :
□ Fonts :
□ u-boot :

Select Chart Data Files

Select IO Modules Software.

□ CL-712-100-20;

Select IO Display Modules Software.

CL-712-100-20 Module2

□ splash :
□ rootfs :
□ app :
□ boot :
□ data :
□ xml :
□ Images :
□ Fonts :
□ u-boot :

Packet Errors

Wrong Master Software File.

Build Packet And Quit

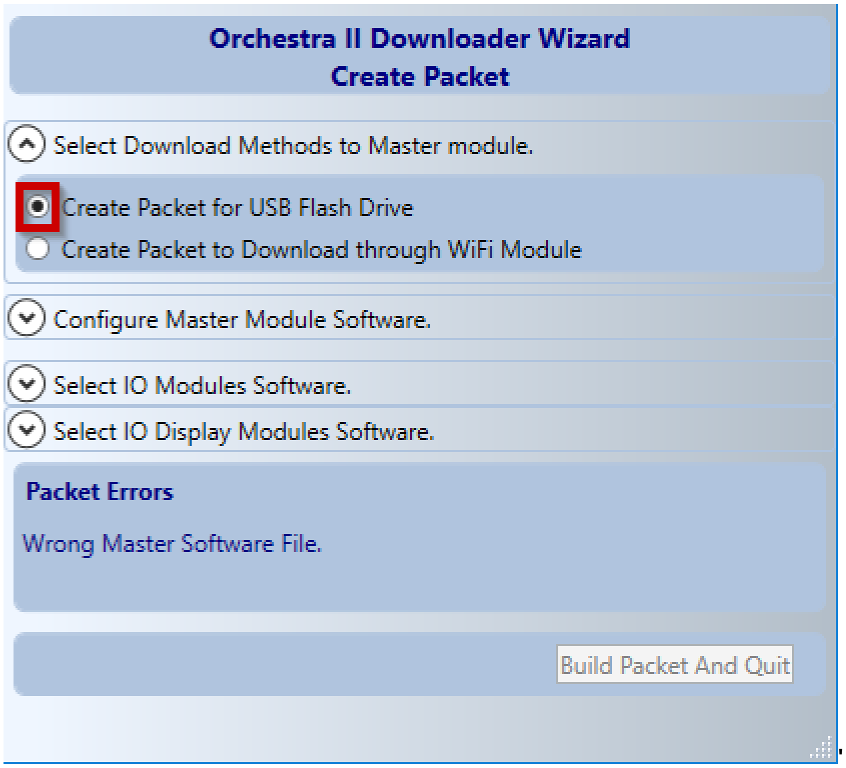
Create USB Flash Drive Packet

Related:

Using the Create Packet Wizard 243

10.7.1 Using the Create Packet Wizard

1. Compile a project. All the errors must be resolved before compiling.
2. Go to Filedrop-down Create USB Flash Drive Packet.
3. Go to the Orchestra II Downloader Wizard: Create Packet screen.
4. Go to the Select Download Methods to Master Module.



Orchestra II Downloader Wizard
Create Packet

Select Download Methods to Master module.

● Create Packet for USB Flash Drive
○ Create Packet to Download through WiFi Module

Configure Master Module Software.

Select IO Modules Software.

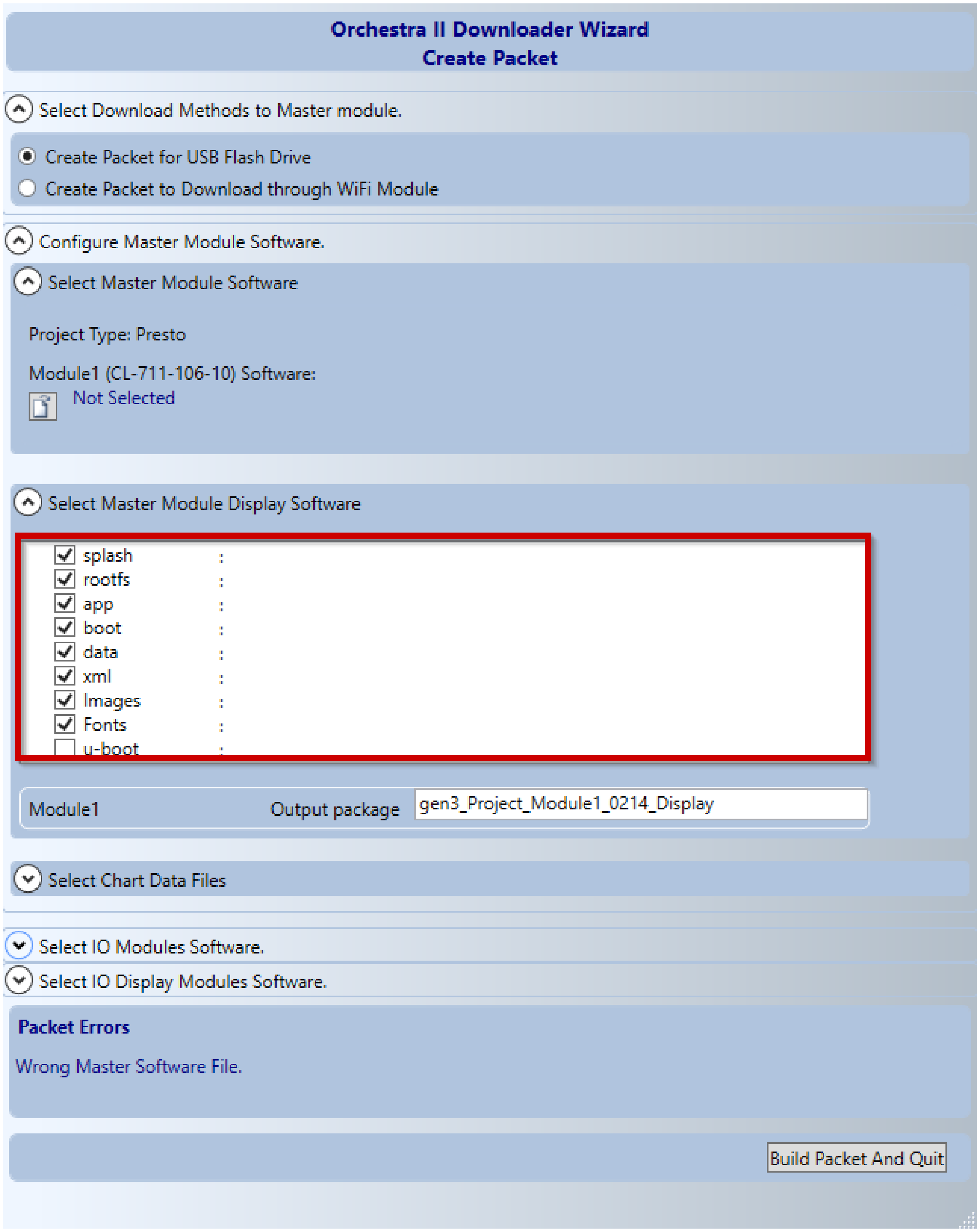
Select IO Display Modules Software.

Packet Errors

Wrong Master Software File.

Select Download Methods to Master Module

5. Click the Create Packet for USB Flash Drive radio button.
6. Go to the Configure Master Module Software drop-down Select Master Module Software pane.
7. Under Module Software, select module software.
8. Go to the Master Module Display Software pane.
9. Click the check boxes next to the files to include them in the download.



Orchestra II Downloader Wizard
Create Packet

Select Download Methods to Master module.

● Create Packet for USB Flash Drive
○ Create Packet to Download through WiFi Module

Configure Master Module Software.

Select Master Module Software

Project Type: Presto

Module1 (CL-711-106-10) Software:
Not Selected

Select Master Module Display Software

☑ splash :
☑ rootfs :
☑ app :
☑ boot :
☑ data :
☑ xml :
☑ Images :
☑ Fonts :
□ u-boot :

Module1 Output package gen3_Project_Module1_0214_Display

Select Chart Data Files

Select IO Modules Software.

Select IO Display Modules Software.

Packet Errors

Wrong Master Software File.

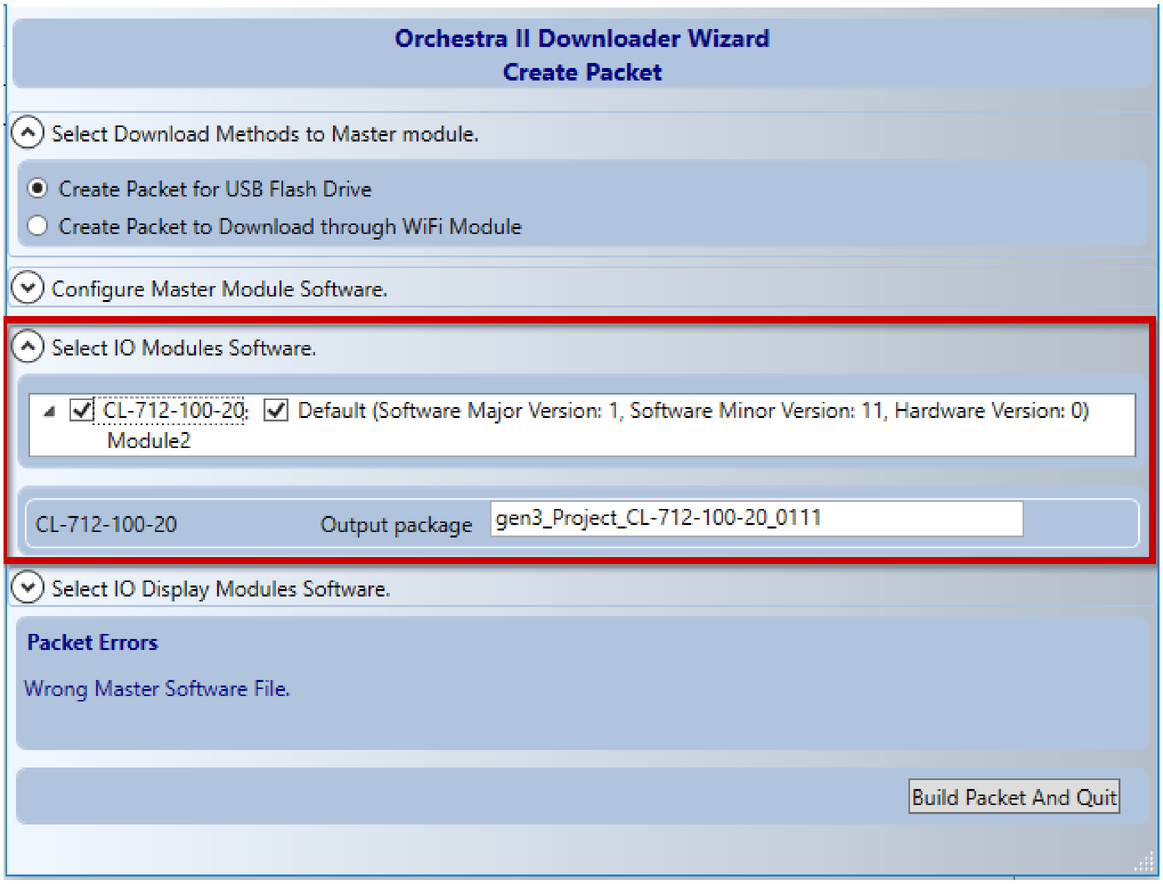
Select Master Module Display Software

10. Go to the Select Chart Data Files pane.

11. Click the check boxes next to the files to include them in the download.

12. Go to the Select IO Modules Software pane.

13. Click the IO module that you want to include in the download.



Orchestra II Downloader Wizard
Create Packet

Select Download Methods to Master module.

● Create Packet for USB Flash Drive
○ Create Packet to Download through WiFi Module

Configure Master Module Software.

Select IO Modules Software.

☑ CL-712-100-20: Module2 ☑ Default (Software Major Version: 1, Software Minor Version: 11, Hardware Version: 0)

CL-712-100-20 Output package gen3_Project_CL-712-100-20_0111

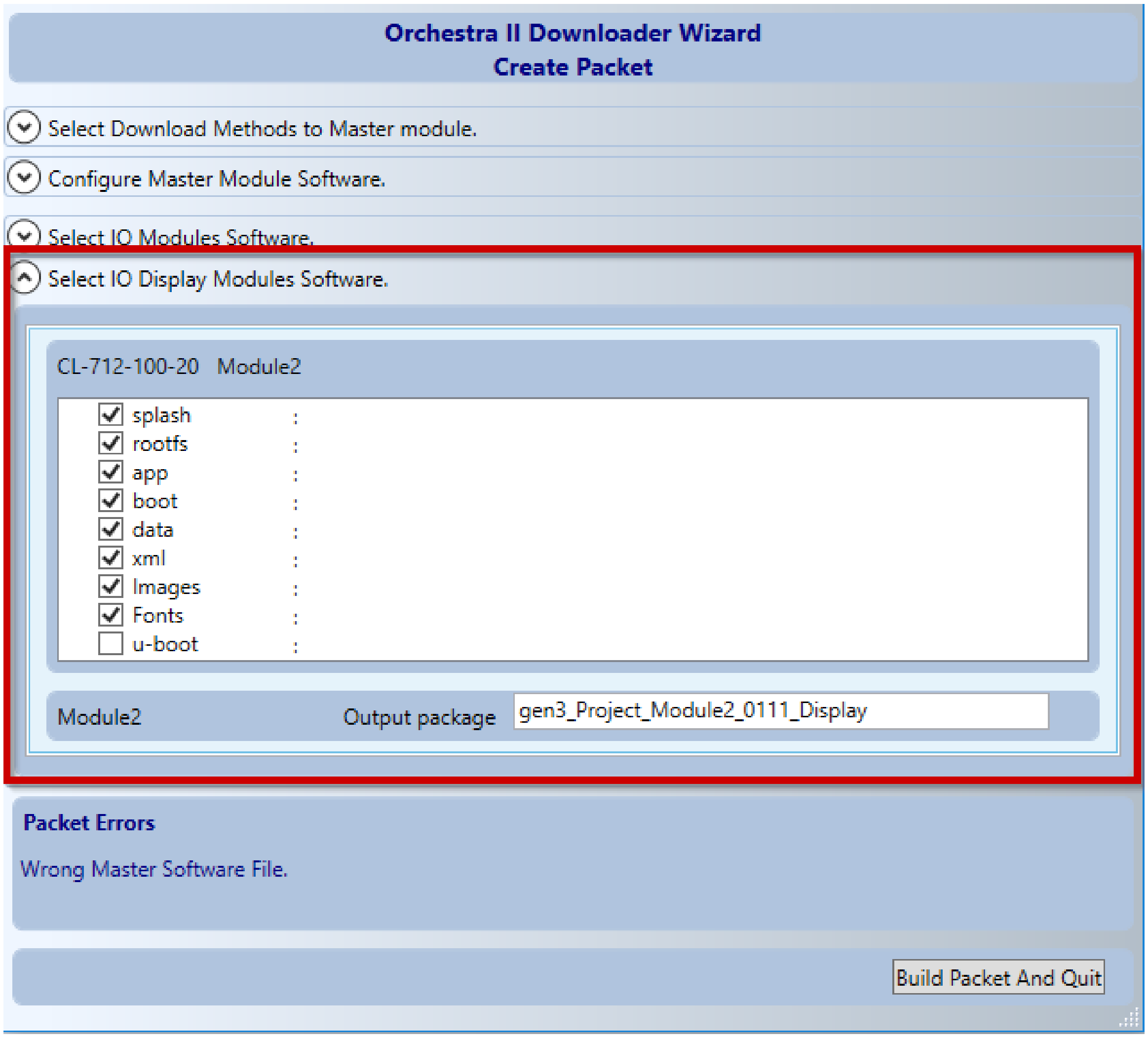
Select IO Display Modules Software.

Packet Errors

Wrong Master Software File.

Select IO Modules Software

14. Go to the Select IO Display Modules Software pane.



Orchestra II Downloader Wizard
Create Packet

Select Download Methods to Master module.

Configure Master Module Software.

Select IO Modules Software.

Select IO Display Modules Software.

CL-712-100-20 Module2

☑ splash :
☑ rootfs :
☑ app :
☑ boot :
☑ data :
☑ xml :
☑ Images :
☑ Fonts :
□ u-boot :

Module2 Output package gen3_Project_Module2_0111_Display

Packet Errors

Wrong Master Software File.

Select IO Display Modules Software

15. Click the check boxes to include the software with the module in the download.
16. Resolve any packet errors.
17. Click the Build Packet and Quit button.

Related:

Create USB Flash Drive Packet 241
Compile a Project 330
Alerts Panel 32

Appendix

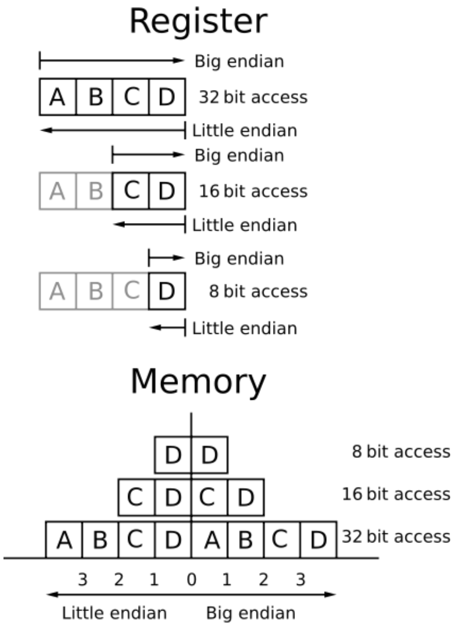
11 Appendix

11.1 Properties List

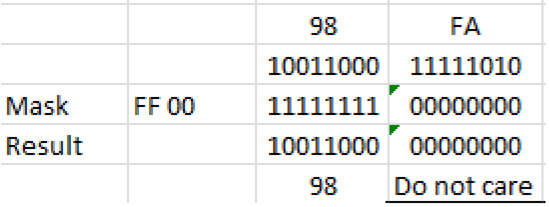
Property	Description
Array	Clicking this brings up a drop-down menu to edit an array that will be associated with the Variable.
A	Target data item that will have its value updated based on the outcome of the Operator Block.
Active Columns Count	This will limit the number of columns the user will be able to scroll through to only what is active or populated at that time.
Active Rows Count	This will limit the number of rows the user will be able to scroll through to only what is active or populated at that time.
Adj Loop Time (msec)	Select how long the run time loop takes on the master module. If the application overruns the loop time, it can cause the firmware to not operate correctly. The user needs to define this so it will not overrun Use Conductor™ to see how long the loop time currently takes.
Arguments	Field for a string list that is set up in the Resources section. The string list needs to be of the ListFormat of FileName Format.
Background Color	Choose the color of the screen background with this option.
Background Image	If an image is desired for the background, clicking the "…" button will allow the user to select an image from a location on their computer.
Border Color	Choose the color of the border surrounding the label.
Border Style	Choose the style of border the label possesses.
Border Width	Determines how thick the border is.
Brightness	This value controls how bright the video will appear in a range of 0 to 255, where 255 is the brightest.
Byte	Appears when Data Byte Filter is Enabled, input what data to filter on.
CAN Line	Select the CAN line in which the message is being received on.
CC Offset	This property sets the value of the duty cycle for the current closed loop control to begin at when going from a command of 0 to a non-zero command.
Center Base Color	This property controls the color of the circular base of the needle that lies below the needle hub.
Center Base Diameter	Change the size of the center base, in pixels.
Center Deadband	The amount of movement that is required from the center position to consider the Input active.
Center Hub Color	Property to determine the color of the needle hub.
Center Hub Diameter	Change the size of the hub, in pixels.

Center Location	Determines the horizontal and vertical positions that the center of the needle hub resides within the widget.
Center X	Determines the horizontal position of the center of the needle hub within the widget, in pixels.
Center Y	Determines the vertical position of the center of the needle hub within the widget, in pixels.
Channel	This property directs the widget to the channel in which the video feed will be coming from.
Color Depth	The number of bits that are available for RGB.
Color Saturation	This value controls the saturation of the colors within the video in a range of 0 to 255 where 255 is highest saturation level.
Columns Count	Defines the number of columns the table will have.
Command	Choose FWUA, TouchScreenCalibrate, or TouchScreenVerify. Use FWUA to download to the module. Use TouchScreenCalibrate to initialize the calibration program for touch screens. Use TouchscreenVerify to run the verification program for touch screens.
Compile Option	Choose Presto™, Presto™ with Rungs, or Rungs.
Contrast	This value controls the video contrast in a range of 0 to 255 where 255 is the highest level of contrast.
Corner Radius	Changing this value will affect how rounded the corners of the label appear to be.
Current Feedback Type	Can choose from Single wire or Dual wire to measure the current feedback of the Output. a. Single Wire – This mode uses an approximation when determining the current feedback for use in a current controlled application. This method is simpler and less I/O intensive, as it only uses a single wire connection on the Module, but it can be inaccurate based on approximations. Use this mode if the current control does not need to be precise. Dual Wire – This mode uses another I/O connection to receive the current feedback from the device connected to the Output. With a return line for the current, a more precise measurement can be made and Output adjusted for a current controlled application. Use this mode if precision is critical.
Current Report Rate	When the Diagnostic Requirement check box for Over current is checked, this option appears. This value determines how often, in milliseconds, that the current on the Output is reported for monitoring. Unless it is absolutely necessary, it is generally preferred to keep the rate at 2550mS.
D gain	The derivative gain associated with the PID loop
Data Byte Filtering	Enables the ability to filter a message by a specific byte. When this is enabled, another field will show in which the user will be able to create a

	mask to allow only the desired bytes of information through.
Data Byte Order	This property allows the user to reorder the bytes to be transmitted from the bytes that were received.
Data Length	Appears if the Data Length Adjustment is Enabled, allows user to define the length of the message to be transmitted.
Data Length Adjustment	Appears if the Data Length Adjustment is Enabled, allows user to define the length of the message to be transmitted.
Data Parsing Type	This field allows the user to choose the format of the desired message in either Bits or Bytes.
Data Resolution	Adjust the frequency resolution of the signal from 0.01, 0.05, 0.1, 0.5, 1, and 2Hz.
Debounce OFF (msec)	The amount of time the input must be open before it will switch from the active state to the inactive state.
Debounce ON (msec)	The amount of time the Input must be not open before it will switch from the inactive state to the active state.
Default Array Value	This is the global default value for all elements within the array that have not had a default value manually set already.
Default Rx Status	On power up, this is the default status of the CAN receive.
Default Rx Value	On power up, this is the default value of the CAN receive.
Default Value	Sets the default value of that element within the array.
Delay (0-2550 mS)	When the Diagnostic Requirement check box for Over Current is checked, this option appears. This option lets the user select how long the Output can be in an overcurrent condition before it blows the Digital Fuse and turns the Output to the Device Off.
Diagnostic Requirements	This field has a drop-down box of 5 requirements that can be monitored for diagnostics if so desired. a. Short to Battery – Sets a diagnostic if the Output has detected it is shorted to “Battery”. b. Short to Ground – Sets a diagnostic if the Output has detected it is shorted to “Ground”. c. Open When Off – Sets a diagnostic if the Output has detected an Open condition when commanded Off. d. Open when On – Sets a diagnostic if the Output has detected an Open condition when commanded On. Over Current – Sets a diagnostic if the Output detects an overcurrent condition. Additional options appear when this diagnostic is selected; Current Report Rate, Digital Fuse Delay, and Digital Fuse Set Point.
Direction	This option allows the user to decide if the Counter will increment (count up from the default to max) or decrement (count down from the max to 0).
Direction CAN	Selects the way in which the message was transmitted and stored into memory; either LSB->MSB (least significant byte to most significant byte)

	or MSB->LSB. Below is an example of the difference for Bytes A B and C.  Register Big endian A B C D 32 bit access Little endian Big endian A B C D 16 bit access Little endian Big endian A B C D 8 bit access Little endian Memory D D 8 bit access C D C D 16 bit access A B C D A B C D 32 bit access 3 2 1 0 1 2 3 Little endian Big endian **Direction CAN** a. If the Parsing Type is Bits, then there is no Direction field. Instead, there is a Start Bit field that the user is able to choose at which bit in the message to begin reading.
Display Family	HED® has different display families. This selects the different features available on the screen.
Display Format	The user is able to choose the display format of the message in either Hex or decimal.
Display Index	This is the value to put in the Screen Selection to display that screen.
Display Model	This selects the model of display for which the screen will be associated with.
Encrypt File	Choose True or False. This selects if the IOC file is encrypted or not. Recommended: set to True to encrypt the file, unless the encryption is failing.
End Point (degrees)	Dictates the angle that the needle will travel when at its max value.
End Point (Pixel)	Define how many pixels from the edge of the widget that the bar will end i.e. a value of 0 will fill the bar completely and a value of 10 will fill the bar until the edge is 10 pixels away from the edge of the widget.
End Value	The value that the needle widget will be when it reaches the end point.
Execute	Causes the widget to trigger when true. Defaults to false, with option to set to always run. Also, this can be linked to a data item to trigger the execute command when the data item is greater than 0.

Fill Color	Choose the color that the bar will be to fill the widget.
Fill Direction	Choose the direction that the bar will fill up. In a horizontal orientation, it can fill left to right or right to left. In a vertical position, it can be filled top to bottom or bottom to top.
Fill Start Point (Pixel)	This is the point that the bar will begin filling from, in reference to the start point.
Fill Start Point (Degrees)	This is the angle where the widget will start to fill the progress bar.
Filter Size	Used only for Running Average, and sets how many samples, taken independent of the Report Rate, to average together in order to obtain a value to report back.
Filter Type	There are two types of filters, running average and min/max average. These are software filters used to “clean up” a signal. Running average takes the average over a number of samples. The Min Max Average averages the Min and Max voltages read on the pin, and places that value in the data item each time the Input value changes direction.
Flashing	Determines whether or not the label is flashing.
Flash Off Time (ms)	The amount of time the image will not be visible when flashing.
Flash On Time (ms)	This number determines how quickly the label will flash when in a flash state.
Flyback A	Available only for Single Wire Current Controlled Output. A calculated value, using the Flyback Calculation.xls worksheet, that factors into the approximation of the feedback current when using Single Wire with Flyback Approximation Enabled.
Flyback Approximation	Available only for Single Wire Current Controlled Output. Enabling this allows the use of the Flyback A, B, and C properties to assist in Single Wire current control approximations. This property should be enabled if the Output has an inductive load.
Flyback B	Available only for Single Wire Current Controlled Output. A calculated value, using the Flyback Calculation.xls worksheet, that factors into the approximation of the feedback current when using Single Wire with Flyback Approximation Enabled.
Flyback C	Available only for Single Wire Current Controlled Output. A calculated value, using the Flyback Calculation.xls worksheet, that factors into the approximation of the feedback current when using Single Wire with Flyback Approximation Enabled.
Font	Choose a font from the list of supported fonts.
Font Bold	A true value bolds the font.
Font Color	Choose the color of the font.
Font Italic	A true value italicizes the font.
Font Size	Adjust the size of the font.

<table>
<tr><td>Font Size Height</td><td>The height of the font in pixels. Adjusted automatically based on the font size property.</td></tr>
<tr><td>Font Underline</td><td>A true value underlines the font.</td></tr>
<tr><td>Frequency (Hz)</td><td>This field is where to input the frequency desired for the Output between 40 and 5000Hz.</td></tr>
<tr><td>Grid Line Color</td><td>Choose what color to make the grid lines.</td></tr>
<tr><td>Grid Lines Orientation</td><td>Choose what grid lines are visible; none, horizontal, vertical, both.</td></tr>
<tr><td>Groups</td><td>Can assign a group that the Input or Output is associated with to be used in Conductor™ for easy management and viewing of specific Inputs and Outputs.</td></tr>
<tr><td>Height</td><td>The vertical size, in pixels, of the widget.</td></tr>
<tr><td>Highlight Orientation</td><td>Choose the way to scroll through highlighted areas; moving horizontally, vertically, or cell by cell.</td></tr>
<tr><td>Highlighted Column</td><td>Determine how many columns to highlight at a time.</td></tr>
<tr><td>Highlighted Row</td><td>Determine how many rows to highlight at a time.</td></tr>
<tr><td>Hightlight Color</td><td>Choose what color indicates that the row(s) and/or column(s) are highlighted.</td></tr>
<tr><td>Horizontal Alignment</td><td>The text can be left, center, or right justified.</td></tr>
<tr><td>Horizontal Margin</td><td>Used as spacing between the right and left sides of the font on the respective widget.</td></tr>
<tr><td>Hue</td><td>This value controls the video hue in a range of 0 to 255 where 255 is the highest level of hue.</td></tr>
<tr><td>I gain</td><td>The integral gain associated with the PID loop.</td></tr>
<tr><td>ID</td><td>This is the identifier of the CAN message that the user wants to get a value out of.</td></tr>
<tr><td>ID Length</td><td>Determines the length of the message ID; either 11-bit or 29-bit can be chosen.</td></tr>
<tr><td>ID Mask</td><td>The user can define a mask to use for the message ID in order to filter for the correct message from the desired location. A mask is a Decimal or Hex, pending on Parsing Type, representation of a binary number, that when compared with the binary version of the message ID, allows specific bits to "fall through". Those bits are the ones that form the valued portion of the message ID i.e. for a message ID of 98 A3 in Hex and the only portion of the ID that really matters is the 98 a mask of FF 00 should be used.

<table>
<tr><td></td><td></td><td>98</td><td>FA</td></tr>
<tr><td></td><td></td><td>10011000</td><td>11111010</td></tr>
<tr><td>Mask</td><td>FF 00</td><td>11111111</td><td>00000000</td></tr>
<tr><td>Result</td><td></td><td>10011000</td><td>00000000</td></tr>
<tr><td></td><td></td><td>98</td><td>Do not care</td></tr>
</table>
<b>Identifier Mask</b></td></tr>
</table>

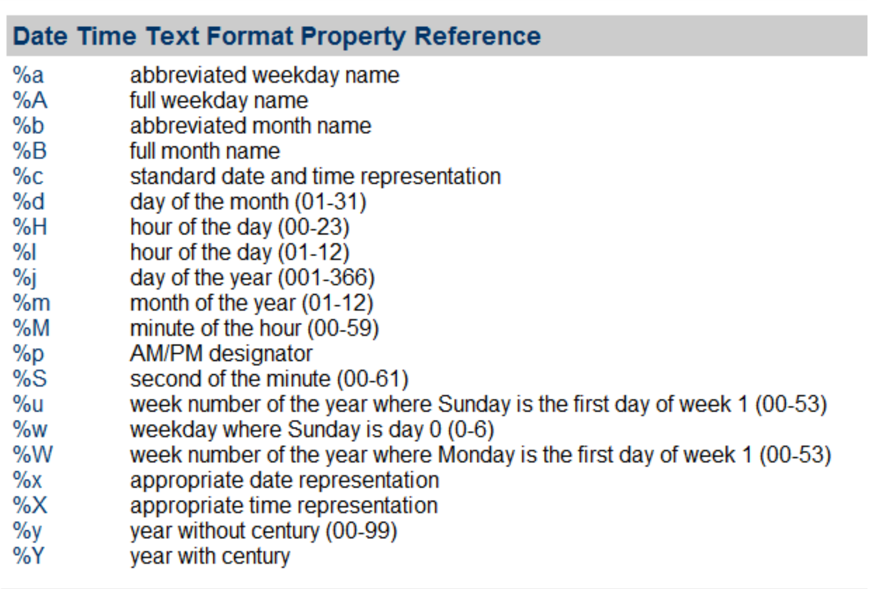
	The converted value from Hex to binary in the example is compared to the binary version of the ID and a bitwise AND operation is done. Anything compared with a 1 is what is desired and anything compared with a 0 is a do not care.
ID Value Adjustment	Enables or disables the ability to choose an ID Length.
Identifier	This is where the user defines the message ID to look for in order to receive the proper message. The ID can either be defined in Hex or Decimal depending on the display format.
Identifier Mask	This sets which bits in the identifier need to match for incoming messages.
Image	Path to and select the desired image to be displayed.
Input	The input signal that will be used as a reference to run the PID loop
Input Center	This is the value of the Input when in a rest position
Input Deadband	A range above and below, depending on PID Operator Block, the Input Target that provides the system some hysteresis
Input Max	This is the maximum value of the Input
Input Min	This is the minimum value of the Input
Input Mode	*Menu* that determines the type of Input.
Input Target	The desired value that the PID operation drives to achieve
Is Highlight Enabled	Enable or disable if effect of highlighting.
Is Touchable	Sets whether or not this item can be interacted with via touchscreen.
K0 Gain	Gain for the current control and can be derived from the equation: $(Propotional\ Gain + (Inegral\ Gain\ \times Time))$ where Time is the loop time.
K1 Gain	This is the proportional gain for the current control.
Latching	Setting this False will set the Input state for ON when the pin is active and OFF when it is inactive, similar to a momentary switch. If set True, when the pin changes from inactive to active, the Input will toggle between reporting ON and OFF and hold that value until the next transition from inactive to active.
Length	Determines the size of the data being read in which is determined by the Parsing Type. For Bytes the Length can be 1 or 2 and for Bits 1 to 16.
Location	Determines where the upper left corner of the widget resides on the screen where 0,0 (pixels) is the upper left most corner of the screen.
Major Version	User can use this for setting the version number of their application.
Mask	This mask is for filtering specific data within the message to pass through.
Max	This field is used to set what the max value of the Variable can be.

Max Frequency (1-10000Hz)	Set what the max expected frequency will be from 1 to 10000Hz.
Max Input Resistance (?)	The Max Input Resistance can be changed with a range of 1? to 65535? .
Max Input Voltage (mV)	The Max Input voltage is used for validation inside of orchestra to ensure the user assigns it to a pin that can support the range.
Max Value	This is the maximum value that we allow the user to set the data item to
Max+	The amount allowed for the Input to go above the Input Max before it is considered and error
Memo	Space for an optional internal note for the user to use if desired.
Min	This field is used to set what the min value of the Variable can be.
Min-	The amount allowed for the Input to go below the Input Min before it is considered and error.
Min Transmit Period	This is the amount of time, where 100 is equal to 1 second, that must elapse before the message is transmitted from an I/O module to the Master Module. This property can be used to slow down the message some by only receiving it on given intervals.
Minor Version	User can use this for setting the version number of their application.
Mission Critical Settings	Here is where the Mission Critical Settings for what the Output will do when in a Mission Critical condition. a. Turn Off – The Output is turned off when in a Mission Critical condition. b. Turn On – The Output is turned on when in a Mission Critical condition. c. Turn On and Flash – The Output is turned on and Flash setting is enabled when in a Mission Critical condition. d. Maintain Current State – Commands the Output to maintain whatever state it was in when the Mission Critical condition was detected.
Module	This is the module that will receive the CAN message.
Module Type	Part number of the selected module.
Multi-Line	This option if true, allows the label to have multiple lines and will wrap the text to fit within the defined size of the label.
Name	Space to enter a unique name for the Input/Output/Display/Property. For Compile Option: Set automatically, based off of file name.
Number of Elements	Defines the number of elements within the array.
Off Time	This determines the length of time, in milliseconds, for the Output to be off, within the Period, when in a Flash condition. This value must be less than the Period value.

Offset	Optional setting used for adjusting the incoming value to the desired units for the telematics service, or linking to a display widget.
On Time	This determines the length of time, in milliseconds, for the Output to be on, within the Period, when in a Flash condition. This value must be less than the Period value.
Orientation	This determines how the bar is placed on the screen, in either a vertical or horizontal fashion.
Outline Color	Choose the color of the needle outline.
Outline Path Color	Choose the color that the outline path will have.
Outline Path Radius	This is the radius that the outline will follow.
Output Max	This is the max value that will be placed within the A term to drive an output based on the result of the PID calculation
Output Max Current	This sets what the maximum amount of current that the application will draw. This is only used to ensure that when they assign the output to a module they pick an output that will support the current. Does not affect run time at all.
Output Max(mA)	Define the maximum current for the Output to command, in milliamps. Please make note that not all Outputs on all modules have the same current limitations, please refer to the specific module data sheet to determine what the Output can allow.
Output Mode	This field is where the mode of Output is chosen; Digital, PWM, Constant Current, or Frequency.
Output Scaling	This is a scaling factor for the output value where 1000 is equal to a factor of 1
Output Threshold	This is the minimum value that will be placed within the A term to drive an output based on the result of the PID calculation
Output Type	This field sets whether the Output will be Sourcing, Sinking, or a Servo. a. Sourcing – The Output is sourcing the current to the device, connecting the pin to “Battery” when On. b. Sinking – The Output is sinking the current from the device, connecting the pin to “Ground” when On. c. Servo – This mode allows the Output to both sink and source the current to the device allowing it to be connected to either “Ground” or “Battery” when On and Open when Off.
Overlay Image	Path to and select an image to overlay the needle if desired.
Overlay Location	This is the location the upper left corner of the overlay image will appear where 0,0 is the upper left most pixel of the needle widget.
Overlay X	The horizontal location of the overlay image, in pixels, on the needle widget.

Overlay Y	The vertical location of the overlay image, in pixels, on the needle widget.
P gain	The proportional gain associated with the PID loop.
Password Level 1	This is the password attached to Level 1 in Conductor™.
Password Level 2	This is the password attached to Level 2 in Conductor™.
Password Level 3	This is the password attached to Level 3 in Conductor™.
Period	This value is what determines how long the period of the Flash, typically found by adding the Off Time and On Times together.
Pointer Color	This adjusts the pointer color.
Pointer Length	The length of the pointer part of the needle, in pixels.
Pointer Style	Able to choose between a triangle or block style. The triangle will come to a point at the end while the block will have a uniform thickness throughout the entire length.
Pointer Width	Determines how thick the pointer is. For the triangle style the width is what the base starts at before tapering to the point.
Radius	This value will determine the size of the progress bar, in pixels. Increasing this number will increase the size of the circle that the progress bar will fill.
Read Security Level	Determines the level dongle needed to see the value of the Input/Output within Conductor™. This can be Level 1, 2, or 3 with 1 needing the highest security clearance.
Report Rate (msec)	The Report Rate controls how often, over CAN, the I/O Module will report the value on the pin, and can be changed from 10ms to 2550ms in increments of 10ms. Note that if a Master Module uses a VTD Input the Report Rate will not be taken into account and the value will be updated every loop.
Reset EEPROMS	Choose True or False. Applicable for Rungs project only. This tells the Application Downloader whether or not it should reset the EEPROMS to default during programming.
Reset Timers	Choose True or False. Applicable for Rungs project only. This tells the Application Downloader whether or not it should reset the EEPROMS to default during programming.
Resolution	Optional setting used for adjusting the incoming value to the desired units for the telematics service, or linking to a display widget.
Rest Point (degrees)	Dictates the angle at which the needle widget begins.
Rotation Direction	Controls the direction in which the needle will travel from its rest point.
Rows Count	Defines the number of rows the table will have.
Safe Mode Below Voltage	Choose True or False. Enables the firmware to detect when the voltage drops below the volt setting, and sets the run mode to safe mode until the voltage goes above this setting.
Safe Mode Low Volt Settings	Volt setting for safe mode configuration.

Safe Mode Settings	Here is where the Safe Mode Settings for what the Output will do when in a Safe Mode condition. a. Turn Off – The Output is turned off when in a Safe Mode condition. b. Turn On – The Output is turned on when in a Safe Mode condition. c. Turn On and Flash – The Output is turned on and Flash setting is enabled when in a Safe Mode condition. d. Maintain Current State – Commands the Output to maintain whatever state it was in when the Safe Mode condition was detected.
Save On Shutdown?	The default is No, changing this to Yes will save the value stored within the Time Counter when the system is shut down in order to resume at the same point when the system is restarted.
Screen Selection	Choose which screen to display during run time.
Set Point	If the output exceeds this current for the more than the delay time it will turn the output off and report it as over current.
Set Point (1-80000 mA)	When the Diagnostic Requirement check box for Over Current is checked this option appears. This option allows the user to determine the current setting that the Output must exceed before it detects an Over Current event.
Show Outline	Makes the outline visible or not.
Show Outline Path	This property controls whether or not the outline path will be visible.
Show Shadow	Creates a shadow below the needle to make it look 3D.
Size	Determines how large the widget/display will appear to be, comprised of the height and width 0,0 (pixels) respectively.
Slew Off	This is the amount of time it takes for the Output to go from 100% to 0, in milliseconds with a max time of 1000mS.
Slew On	This is the amount of time it takes for the Output to go from 0 to 100%, in milliseconds with a max time of 1000mS.
Source Type	Choose whether the signal is a sourcing or sinking Input.
Splash Image File	This is the image that is displayed during boot up of the display before the application starts drawing.
Start Byte	Defines the most significant byte, i.e. if LSB->MSB and start byte is 3 it would read in bytes 3 and 2 while MSB->LSB with start byte 3 would read in bytes 3 and 4.
Start Point (degrees)	Determines the angle at which the needle widget starts its travel from as an offset from the Rest Point. May not always equal the rest point.
Start Point (Pixel)	This value will determine how far from the edge the bar starts at.
Start Value	This is the value of the widget when at the start point.

State Enumerations	Clicking on this property brings up a mini menu to add states. Within that mini menu is where the user can also edit the names and numbers of those states that have been added.
System CAN	Choose Dual or Single. Sets if the clients can use single or dual CAN.
Tail Color	This adjusts the tail color.
Tail Length	The length of the tail part of the needle, in pixels.
Tail Style	Able to choose between a triangle or block style. The triangle will come to a point at the end while the block will have a uniform thickness throughout the entire length.
Tail Width	Determines how thick the tail is. For the triangle style the width is what the base starts at before tapering to the point.
Text Format	This is the information the label will display from either a manual entry or a string list.  Date Time Text Format Property Reference %a abbreviated weekday name %A full weekday name %b abbreviated month name %B full month name %c standard date and time representation %d day of the month (01-31) %H hour of the day (00-23) %I hour of the day (01-12) %j day of the year (001-366) %m month of the year (01-12) %M minute of the hour (00-59) %p AM/PM designator %S second of the minute (00-61) %u week number of the year where Sunday is the first day of week 1 (00-53) %w weekday where Sunday is day 0 (0-6) %W week number of the year where Monday is the first day of week 1 (00-53) %x appropriate date representation %X appropriate time representation %y year without century (00-99) %Y year with century **Text Format Property Reference**
Text Location	Determines the location of the text within the widget where 0,0 are the X and Y coordinates, in pixels.
Text Visible	This options controls whether or not the text will be seen.
Text X	The horizontal location within the widget that the text will begin, in pixels.
Text Y	The vertical location within the widget that the text will begin, in pixels.
Time Interval	This determines when the Counter will either increment or decrement. Loop Time – Runs as quickly as the application is executed, ~10mS. a. 1 Second – Will run once a second. b. 10 Seconds – Executes once every 10 seconds. c. 1 Minute – Executes once every 1 minute. d. 10 Minutes – Executes once every 10 minutes.

Touch Size	This determines the size of the area that will register a touch for the item when using a touchscreen.
Translation	This option determines what language the text will be displayed in. Leaving it on the “follow display” option allows the language to be changed dynamically by only adjusting the language of the display, otherwise the translation will remain static to what is chosen with this property.
Tx Rate	The rate in which a message is transmitted based on the 10ms loop time.
Tx Status	Determines when a message can be transmitted. a. Disabled – This message will not be transmitted, can act as a stop. b. Pass Through – The message will be passed through as it is received. c. On Report Rate – Message will be transmitted at the interval determined by the Tx Rate property.
Type	This property sets what the size of the Variable can be either a 16-bit unsigned or 32-bit unsigned (65,535 or 4,294,967,295). There is an option for Alarm as well which is used in a case of the Variable being an alarm to send notice to the network through the telematics Module.
Units	This is an option field to associate a unit description with the Variable to be viewable within Conductor™.
Upper Left Column	Dictates what column will be the starting column to be used or viewed within the table.
Upper Left Row	Dictates what row will be the starting row to be used or viewed within the table.
Value	Input variable used within the widget/constant.
Vertical Alignment	The text can be vertically top, center, or bottom justified.
Vertical Margin	Used as spacing between the top and bottom sides of the font on the respective widget.
Visible	Determines if and when the label is visible. Linking this property to a data item can determine when the label becomes visible on the screen based on the data item returning a true of false value.
Width	The horizontal size, in pixels, of the widget.
Wire Number	Space for an internal note to document the Input’s or Output’s wire or harness number if desired.
Write Security Level	Determines the level dongle needed to edit the value of the Input/Output within Conductor™. This can be Level 1, 2, or 3 with 1 needing the highest security clearance.
X	The horizontal position, in pixels, on the screen where the label resides.
Y	The vertical position, in pixels, on the screen where the label resides.
Z Order	The lower this number is the lower it will be in the “layer” meaning it will be drawn before higher numbers. The higher numbers will lay over the lower

	“layers”, so the user must be careful not to cover up something that they wish to be visible with a higher z ordered widget.

Related:

Glossary 263

11.2 Revision History

Rev No.	Description	Date	User
1	Initial rough draft release	3/31/2014	J. Kothrade
2	Updated GUI and screenshots, added Application Notes 268 and Troubleshooting Guide.	9/4/2018	K. Oscar and J. Kothrade
3	Integrated Conductor 168 Manual	10/19/2018	K. Oscar and J. Kothrade

11.3 Glossary

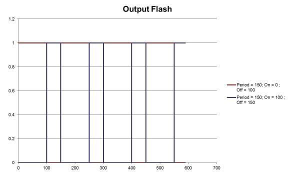
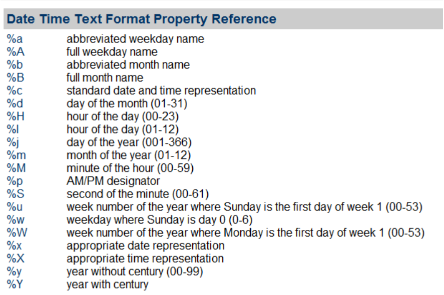
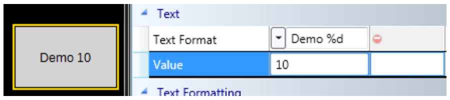
Term	**Definition**
Application	Software created by the user to control module specific functions. This software is downloaded to the master module.
Baud	Symbol rate or modulation rate in symbols per second or pulses per second.
CAN	Please reference Bosch 2.0 A and B Controller Area Network Specification.
CAN Link Module	HED® product utilizing the CANLink® protocol.
CAN Link Protocol	HED® proprietary J1939 compatible CAN protocol.
Display	Programmable piece of hardware that can give a visual representation of the application.
Flash	 **Flash Example**
HarnID	Abbreviation for Harness ID. Harness IDs are used by I/O modules on the system, particularly when there are multiple I/O modules of the same type. Harness IDs are used by control to tell the modules apart. **There are two ways to set Harness IDs in Orchestra®.** 1. Use 1 to 4 pins on the module. 2. Set the internal software to a specific Harness ID. **Range:** 0-15 in decimal, 0-F in hex.
I/O	Abbreviation for Input/Output. A module's means of interface to the physical world.
Modes	Select what will be edited; a single cell, entire row, or entire column.
Screen Resolution	Changing the Display Model will adjust the resolution automatically.
Selected Properties	This allows the user to select a specific cell, or edit the height or width of the selected row or column respectively.
Show Shadow	Choose whether or not to show a shadow effect of the needle. Please note that depending on the background color the shadow may not show or show well.

Table Area	Rows and columns of the created table.
Text Format	This is the information the label will display from either a manual entry or a string list.  **Date Time Text Format Property Reference** %a abbreviated weekday name %A full weekday name %b abbreviated month name %B full month name %c standard date and time representation %d day of the month (01-31) %H hour of the day (00-23) %I hour of the day (01-12) %j day of the year (001-366) %m month of the year (01-12) %M minute of the hour (00-59) %p AM/PM designator %S second of the minute (00-61) %u week number of the year where Sunday is the first day of week 1 (00-53) %w weekday where Sunday is day 0 (0-6) %W week number of the year where Monday is the first day of week 1 (00-53) %x appropriate date representation %X appropriate time representation %y year without century (00-99) %Y year with century **Supported Text Formatting**  Demo 10 Text Text Format: Demo %d Value: 10 Text Formatting **Label Text Formatting Example**
Widget Properties	The user can select which cell, row, or column will have either an image or label. After selecting image or label, those properties will appear within that window to be edited.

Related:

Properties List 248

Term Definitions 13

Application Notes

12 Application Notes

12.1 Orchestra® Quick Start Guide

12.1.1 Introduction

This Quick Start guide explains how to set up a new Orchestra® project.

Related:

Create a New Project 268
Add Modules 272
Add Screen 284

12.1.2 Create a New Project

Create a new Orchestra® project by plugging a license dongle into the computer, opening Orchestra®, adding modules, and adding screens.

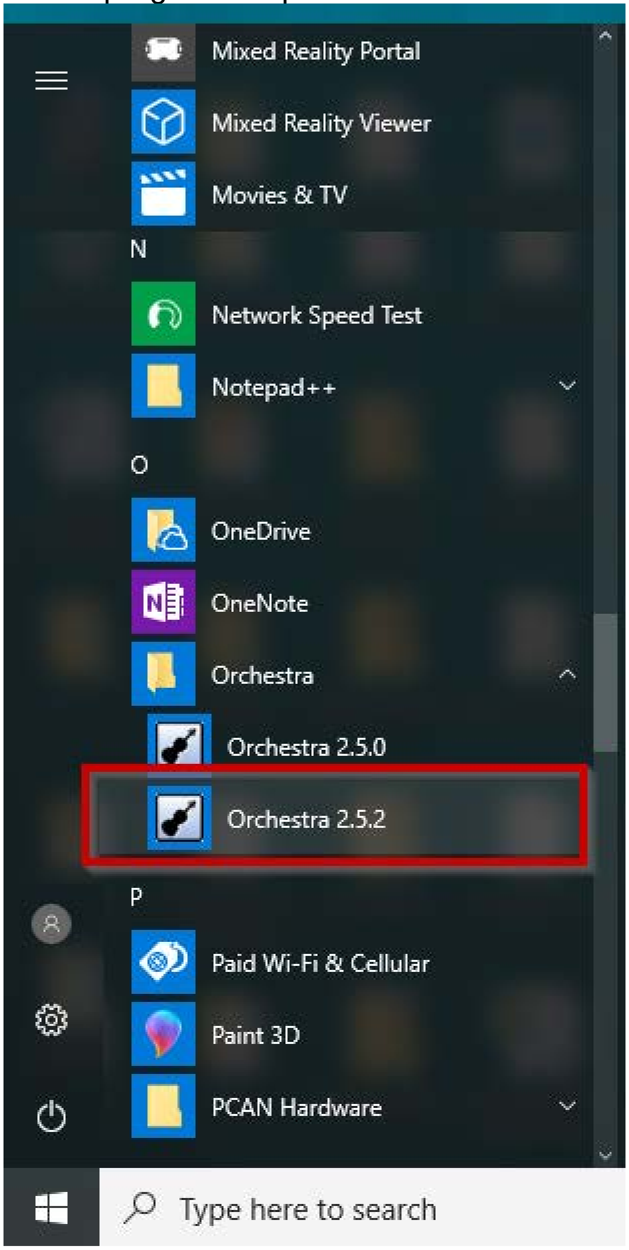
Related:

Open Orchestra 269
Start a Project 271
Add Modules 272
Add Screen 284

12.1.2.1 Open Orchestra®

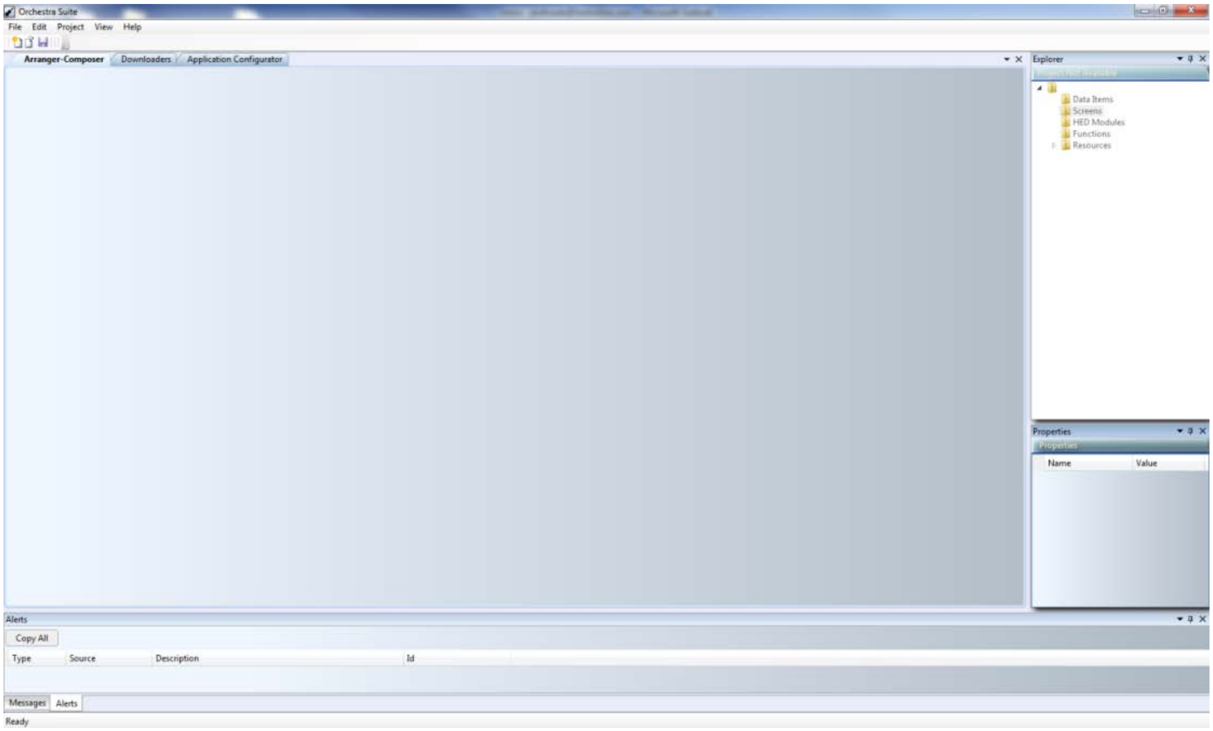
Create a new project by opening Orchestra®.

1. Open the Start menu.
2. Find the Orchestra® program in the All Programs pane.
3. Click the Orchestra® program to open it.



Opening Orchestra®

A window like the one below opens.



Orchestra® Beginning Screen

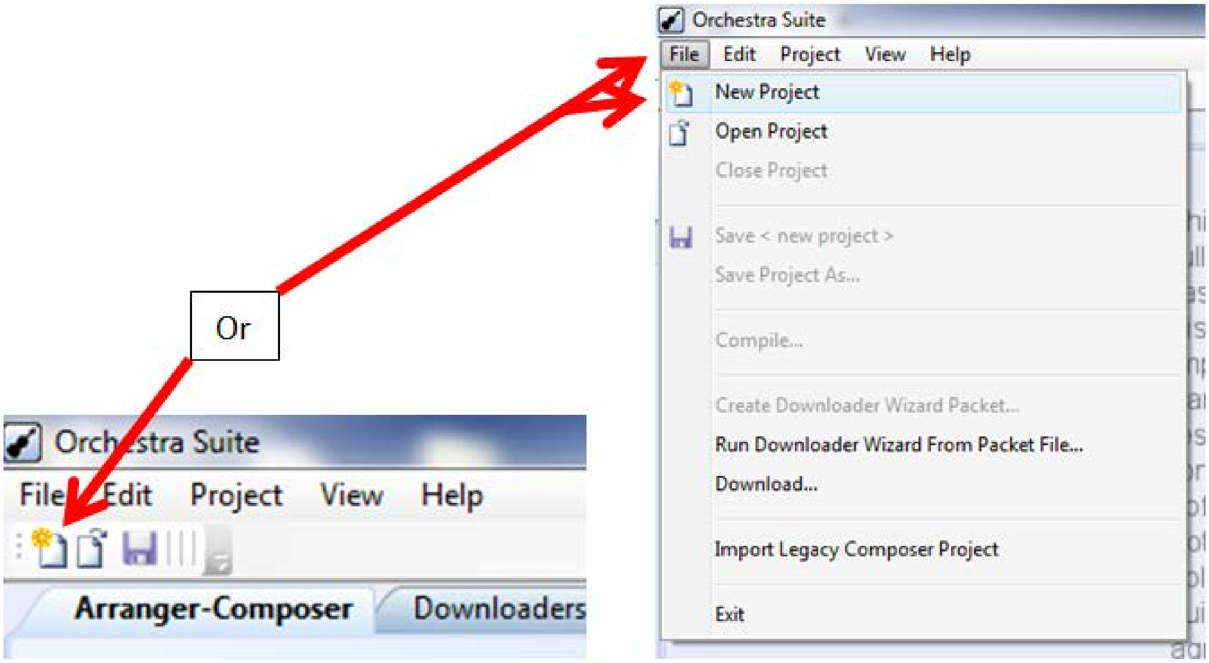
Related:

[Start a Project] 271

12.1.2.2 Start a Project

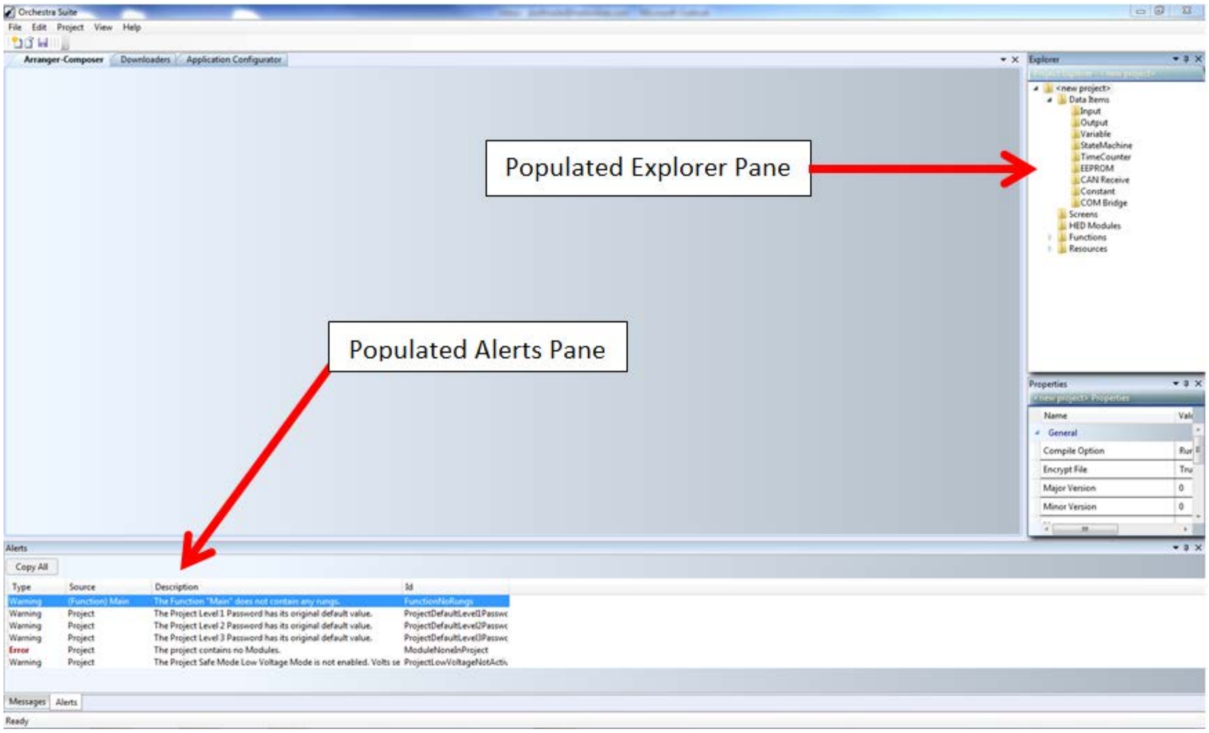
There are two ways to start a new project.

1. Click the blank page icon underneath the File menu.
2. Go into the File drop-down menu and select New Project.



Creating New Project

3. Once the new project is created, the Explorer and Alerts panes populate.



Populated Explorer Pane

Populated Alerts Pane

Name	Val
General	
Compile Option	Rur
Encrypt File	Tru
Major Version	0
Minor Version	0

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc
Warning	Project	The Project Level 2 Password has its original default value.	ProjectDefaultLevel2Passwc
Warning	Project	The Project Level 3 Password has its original default value.	ProjectDefaultLevel3Passwc
Error	Project	The project contains no Modules.	ModuleNoneInProject
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActiv

Explorer and Alerts Panes

Related:

Open Orchestra 269

12.1.2.3 Add Modules

Use the Module tab to add a master module to the project. Then, add IO modules or display modules.

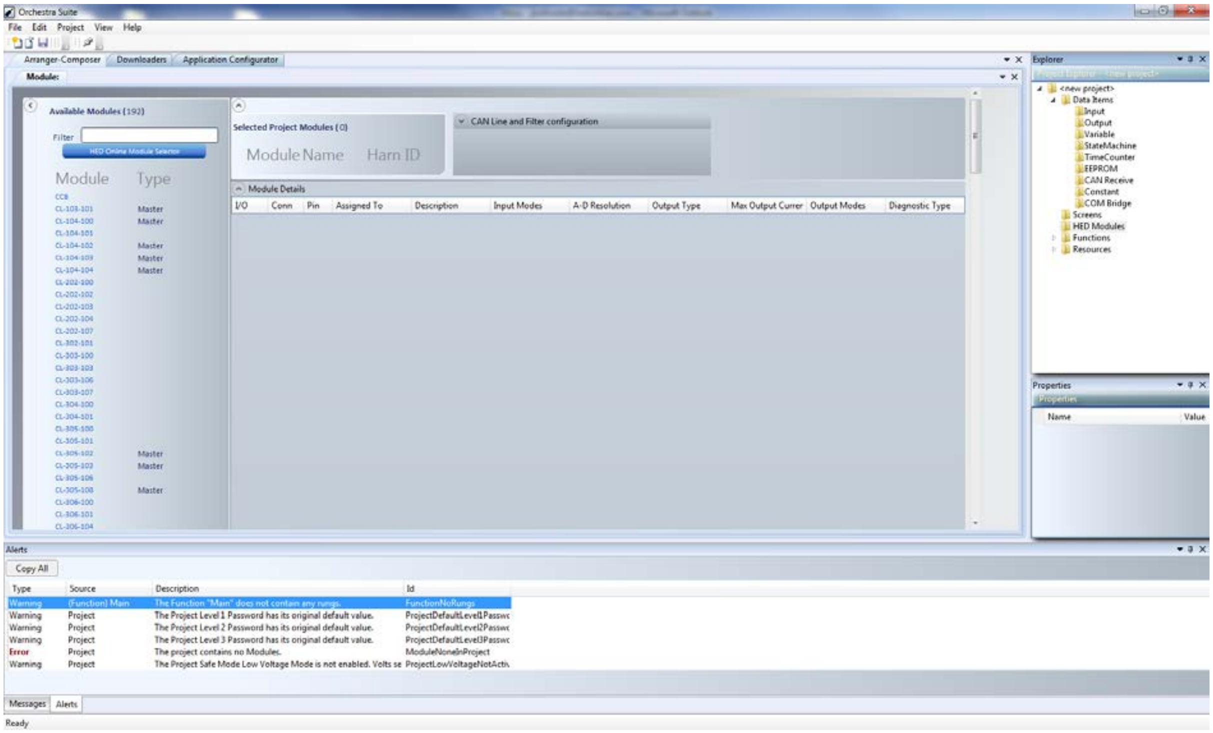
Related Pages:

Add a Master Module 273
Add I/O Modules 281
Module Tab 40

12.1.2.3.1 Add a Master Module

To continue creating a new project, add a master module to the project.

1. Right click on the HED® Modules folder.
2. Left click the “Select Module(s)…” option that appears.
3. The Designer pane populates with an interactive list of all of the modules supported within Orchestra®, as well as displaying whether or not that particular module is a master module.
4. Select a master module.



Available Modules (192)

Module	Type
CCB	
CL-103-101	Master
CL-104-100	Master
CL-104-101	
CL-104-102	Master
CL-104-103	Master
CL-104-104	Master
CL-202-100	
CL-202-102	
CL-202-103	
CL-202-104	
CL-202-107	
CL-302-101	
CL-303-100	
CL-303-103	
CL-303-106	
CL-303-107	
CL-304-100	
CL-304-101	
CL-305-100	
CL-305-101	
CL-305-102	Master
CL-305-103	Master
CL-305-106	
CL-305-108	Master
CL-306-100	
CL-306-101	
CL-306-104	

Alerts

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc
Warning	Project	The Project Level 2 Password has its original default value.	ProjectDefaultLevel2Passwc
Warning	Project	The Project Level 3 Password has its original default value.	ProjectDefaultLevel3Passwc
Error	Project	The project contains no Modules.	ModuleNoneInProject
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActiv

Module Selection List

5. The selected master module is added to the HED® Modules folder.
6. The Module Details section fills with all of the detailed pin assignments.

Orchestra® automatically removes all other master modules from the list, since only one master module can exist in each application.

Related:
Add I/O Modules 281

12.1.2.3.2 Assign Data Items to Pins

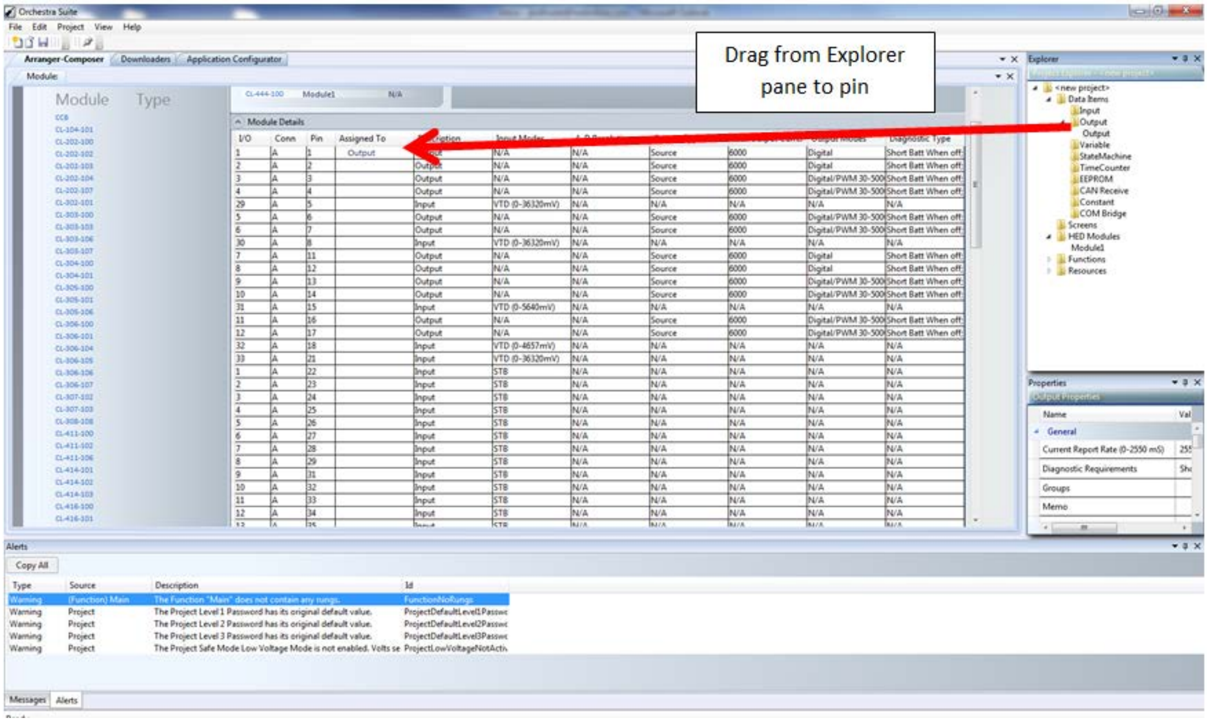
The user can drag and drop the proper data items to a pin to assign and link that data item with that pin on the module. To do that, create a data item using the Explorer Pane.

Related:

Create a Data Item 274
Data Items 54

12.1.2.3.2.1 Create a Data Item

1. Go to the Explorer Pane.
2. Right click on the data item that matches the pin assignment requirements.
3. Click the Add option.
4. The data item is shown as a subitem in its respective subfolder.
5. Edit the name and other properties by highlighting it and using the Properties Pane.
6. Match the data item properties to the pin assignment requirements.



Module Type

CCB
CL-104-101
CL-202-100
CL-202-102
CL-202-103
CL-202-104
CL-202-107
CL-302-101
CL-303-100
CL-303-103
CL-303-106
CL-303-107
CL-304-100
CL-304-101
CL-305-100
CL-305-101
CL-305-106
CL-306-100
CL-306-101
CL-306-104
CL-306-105
CL-306-106
CL-306-107
CL-307-102
CL-307-103
CL-308-108
CL-411-100
CL-411-102
CL-411-106
CL-414-101
CL-414-102
CL-414-103
CL-416-100
CL-416-101

CL-444-100 Module1 N/A

I/O	Conn	Pin	Assigned To	Description	Input Modes	A/D Resolution	[illegible]	Output Current	Output Modes	Diagnostic Type
1	A	1	Output	Output	N/A	N/A	Source	6000	Digital	Short Batt When off
2	A	2		Output	N/A	N/A	Source	6000	Digital	Short Batt When off
3	A	3		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
4	A	4		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
29	A	5		Input	VTD (0-36320mV)	N/A	N/A	N/A	N/A	N/A
5	A	6		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
6	A	7		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
30	A	8		Input	VTD (0-36320mV)	N/A	N/A	N/A	N/A	N/A
7	A	11		Output	N/A	N/A	Source	6000	Digital	Short Batt When off
8	A	12		Output	N/A	N/A	Source	6000	Digital	Short Batt When off
9	A	13		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
10	A	14		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
31	A	15		Input	VTD (0-5640mV)	N/A	N/A	N/A	N/A	N/A
11	A	16		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
12	A	17		Output	N/A	N/A	Source	6000	Digital/PWM 30-500	Short Batt When off
32	A	18		Input	VTD (0-4657mV)	N/A	N/A	N/A	N/A	N/A
33	A	21		Input	VTD (0-36320mV)	N/A	N/A	N/A	N/A	N/A
1	A	22		Input	STB	N/A	N/A	N/A	N/A	N/A
2	A	23		Input	STB	N/A	N/A	N/A	N/A	N/A
3	A	24		Input	STB	N/A	N/A	N/A	N/A	N/A
4	A	25		Input	STB	N/A	N/A	N/A	N/A	N/A
5	A	26		Input	STB	N/A	N/A	N/A	N/A	N/A
6	A	27		Input	STB	N/A	N/A	N/A	N/A	N/A
7	A	28		Input	STB	N/A	N/A	N/A	N/A	N/A
8	A	29		Input	STB	N/A	N/A	N/A	N/A	N/A
9	A	31		Input	STB	N/A	N/A	N/A	N/A	N/A
10	A	32		Input	STB	N/A	N/A	N/A	N/A	N/A
11	A	33		Input	STB	N/A	N/A	N/A	N/A	N/A
12	A	34		Input	STB	N/A	N/A	N/A	N/A	N/A
13	A	35		Input	STB	N/A	N/A	N/A	N/A	N/A

<new project>
Data Items
Input
Output
Output
Variable
StateMachine
TimeCounter
EEPROM
CAN Receive
Constant
COM Bridge
Screens
HED Modules
Module1
Functions
Resources

Name	Val
General	
Current Report Rate (0-2550 mS)	255
Diagnostic Requirements	Sh
Groups	
Memo	

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passw
Warning	Project	The Project Level 2 Password has its original default value.	ProjectDefaultLevel2Passw
Warning	Project	The Project Level 3 Password has its original default value.	ProjectDefaultLevel3Passw
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActiv

Data Item Linking

Note: Orchestra® will prevent the user from assigning a data item to a pin that cannot support that data item, i.e. assigning a Switch to Battery (STB) defined Input to a Voltage to Digital (VTD) Input pin.

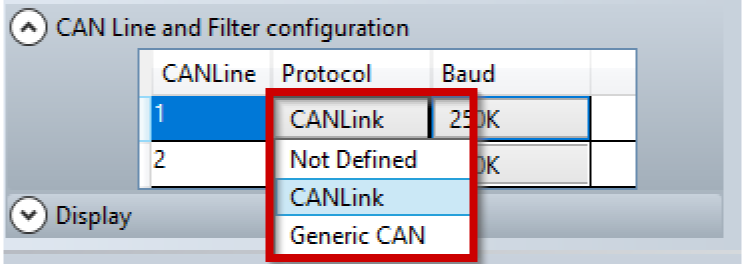
Each subsequent module added to the project is added as an I/O Module 278, and, like the Master Module, is listed within the HED® Modules folder.

Related:
Assign Data Items to Pins 274
Data Items 54

12.1.2.3.3 Assign Module Protocol

Once all of the Modules are selected, the user should determine and assign what kind of protocol the modules will use.

1. Go to the CAN Line and Filter Configuration pane.
2. Go to the Protocol field. The protocol button says “Not Defined”.
3. Click the protocol button.
4. Choose a protocol option. The protocol options are CANLink®, Generic CAN, or Not Defined.



Module Protocol Drop-Down Menu

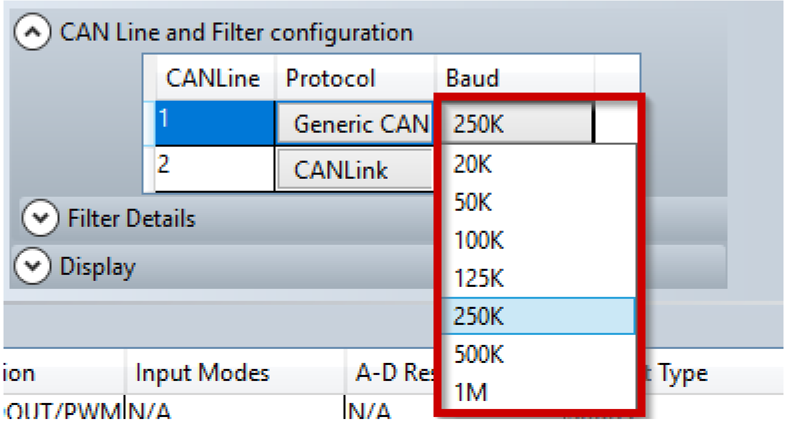
5. If there are more than one CAN lines available and only one is desired, the user can choose which line to use and then assign the other as Not Defined.

Related:
CAN Line and Filter Configuration 42
Change Baud Rate 276
Set Filter Details 276
Set Display Details 278

12.1.2.3.3.1 Change Baud Rate

1. Select a protocol to enable baud rate options.
2. Go to the Baud rate field and click the number.
3. Select a rate from the drop-down menu.
4. Change the baud rate by clicking on the number and selecting which rate to use.

If CANLink® or Not Defined is chosen as the protocol, the only baud rate available to use is 250Kbps. A wide range of rates can be chosen for Generic CAN from 20K, 50K, 100K, 125K, 250K, 500K, and 1M.



CANLine	Protocol	Baud
1	Generic CAN	250K
2	CANLink	

Module Baud Rate Drop-Down Menu

When Generic CAN is selected, another option appears to set up a filter for each channel, if so desired.

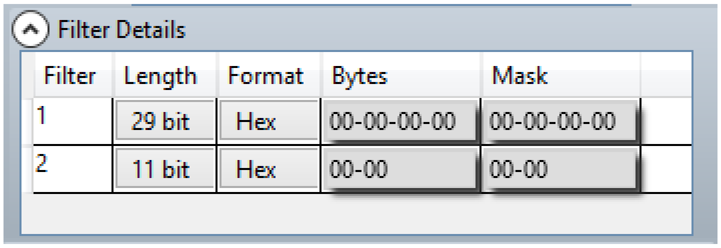
Related:

Assign Module Protocol 275

12.1.2.3.3.2 Set Filter Details

1. Add a master module to the project.
2. Go to the CAN Line and Filter Configuration pane.
3. Set the module protocol to Generic Can to enable the Filter Details pane.
4. Go to the Filter Details pane.
5. Go to the Length cell and choose 11 or 29 bit.
6. Go to the Format cell and choose Hex or Dec.
7. Go to the Bytes field and set the incoming CAN message identifier.
8. Go to the Mask field and set the incoming CAN message mask to match the Bytes ID.

9. Repeat as needed for additional CAN Lines.



Filter Details

Filter	Length	Format	Bytes	Mask
1	29 bit	Hex	00-00-00-00	00-00-00-00
2	11 bit	Hex	00-00	00-00

Module Filter Details

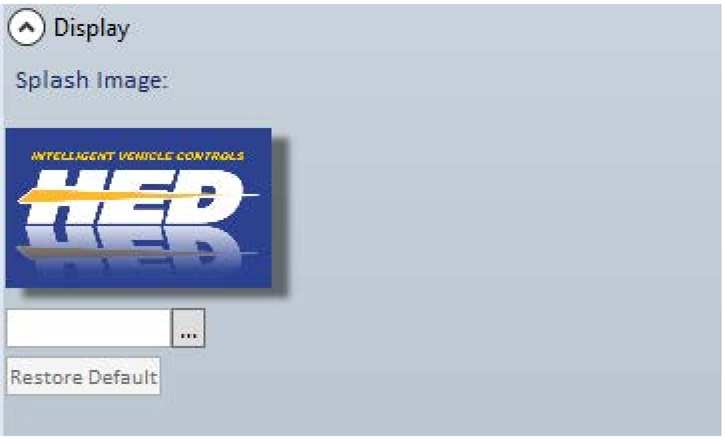
Related:

Assign Module Protocol 275

12.1.2.3.3.3 Set Display Details

Display

1. Add an image to the project folder.
2. Go to the Splash Image field.

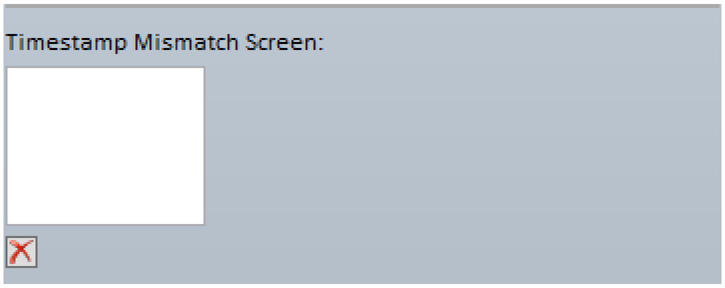


Splash Image Field

3. Browse for the image and click Open to assign it as the splash image.

Timestamp Mismatch Screen

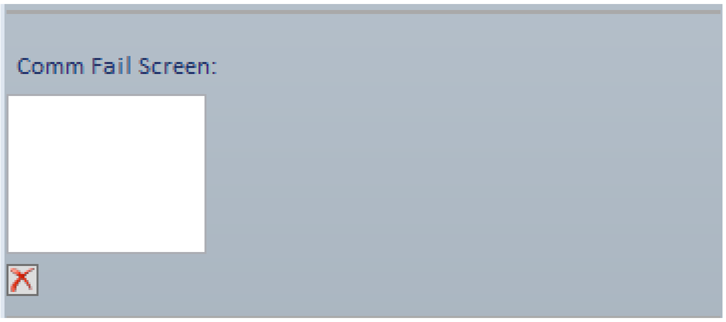
1. Create a screen 284 to display when there is a timestamp mismatch, and enter a descriptive name for the screen.
2. Go to the Project Explorer pane.
3. Go to the Screens folder 29 and click the screen to display when there is a timestamp mismatch.
4. Drag the screen to the Timestamp Mismatch Screen box.



Timestamp Mismatch Screen

Comm Fail Screen

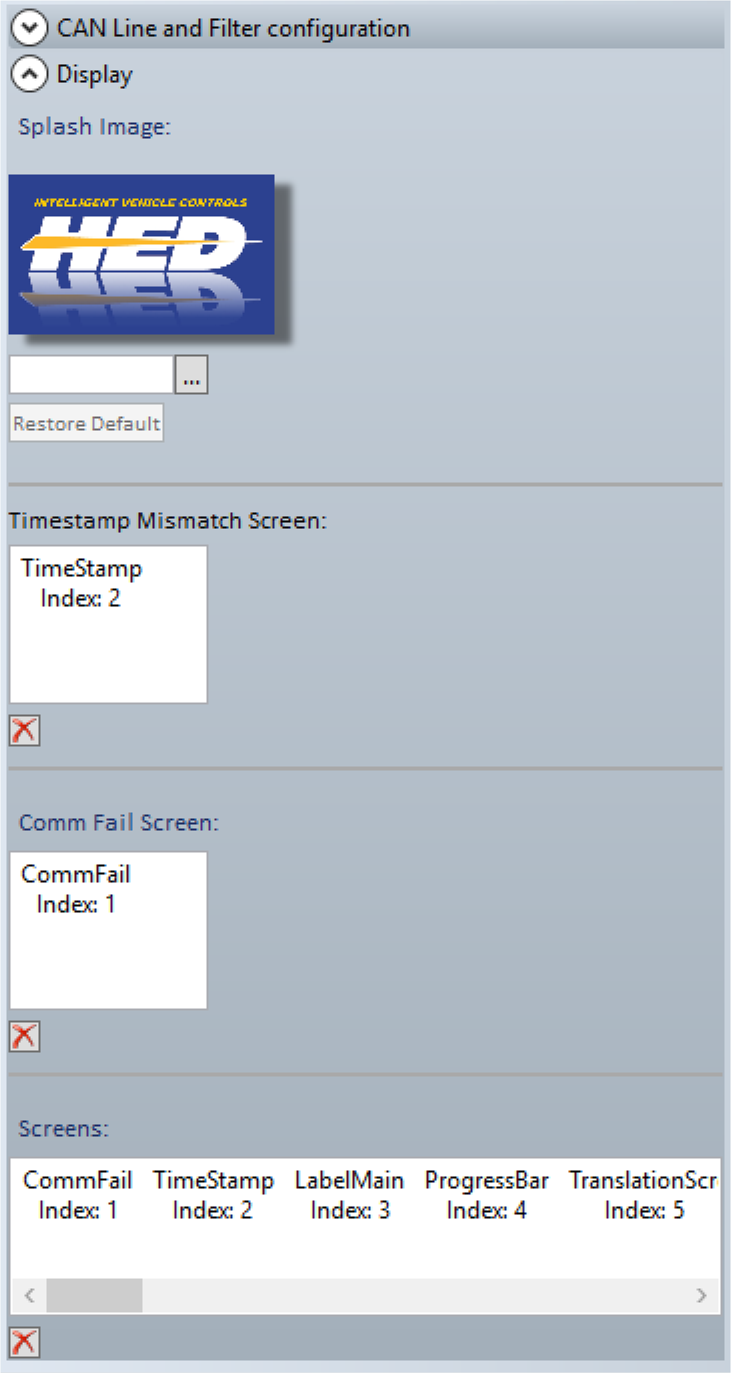
1. Create a screen 284 to display when there is a timestamp mismatch, and enter a descriptive name for the screen.
2. Go to the Project Explorer pane.
3. Go to the Screens folder 29 and click the screen to display when there is a timestamp mismatch.
4. Drag the screen to the Comm Fail Screen box.



Comm Fail Screen

Screens

1. Create all the screens 284 for your project.
2. Go to the Project Explorer pane.
3. Click and drag each screen to the Screens box.



CAN Line and Filter configuration
Display

Splash Image:

INTELLIGENT VEHICLE CONTROLS
HED
...
Restore Default

Timestamp Mismatch Screen:

TimeStamp
Index: 2

Comm Fail Screen:

CommFail
Index: 1

Screens:

CommFail	TimeStamp	LabelMain	ProgressBar	TranslationScr
Index: 1	Index: 2	Index: 3	Index: 4	Index: 5

Display Details

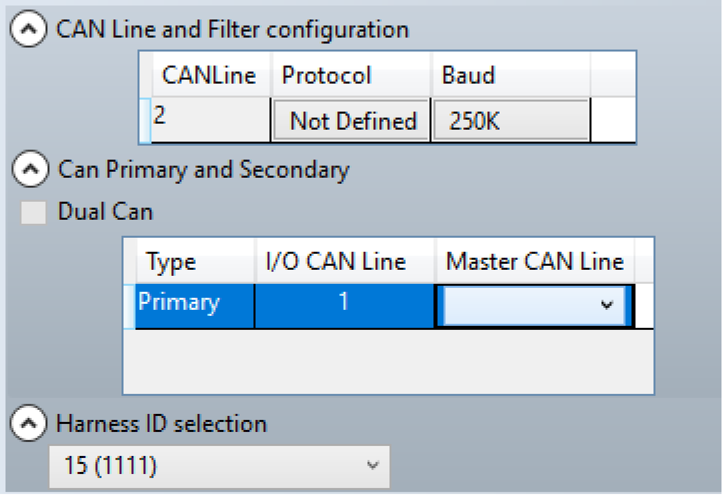
Related:

Display 46

Add Modules 272

12.1.2.3.4 Add I/O Modules

1. Double click a Module in the folder to bring up the Module Details in the Designer pane.
2. Set pin assignments 274.
3. Go to the Properties pane.
4. Adjust the loop time, name, and read/write security settings 51.
5. Select additional modules, as needed.
6. Go to the CAN Line and Filter menu 275.
7. Assign module protocol, baud rate, I/O Module CAN Lines, and I/O Module Harness Codes.



CAN Line and Filter configuration

CANLine	Protocol	Baud
2	Not Defined	250K

Can Primary and Secondary

□ Dual Can

Type	I/O CAN Line	Master CAN Line
Primary	1	

Harness ID selection

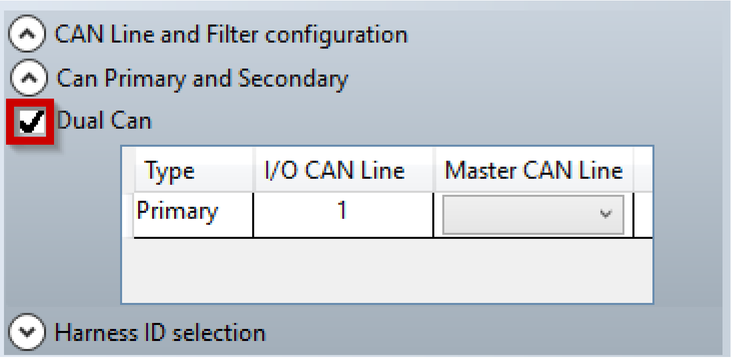
15 (1111)

IO Module Configuration

Related:
Add a Master Module 273

12.1.2.3.4.1 Set CAN Primary and Secondary

1. Add a master module 273 and set the CAN line protocol 275.
2. Add IO modules.
3. Go to the CAN Primary and Secondary pane.
4. To enable a secondary CAN line, click the Dual CAN check box.



CAN Line and Filter configuration

Can Primary and Secondary

☑ Dual Can

Type	I/O CAN Line	Master CAN Line
Primary	1	

Harness ID selection

Dual CAN Check Box

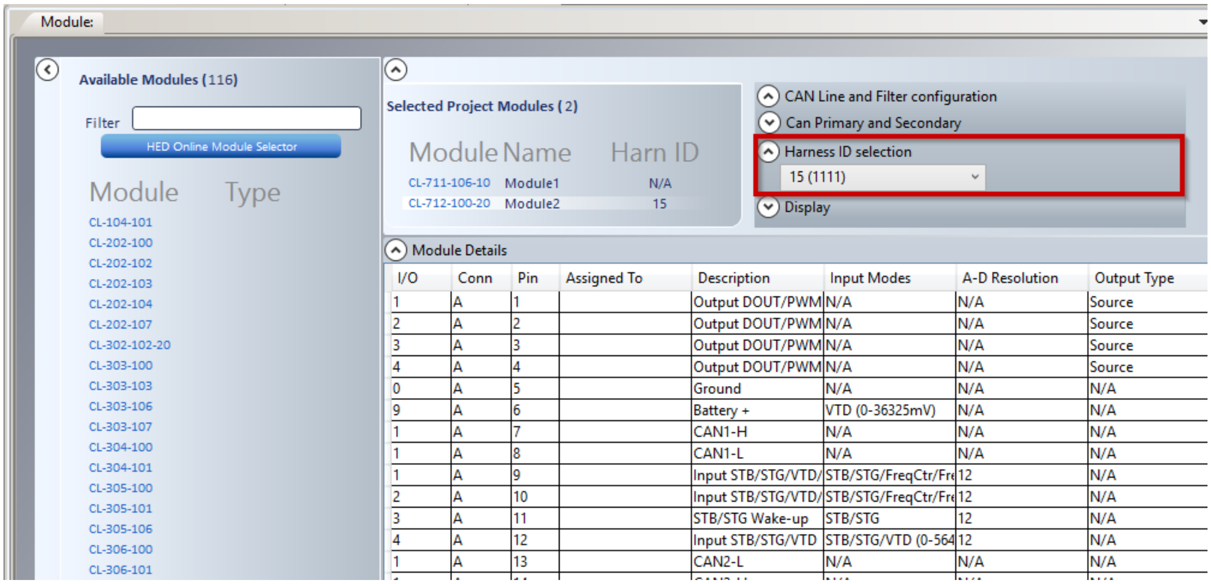
5. Go to the Master CAN Line field.
6. Choose a CAN line from the drop-down menu.

Related:

CAN Primary and Secondary 45

12.1.2.3.4.2 Select Harness ID

1. Open a project and add a master module.
2. Go to the CAN Line and Filter configuration pane and set the protocol to Generic CAN or CANLink®.
3. Set the master module baud rate.
4. Add an I/O display module.
5. Go to the Harness ID Selection pane.



Selected Project Modules (2)

Module	Name	Harn ID
CL-711-106-10	Module1	N/A
CL-712-100-20	Module2	15

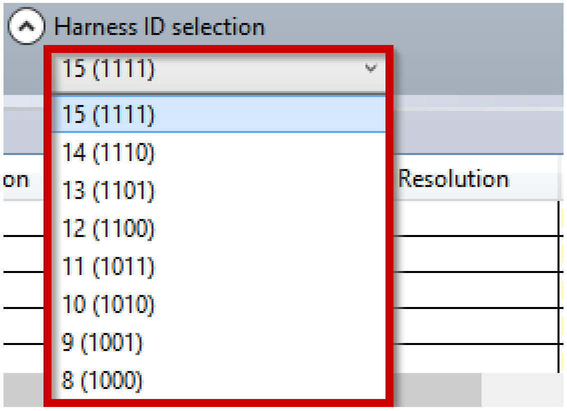
Harness ID selection: 15 (1111)

Module Details

I/O	Conn	Pin	Assigned To	Description	Input Modes	A-D Resolution	Output Type
1	A	1		Output DOUT/PWM	N/A	N/A	Source
2	A	2		Output DOUT/PWM	N/A	N/A	Source
3	A	3		Output DOUT/PWM	N/A	N/A	Source
4	A	4		Output DOUT/PWM	N/A	N/A	Source
0	A	5		Ground	N/A	N/A	N/A
9	A	6		Battery +	VTD (0-36325mV)	N/A	N/A
1	A	7		CAN1-H	N/A	N/A	N/A
1	A	8		CAN1-L	N/A	N/A	N/A
1	A	9		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A
2	A	10		Input STB/STG/VTD/	STB/STG/FreqCtr/Fre	12	N/A
3	A	11		STB/STG Wake-up	STB/STG	12	N/A
4	A	12		Input STB/STG/VTD	STB/STG/VTD (0-564	12	N/A
1	A	13		CAN2-L	N/A	N/A	N/A

Harness ID Selection Pane

6. Choose a harness ID from the drop-down menu.



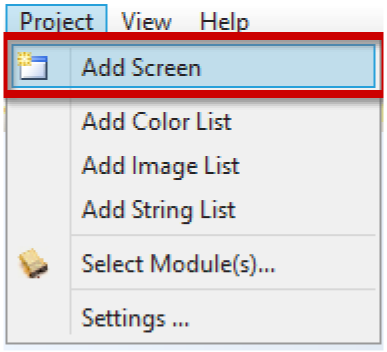
Module Harness ID Drop-Down

Related:

Add I/O Modules 281
Harness ID Selection 44

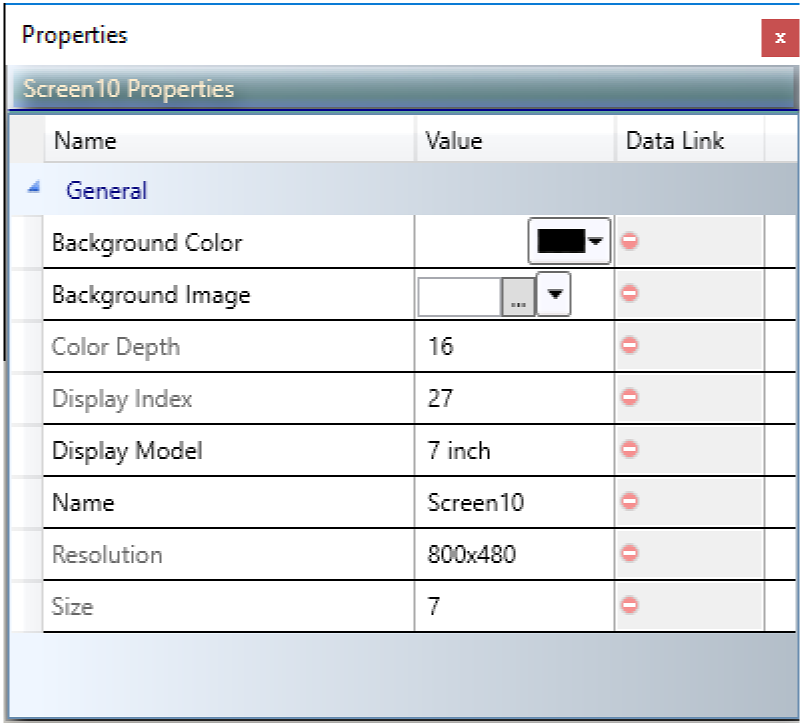
12.1.2.4 Add Screen

1. Click Project→ Add Screen.



Project Menu: Add Screen

2. Go to the Explorer Pane.
3. Click on the new screen in the folder structure.
4. Go to the Properties pane.
5. Set the Screen properties.



Screen10 Properties

Name	Value	Data Link
General		
Background Color	■	⊖
Background Image	...	⊖
Color Depth	16	⊖
Display Index	27	⊖
Display Model	7 inch	⊖
Name	Screen10	⊖
Resolution	800x480	⊖
Size	7	⊖

Screen Properties

6. Go to the Screen Designer pane.
7. Go to the Widgets Panel.
8. Add Widgets 322 from the Widgets Panel.

9. Click the widget and drag it to the desired location on the screen.

Related:

Add Screen 22
Add Widgets 322

12.2 Edit Properties Using the Properties Pane

Related:
Edit Project Properties 286
Edit Module Properties 290
Properties Pane 33

12.2.1 Edit Project Properties

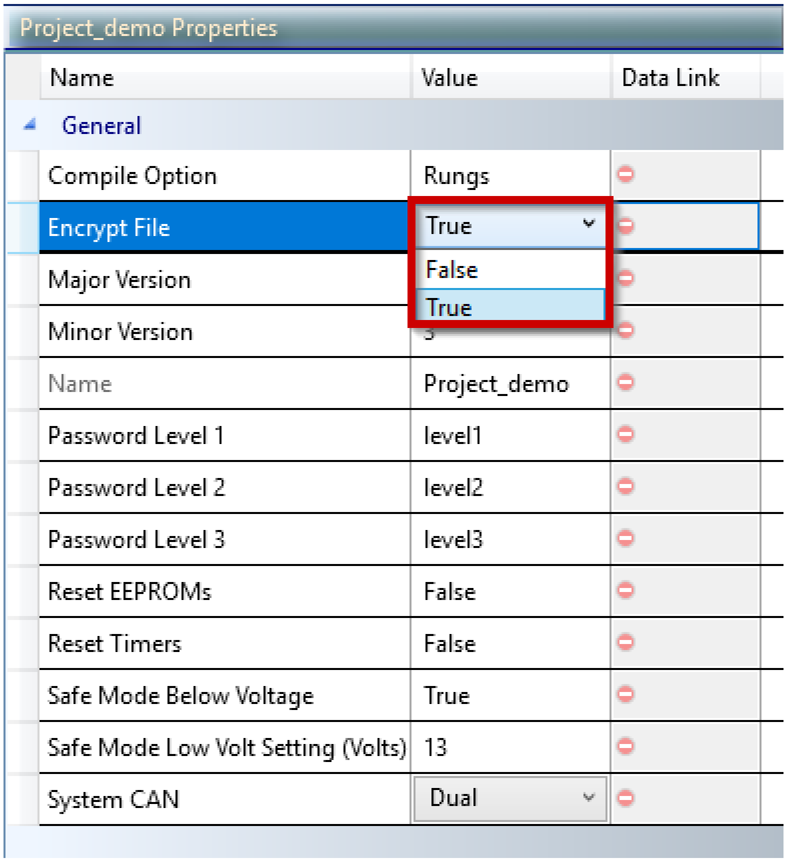
1. Go to the Explorer pane 26 and click the Project folder.

2. Go to the Properties pane 33.

3.Click the Compile Option 249 value field, and choose Rungs, Presto™ with Rungs, or Presto™.

5. Click the Encrypt File 251 value field, and choose True or False.

Note: HED® recommends setting Encrypt File to True.



Project_demo Properties

Name	Value	Data Link
General		
Compile Option	Rungs	
Encrypt File	True False True	
Major Version		
Minor Version	3	
Name	Project_demo	
Password Level 1	level1	
Password Level 2	level2	
Password Level 3	level3	
Reset EEPROMs	False	
Reset Timers	False	
Safe Mode Below Voltage	True	
Safe Mode Low Volt Setting (Volts)	13	
System CAN	Dual	

Set File Encryption

7. Click the Major Version 254 value field and enter the version number.

8. Click the Minor Version 255 value field and enter the version number.

9. Click the Password Level 1 257 value field and enter the password used in Conductor™ for Level 1 privileges.

10.Click the Password Level 2 257 value field and enter the password used in Conductor™ for Level 2 privileges.

11. Click the Password Level 3 257 value field and enter the password used in Conductor™ for Level 3 privileges.

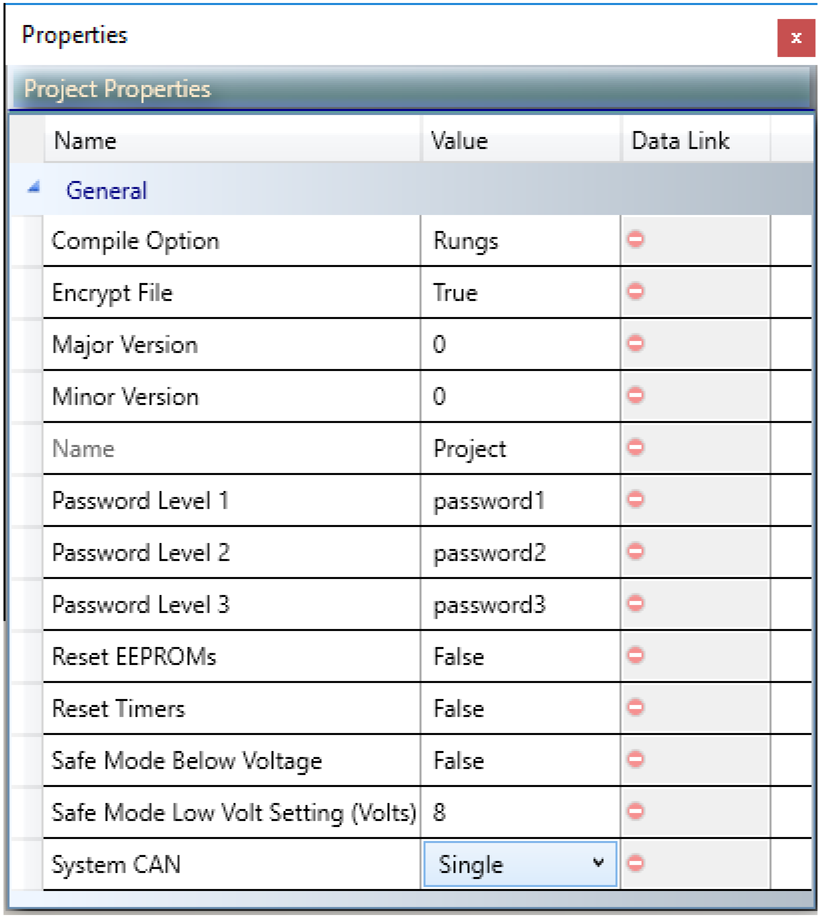
12. Click the Reset EEPROMS 257 value field and choose True or False.

13.Click the Reset Timers 257 value field and choose True or False.

14. Click the Safe Mode Below Voltage 257 value field and choose True or False.

15. Click the Safe Mode Low Volt Setting 257 value field and enter a number.

16. Click the System CAN 259 value field and choose either Single or Dual.



Properties

Project Properties

Name	Value	Data Link
General		
Compile Option	Rungs	⊖
Encrypt File	True	⊖
Major Version	0	⊖
Minor Version	0	⊖
Name	Project	⊖
Password Level 1	password1	⊖
Password Level 2	password2	⊖
Password Level 3	password3	⊖
Reset EEPROMs	False	⊖
Reset Timers	False	⊖
Safe Mode Below Voltage	False	⊖
Safe Mode Low Volt Setting (Volts)	8	⊖
System CAN	Single	⊖

Project Properties Pane

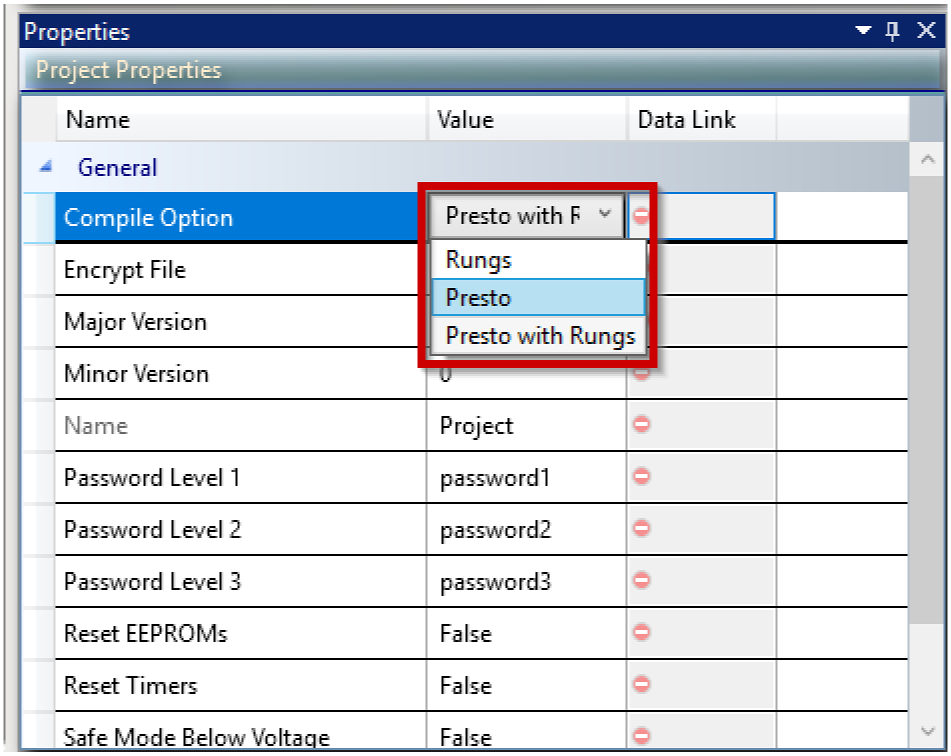
Related:

Set Compile Option 289

Project Properties 34

12.2.1.1 Set Compile Option

1. Go to the Value cell of the Compile Option row.
2. Click the Value cell to open the Compile Option drop-down.
3. Choose Rungs, Presto™, or Presto™ with Rungs as the compile option.
   Rungs: Rungs is the graphic layout of the logic inside of Orchestra®
   Presto™: Presto™ creates a code warrior project that the user writes their logic in C
   Presto™ with Rungs:



Name	Value	Data Link
Compile Option	Presto with F	
Encrypt File		
Major Version		
Minor Version	0	
Name	Project	
Password Level 1	password1	
Password Level 2	password2	
Password Level 3	password3	
Reset EEPROMs	False	
Reset Timers	False	
Safe Mode Below Voltage	False	

Project Compile Options Menu

Related:

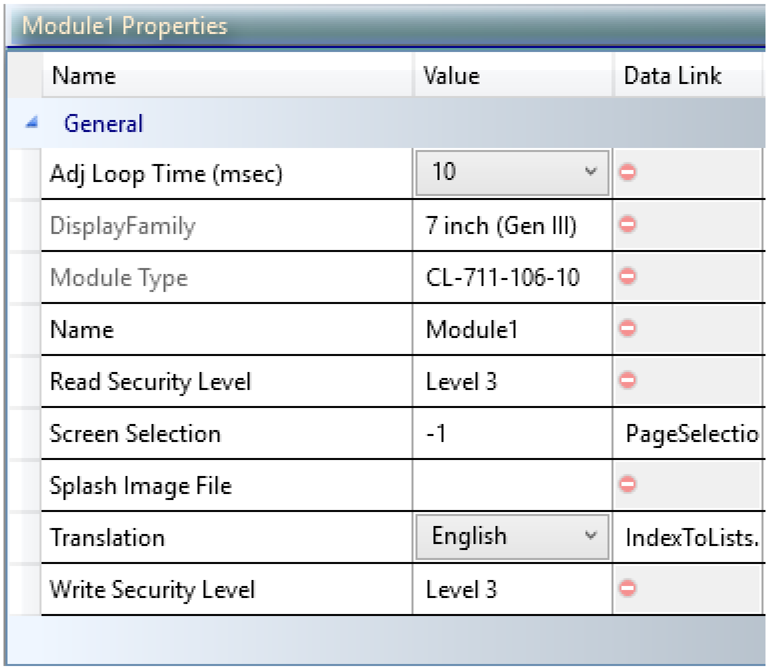
Compile Option 35

Edit Project Properties 286

12.2.2 Edit Module Properties

Each module in the project has different properties, depending on if the module is a display, master, master-display, or I/O module.

Master Module

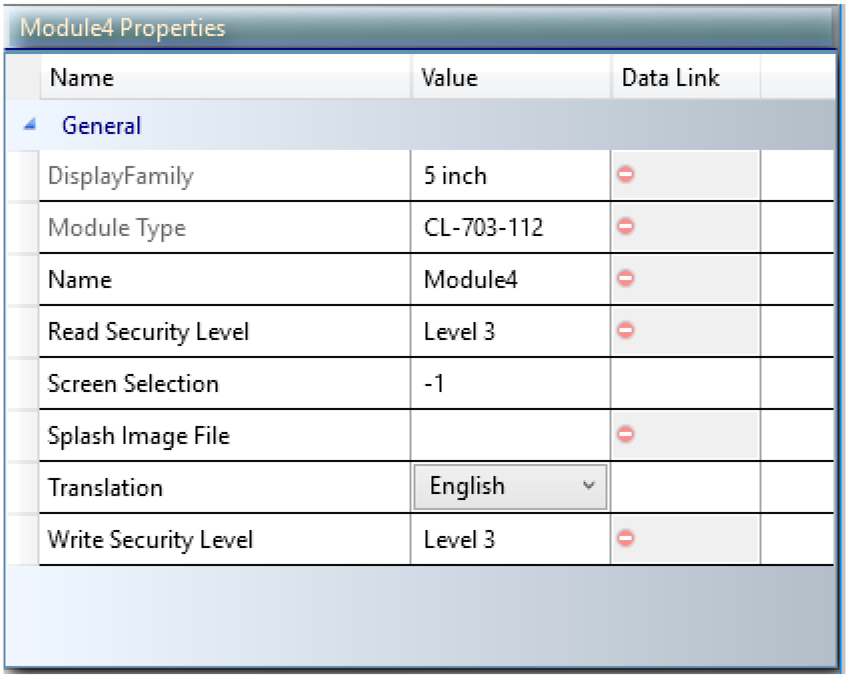


Module1 Properties

Name	Value	Data Link
General		
Adj Loop Time (msec)	10	⊖
DisplayFamily	7 inch (Gen III)	⊖
Module Type	CL-711-106-10	⊖
Name	Module1	⊖
Read Security Level	Level 3	⊖
Screen Selection	-1	PageSelectio
Splash Image File		⊖
Translation	English	IndexToLists.
Write Security Level	Level 3	⊖

Master Module Properties

Display Module

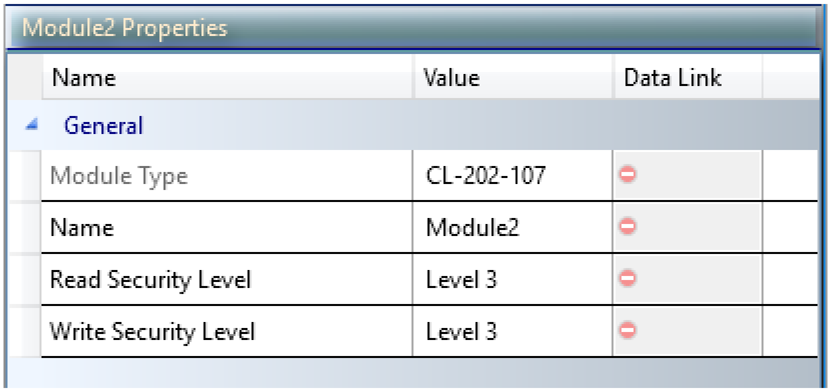


Module4 Properties

Name	Value	Data Link
General		
DisplayFamily	5 inch	⊖
Module Type	CL-703-112	⊖
Name	Module4	⊖
Read Security Level	Level 3	⊖
Screen Selection	-1	
Splash Image File		⊖
Translation	English	
Write Security Level	Level 3	⊖

Display Module

I/O Module



Module2 Properties

Name	Value	Data Link
General		
Module Type	CL-202-107	⊖
Name	Module2	⊖
Read Security Level	Level 3	⊖
Write Security Level	Level 3	⊖

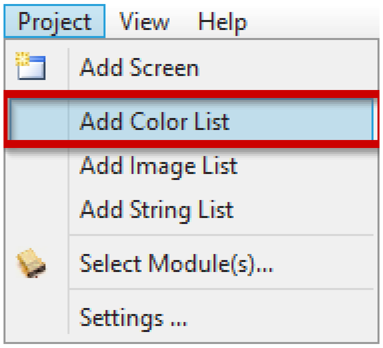
I/O Module Properties

Related:

Edit Properties Using the Properties Pane 286
Module Properties 48

12.3 Add Color List

1. Click Project→Add Color List.



Project Menu-Add Color List

2. Go to the Explorer pane.
3. Go to the Resources folder.
4. Go to the Color List subfolder.
5. Click the new color list.
6. Go to the Properties pane.
7. Set the Properties for the color list.

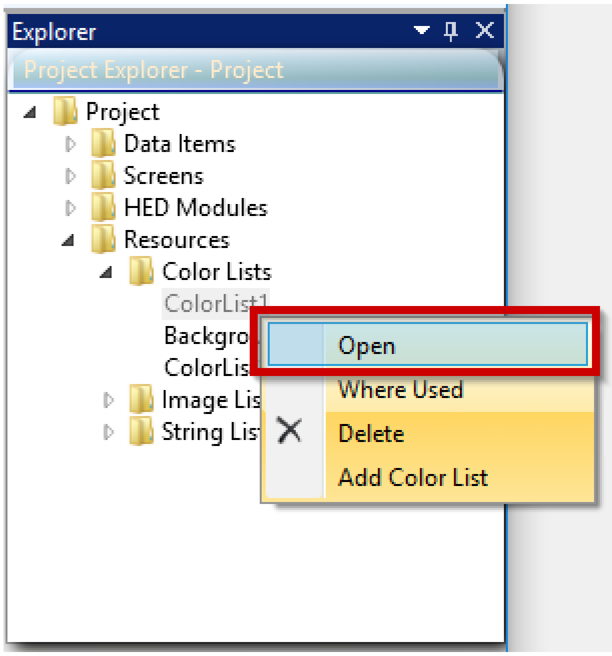
Related:

Add Color List 23

Resources Folder 31

12.3.1 Open

1. Right click on the color list.



Open Color List

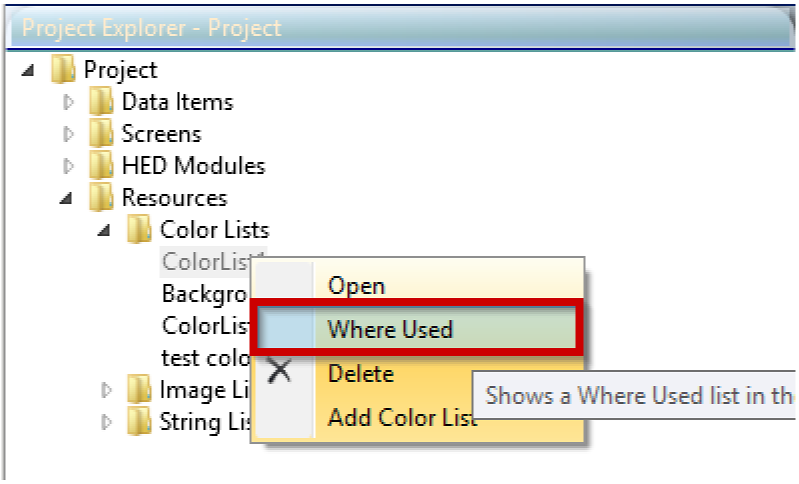
2. Click Open.
3. Go to the ColorList:[name] tab in the Arranger™ pane.
4. Review the ColorList details.

Related:

Add Color List 292
Where Used 294
Add Color to Color List 295

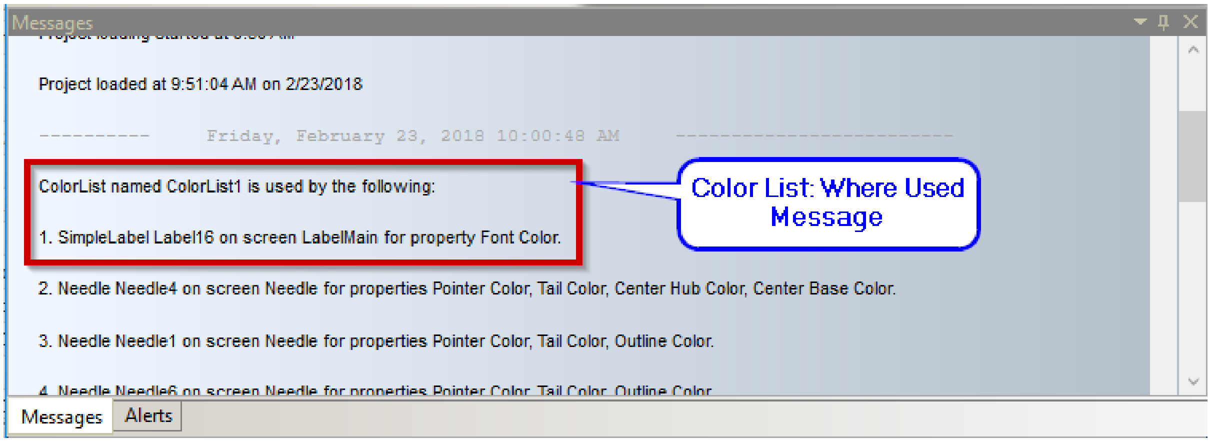
12.3.2 Where Used

1. Right click on the color list.
2. Click Where Used.



Color List-Where Used

3. Go to the Alerts/Messages pane.
4. Click the Messages tab.
5. Scroll to the end of the messages list.
6. Find the Where Used Message.
   a. ColorList named ColorList2 is used by the following:
   `SimpleLabel CrouchingLabel on screen LabelMain for property Background Color.`



```
Project loaded at 9:51:04 AM on 2/23/2018

----------     Friday, February 23, 2018 10:00:48 AM     -------------------------

ColorList named ColorList1 is used by the following:

1. SimpleLabel Label16 on screen LabelMain for property Font Color.

2. Needle Needle4 on screen Needle for properties Pointer Color, Tail Color, Center Hub Color, Center Base Color.

3. Needle Needle1 on screen Needle for properties Pointer Color, Tail Color, Outline Color.

4. Needle Needle6 on screen Needle for properties Pointer Color, Tail Color, Outline Color.
```

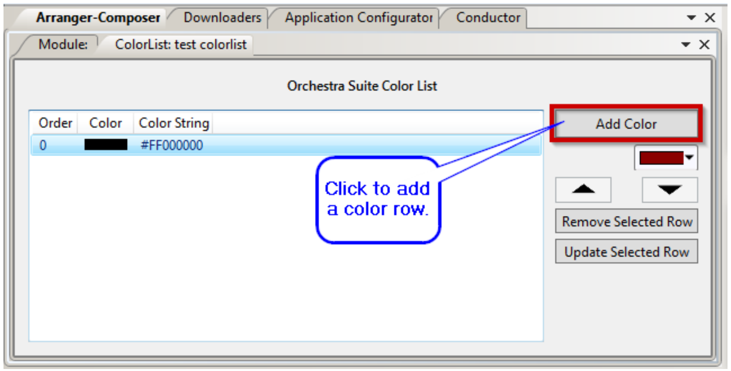
Where Used Message

Related:

Add Color List 292
Open 293
Add Color to Color List 295

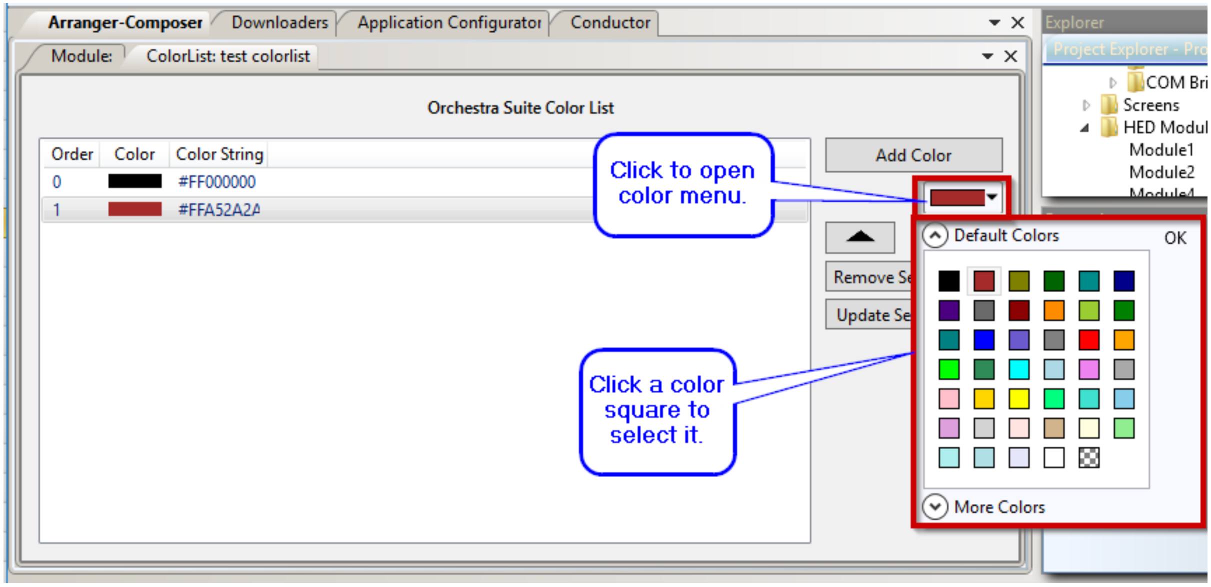
12.3.3 Add Color to Color List

1. Click the ColorList: [name] tab.
2. Click the Add Color button.



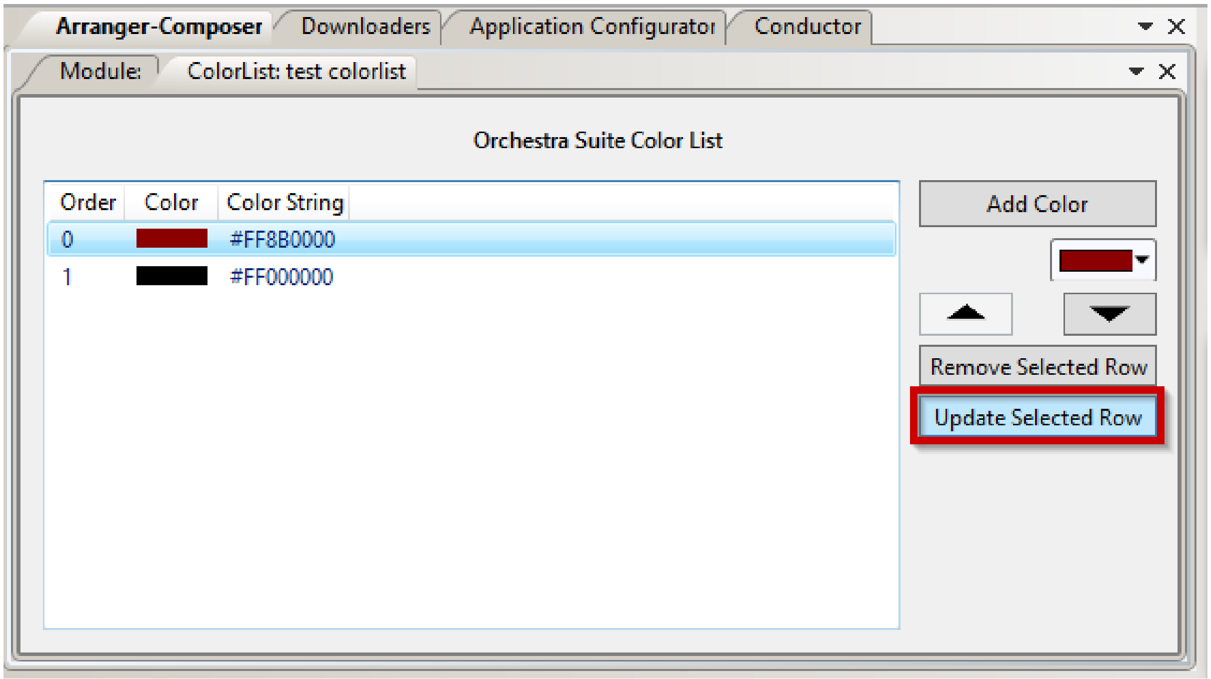
Add Color Row

3. Click the added row of the list.
4. Click the color drop-down menu.
5. Choose a color from the palette.



Choose a Color from the Palette

6. Adjust the order of the color using the Up and Down arrows.
7. Set the changes to the color list by clicking the Update Selected Row button.



Order	Color	Color String
0	🟥	#FF8B0000
1	⬛	#FF000000

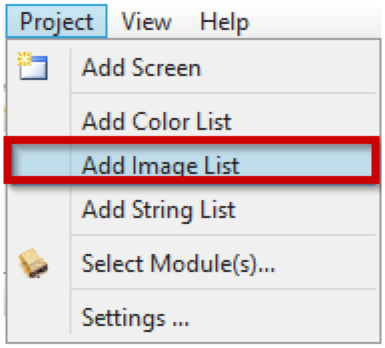
Update Selected Row

Related:

[Add Color List] 292
[Open] 293
[Where Used] 294

12.4 Add Image List

1. Click Project→Add Image List.



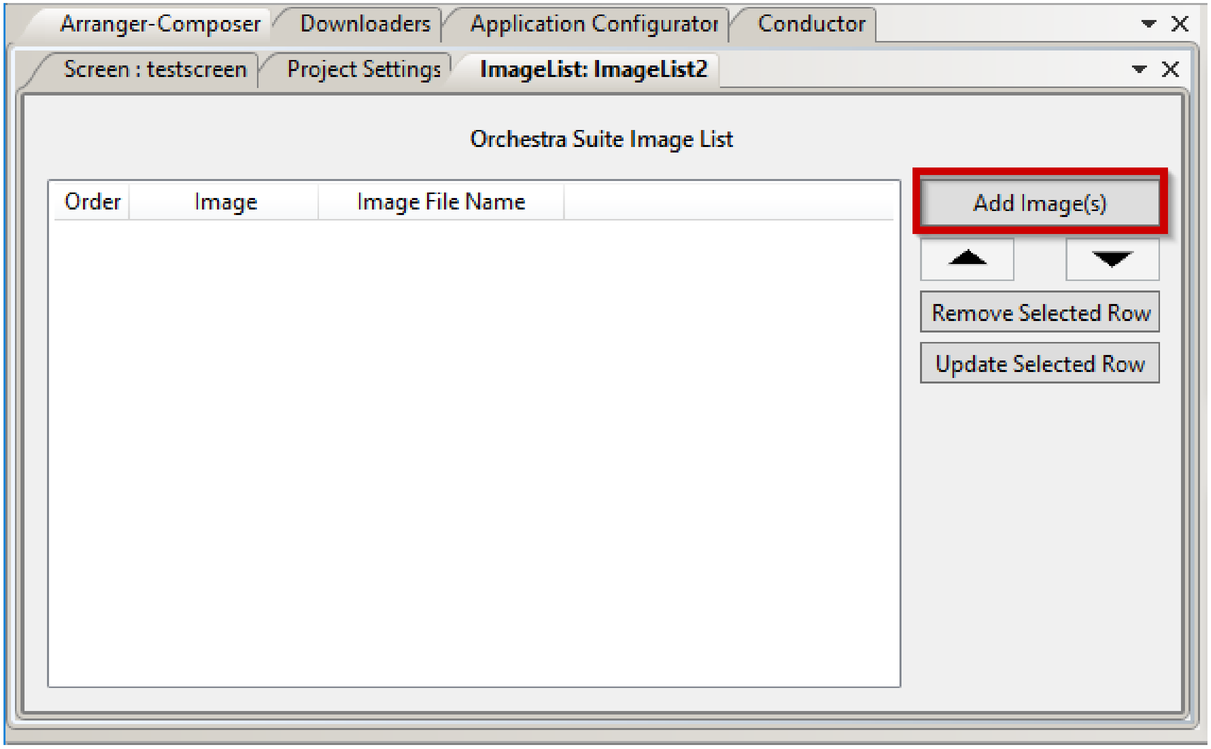
Project-Add Image List

2. Go to the Explorer Pane.
3. Go to the Resources folder.
4. Go to the Image List subfolder.
5. Click the new image list.
6. Go to the Properties pane.
7. Set the Properties for the image list.

Related:
Add Image List 23
Add Image to Image List 298

12.4.1 Add Image to Image List

1. Click the ImageList: [name] tab.
2. Click the Add Image button.



Add Image

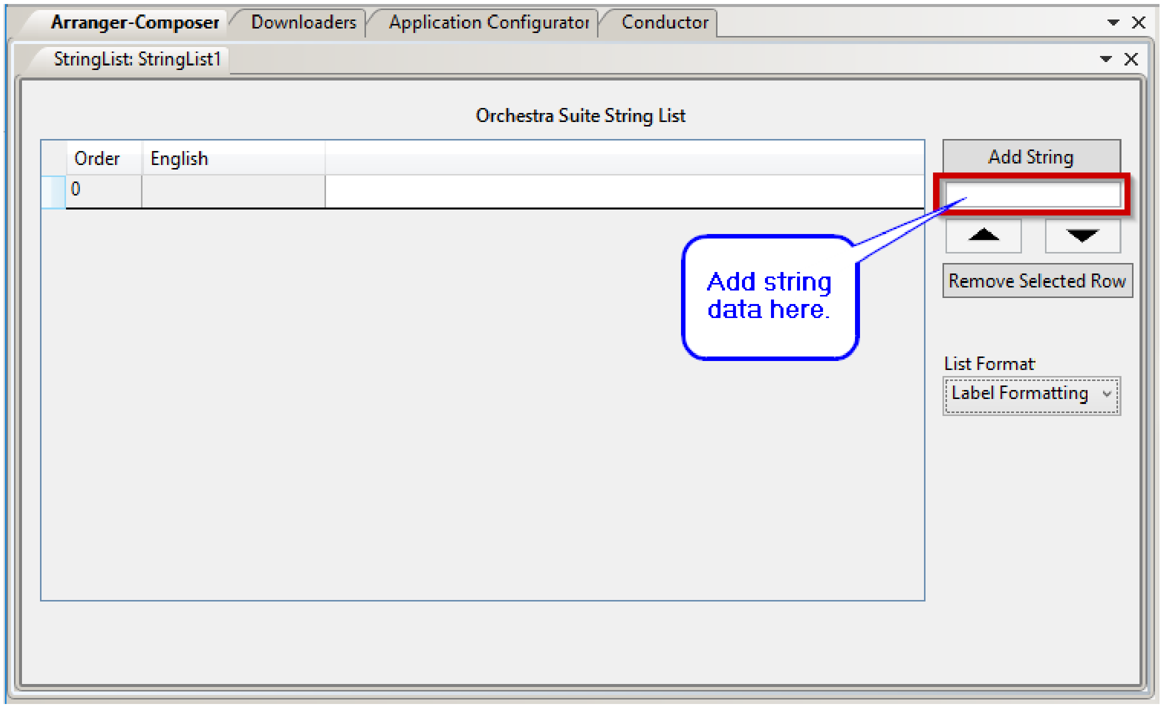
3. Browse for and select an image.
4. Adjust the order of the image using the Up and Down arrows.
5. Set the changes to the image list by clicking the Update Selected Row button.

Related:
Add Image List 297

12.5 Add String List

The String List Designer is used to edit a String List Resource, allowing you to add and remove strings and their translations, and reorder strings in the list.

1. Open a project or begin a new project.
2. Add a master module.
3. Go to the Project Explorer drop-down Resources folder.
4. Go to the String Lists subfolder.
5. Add a new string list.
6. Go to the List format field and choose a format for the string 300.
7. Go to the Add String field.



Add String Field

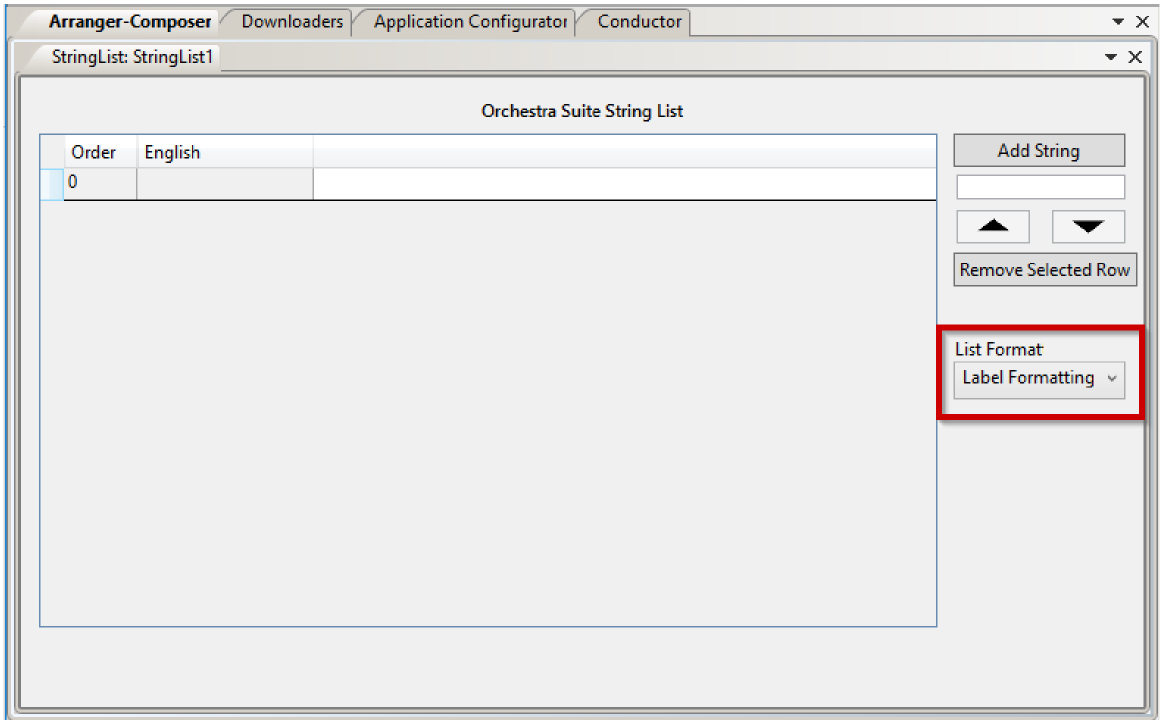
8. Enter the string data. This will be the English word to be translated.
9. Click the Add String button to add the data to the English field.
10. Click an individual cell in the String List table to change the data.
11. Adjust the order of the row using the Up and Down arrows.

Related:

[Change List Format](300)
[Add String List](24)
[Project Settings](21)

12.5.1 Change List Format

1. Click the List Format drop-down menu.



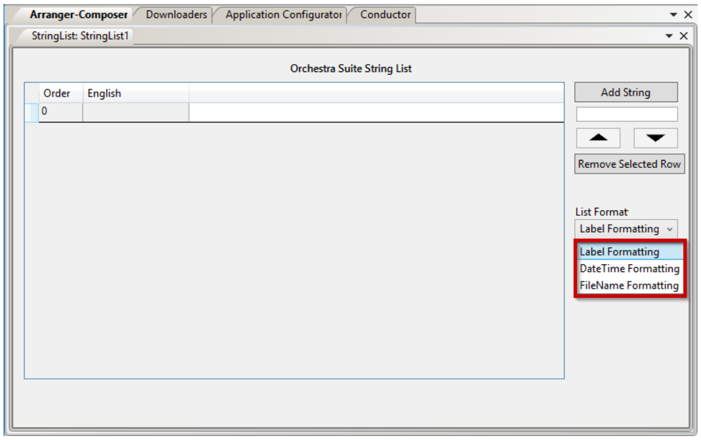
List Format

2. Choose a list format:

    a. Label Formatting

    b. DateTime Formatting

    c. FileName Formatting



List Format: Label Formatting

Order	English
0	

List Format Types

Related:

[Add String List](299)
[Project Settings](21)

12.6 Conductor Application Notes

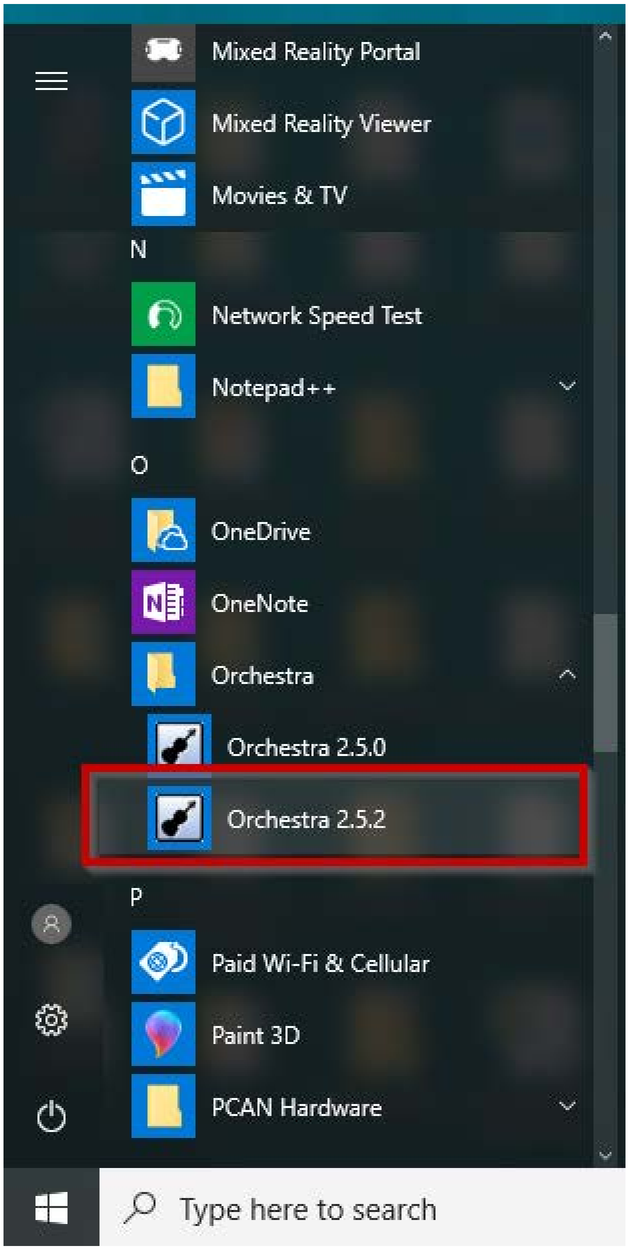
12.6.1 Conductor Quick Start

GENERAL INFORMATION

This guide outlines the steps to launch the Conductor™. This assumes that the master module has been programmed with the appropriate firmware and that all connections to the module have been made and that the module is powered. A dongle or temporary license file is also required for the Conductor™ to operate.

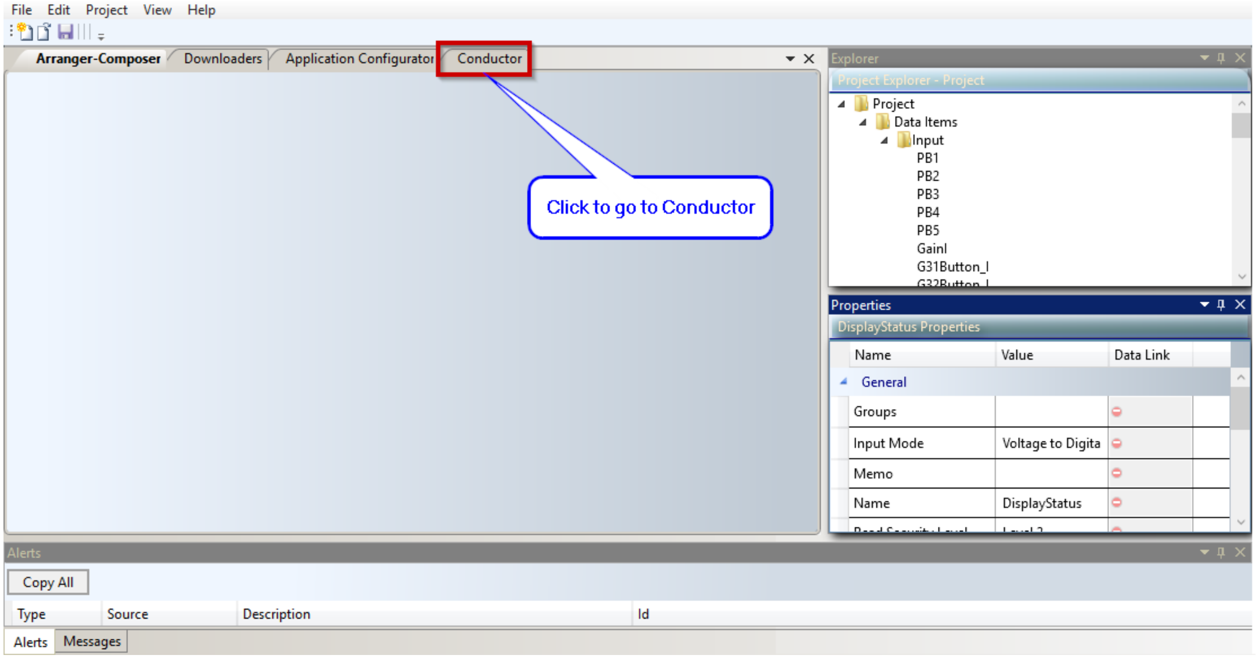
GETTING STARTED

1. Open Orchestra®.



Open Orchestra® from Windows Start Menu

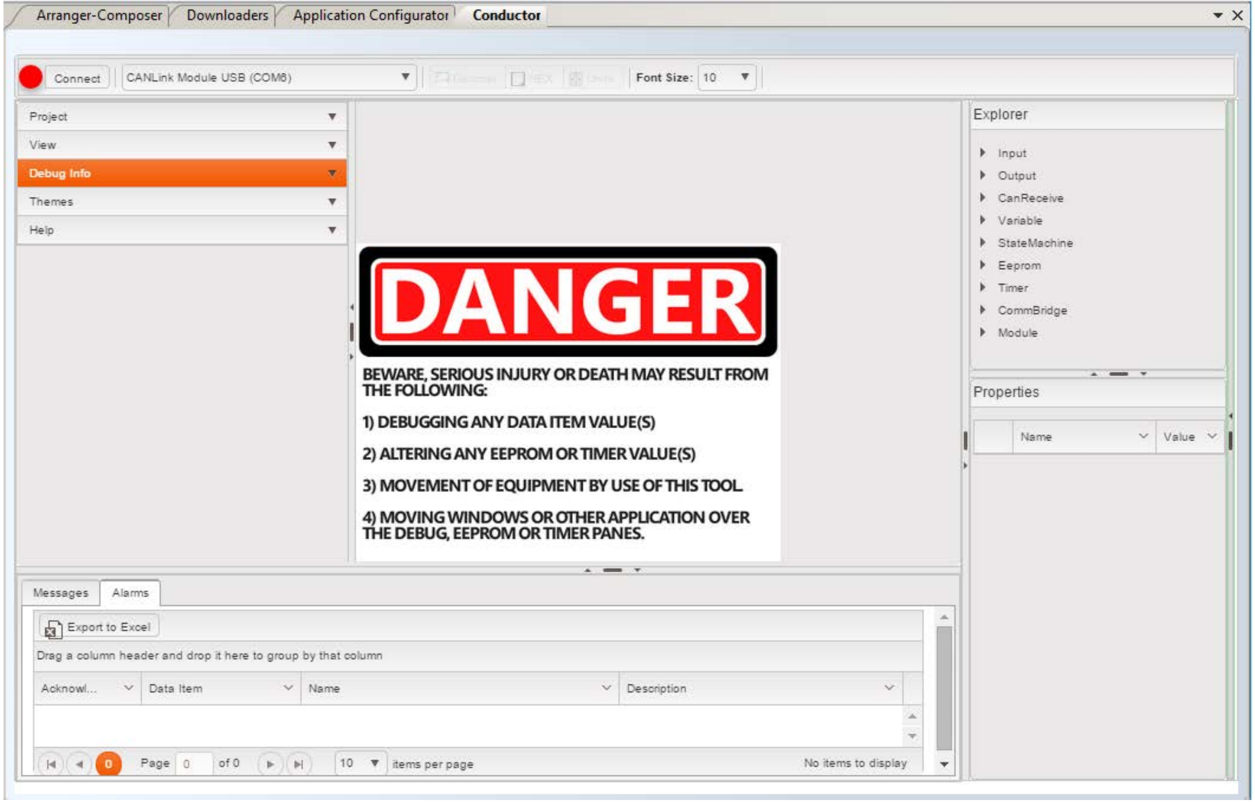
2.Click the Conductor™ tab.



Click to go to Conductor

Conductor™ Tab

3. The Conductor™ main window opens.



DANGER

BEWARE, SERIOUS INJURY OR DEATH MAY RESULT FROM THE FOLLOWING:

1) DEBUGGING ANY DATA ITEM VALUE(S)

2) ALTERING ANY EEPROM OR TIMER VALUE(S)

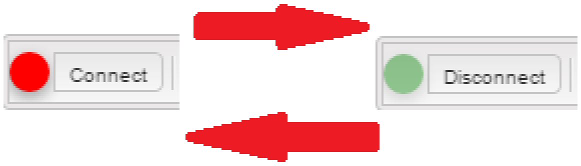
3) MOVEMENT OF EQUIPMENT BY USE OF THIS TOOL.

4) MOVING WINDOWS OR OTHER APPLICATION OVER THE DEBUG, EEPROM OR TIMER PANES.

Drag a column header and drop it here to group by that column

Conductor™ Landing Screen

4.Click the Connect/Disconnect button.



Connect/Disconnect Button

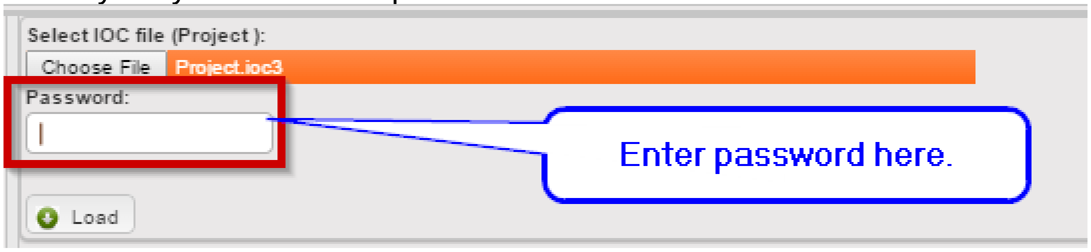
Note: In order to run Conductor™, an equipment-specific file is required. This file will be an .IOc_XX_YY (where XX and YY are used for designating revision) type file and should be obtained from the OEM. If an invalid file is selected, no connection will be made and a prompt will warn of invalid file selection.

5.Select the .IOc_XX_YY file supplied by the OEM and click Open.

Note: If a dongle is not connected, a No Dongle Found message will appear.

6.Connect a dongle to an available USB port and click OK. If a second attempt is made to connect without a dongle, the Conductor™ will open on the next try in *** DEMO MODE *** with limited capabilities.

7.After the proper .IOc file has been selected, enter the password that was supplied to you by the OEM. The password is case-sensitive.


Select IOC file (Project):

Choose File Project.ioc3

Password:

Conductor Password

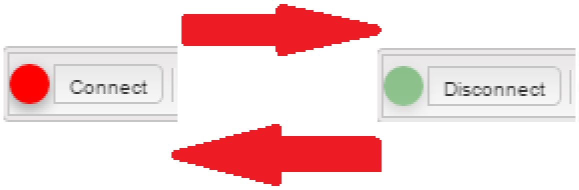
*Note: If an incorrect password is entered three consecutive times, Conductor™ will default to a *** READ ONLY MODE *** where you can observe the status of inputs and outputs etc., but you cannot debug them.*

8. Once a valid password is entered and a proper connection is made between the computer and the OEM product, Conductor™ will be connected. A proper connection can be verified by looking at the Connect/Disconnect button on Conductor's main toolbar.

9. Conductor™ is ready for use at this time.

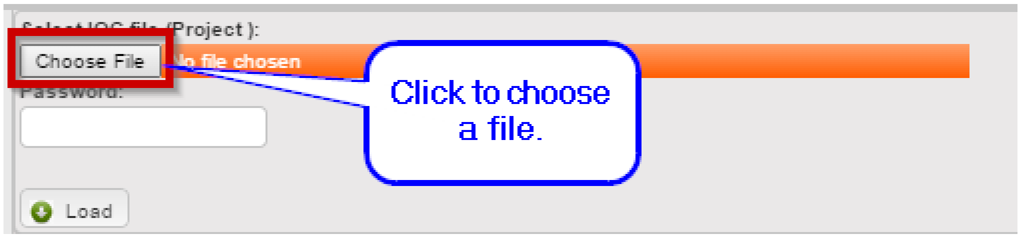
12.6.2 Using Conductor

1.Click the Connect/Disconnect button.



Connect/Disconnect Button

2.Click the Choose File button.



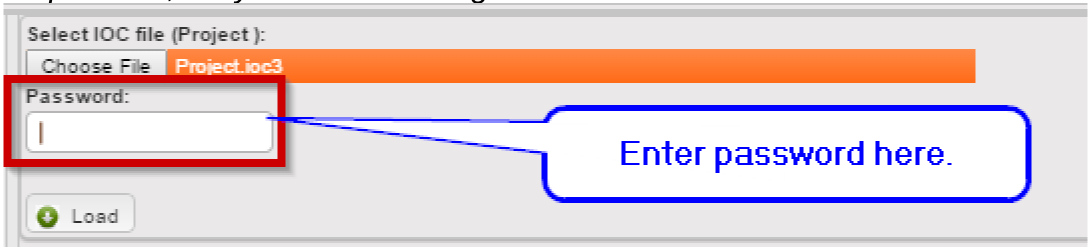
Choose File Button

3. A file browser window opens.

4. Select the .IOc_XX_YY file supplied by the OEM and click Open.

5.Enter the password that was supplied to you by the OEM. The password is case-sensitive.

*Note: If an incorrect password is entered three consecutive times, the Conductor™ will default to a *** READ ONLY MODE *** where you can observe the status of inputs and outputs etc., but you cannot debug them.*



Password Field

6.Once a valid password is entered or Cancel is clicked, and a proper connection is made between your computer and the OEM product, the Conductor™ will be connected.

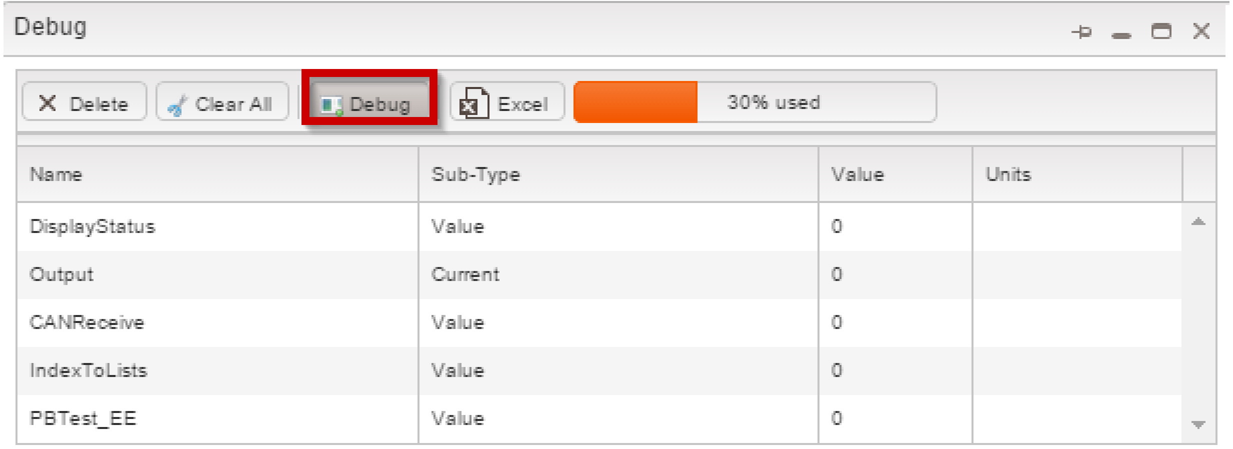
7. Conductor™ is ready for use.

Related:
Conductor Quick Start 302

12.6.3 Use Debug Mode

Once all desired data items have been selected and added to the Debug window, the Debug session can begin.

1. Click the **Debug** button to start the Debug session.



Name	Sub-Type	Value	Units
DisplayStatus	Value	0	
Output	Current	0	
CANReceive	Value	0	
IndexToLists	Value	0	
PBTest_EE	Value	0	

Debug Button

3. The user can now make careful adjustments to any data item in the Debug window by selecting a data item and using the controls in the Value column to make the desired adjustment. Pay close attention to how the data item adjustment impacts control of the OEM product. If the product ever begins to act in an unpredictable or unexpected manner, IMMEDIATELY press the keyboard spacebar to enter the Emergency Stop (E-STOP) - Safe Mode.

4. After all diagnostic troubleshooting has been completed and all data items have been adjusted, click the Debug button to end the Debug session.

WARNING

MASTER MODULE OVERRIDE HAZARD

Whenever Debug is activated, you are assuming direct control of selected Data Item values and potentially bypassing safety interlocks and normal operation. Entering a wrong value may cause unpredictable OEM product behavior. Death or serious injury can result.

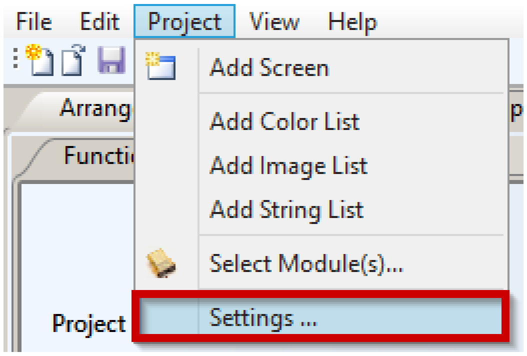
Remember that at any time that the Conductor™ software is connected to an OEM product, the Emergency Stop (E-STOP) - Safe Mode can be activated one of two ways.

1. Depressing the computer keyboard spacebar

2. Disconnecting the communication cable between the computer and the product

12.7 Edit Project Settings

To access the Project Settings screen, select Settings from the Project menu.



Project-Settings

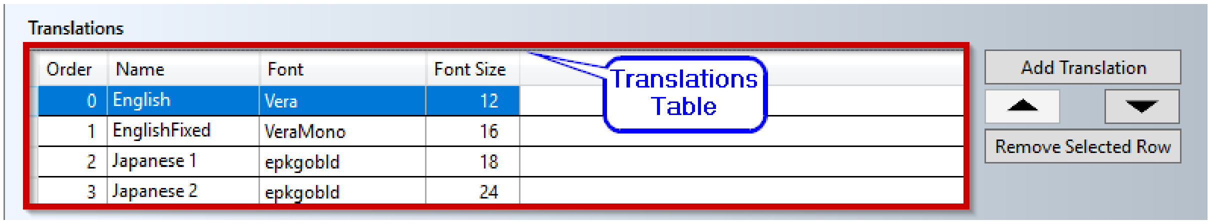
Related:

[Add Translation] 308
[Add Font] 312
[Add Group] 314
[Project Settings] 21

12.7.1 Add Translation

1. Click the Add Translation button to add a row to the Translations table.

2. Go to the Translation table.

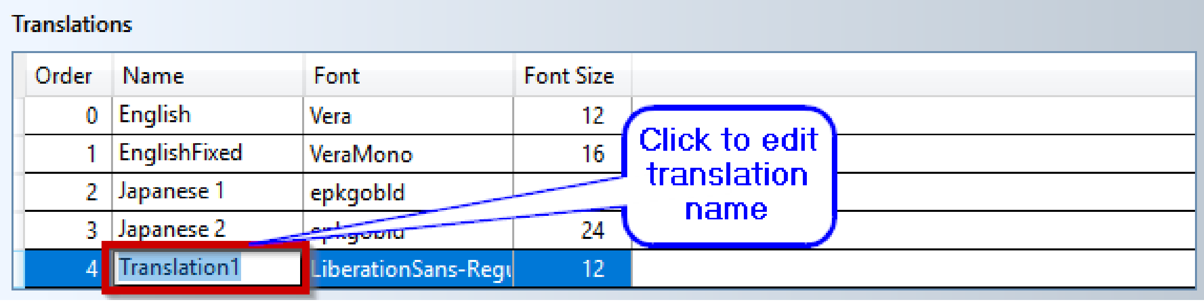


Order	Name	Font	Font Size
0	English	Vera	12
1	EnglishFixed	VeraMono	16
2	Japanese 1	epkgobld	18
3	Japanese 2	epkgobld	24

Translation Table

3. Go to the last row.

4. Click the Name cell.



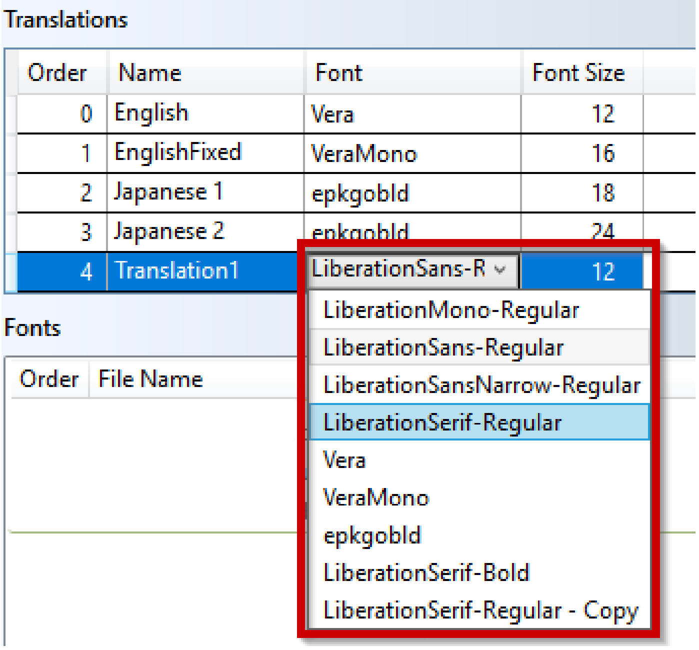
Order	Name	Font	Font Size
0	English	Vera	12
1	EnglishFixed	VeraMono	16
2	Japanese 1	epkgobld	
3	Japanese 2	epkgobld	24
4	Translation1	LiberationSans-Regu	12

Name Cell

5. Edit the name.

6. Click the Font cell.

7. Choose a font from the drop-down menu.



Translations

Order	Name	Font	Font Size
0	English	Vera	12
1	EnglishFixed	VeraMono	16
2	Japanese 1	epkgobld	18
3	Japanese 2	epkgobld	24
4	Translation1	LiberationSans-R	12

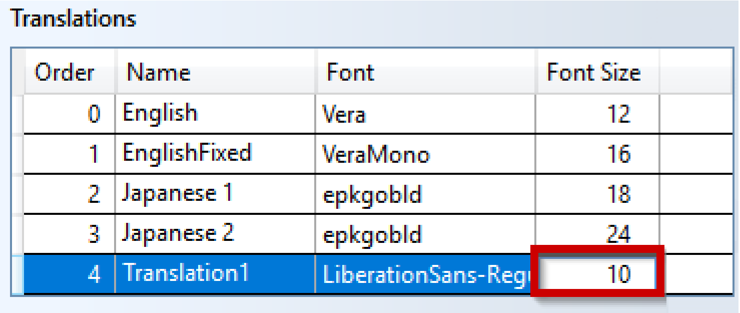
Fonts

Order	File Name

Font Drop-Down Menu

8. Go to the Size cell.

9. Enter the size of the font.



Translations

Order	Name	Font	Font Size
0	English	Vera	12
1	EnglishFixed	VeraMono	16
2	Japanese 1	epkgobld	18
3	Japanese 2	epkgobld	24
4	Translation1	LiberationSans-Regu	10

Font Size Cell

10. Click the Up and Down buttons to change the row order.

Related:

Remove Selected Row 310
Add Image List 297
Project Settings 21

12.7.1.1 Remove Selected Row

1. Go to the Translation table.
2. Click the row to select it.
3. Click the Remove Selected Row button.
4. A warning displays:

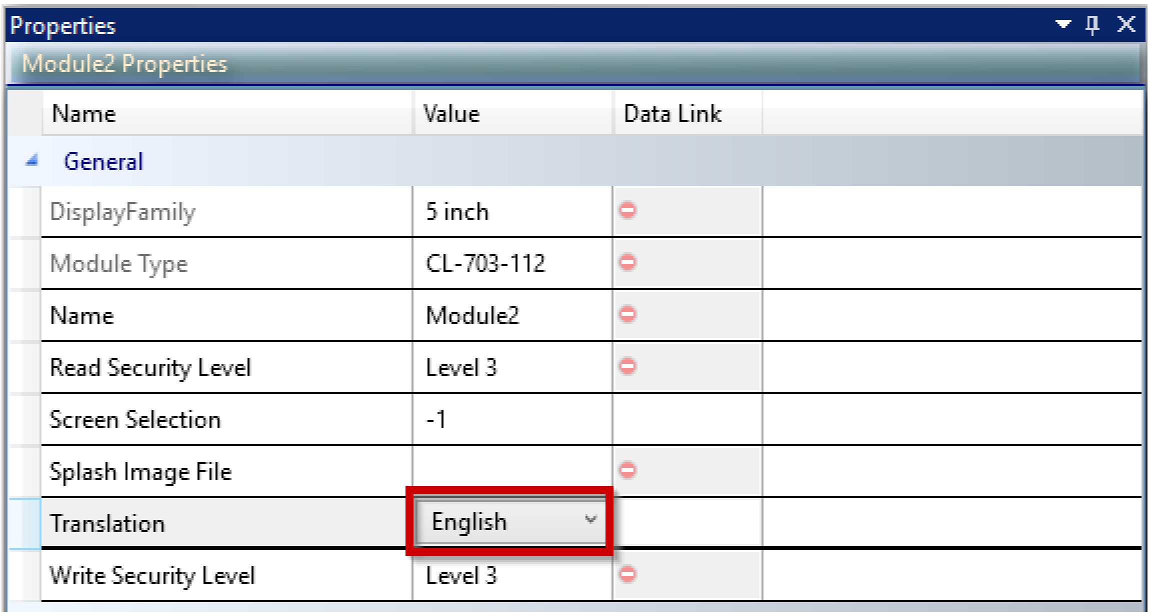
   “Removing a translation definition will delete the matching translation column from all project string lists.”
5. Click OK to remove the row and associated translation definition.
6. Click Cancel to keep the row in the table.

Related:
Add Translation 308

12.7.1.2 Using Translations

1. Add a translation 308 to the Orchestra project.
2. Add display modules 46 to the Orchestra project.
3. Go to the Project Explorer, and select the Display module from the HED Modules folder 29.
4. Go to the Properties pane.

5. Go to the Translation row, and click the drop-down menu in the Value cell.



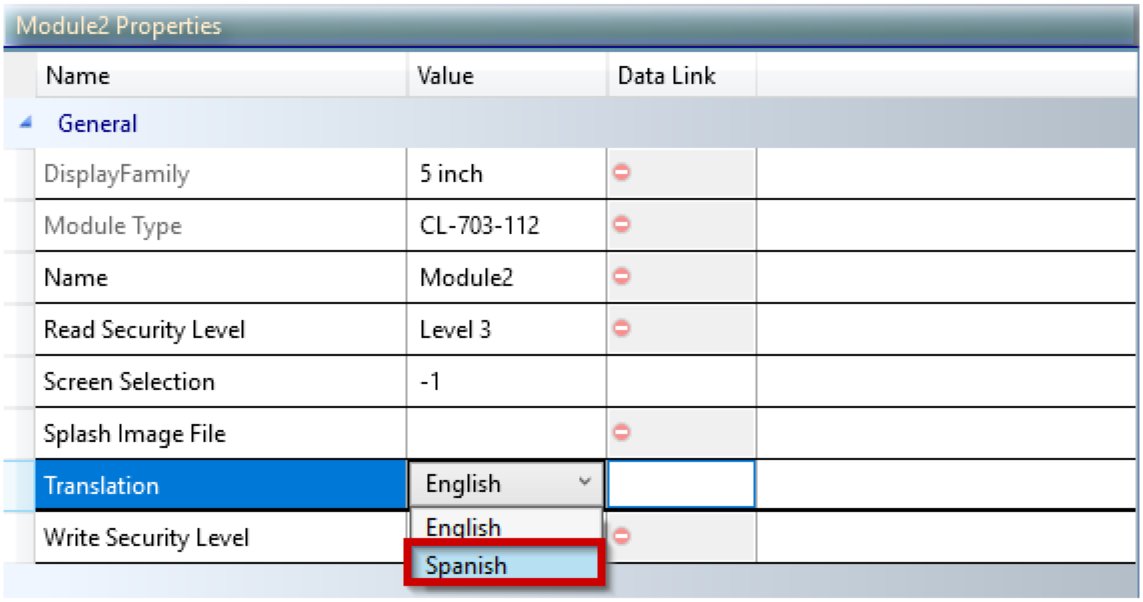
Properties

Module2 Properties

Name	Value	Data Link
General		
DisplayFamily	5 inch	⊖
Module Type	CL-703-112	⊖
Name	Module2	⊖
Read Security Level	Level 3	⊖
Screen Selection	-1	
Splash Image File		⊖
Translation	English	
Write Security Level	Level 3	⊖

Translation Value Cell

6. Choose a translation from the drop-down menu to assign it to the module.



Module2 Properties

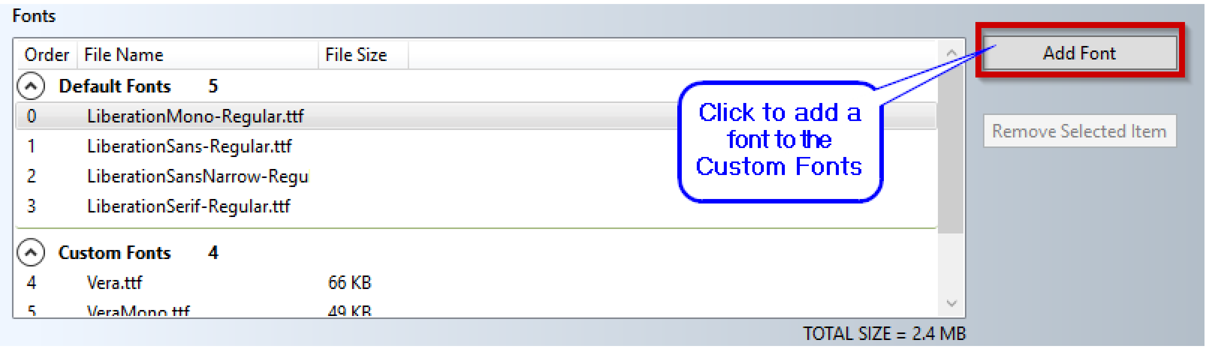
Name	Value	Data Link
General		
DisplayFamily	5 inch	⊖
Module Type	CL-703-112	⊖
Name	Module2	⊖
Read Security Level	Level 3	⊖
Screen Selection	-1	
Splash Image File		⊖
Translation	English English Spanish	
Write Security Level		⊖

Choose a Translation

12.7.2 Add Font

1. Go to the Fonts table.

2. Click the Add Font button.



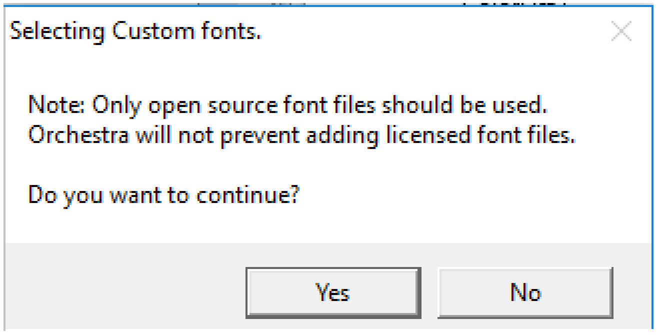
Order	File Name	File Size
Default Fonts 5		
0	LiberationMono-Regular.ttf	
1	LiberationSans-Regular.ttf	
2	LiberationSansNarrow-Regu	
3	LiberationSerif-Regular.ttf	
Custom Fonts 4		
4	Vera.ttf	66 KB
5	VeraMono.ttf	49 KB

TOTAL SIZE = 2.4 MB

Add Font button

3. A warning displays:

   a. Selecting Custom Fonts.
   Note: Only open source font files should be used. Orchestra® will not prevent adding licensed font files. Do you want to continue?"



Selecting Custom fonts.

Note: Only open source font files should be used.
Orchestra will not prevent adding licensed font files.

Do you want to continue?

Selecting Custom Fonts Note

4. Click the Yes button to continue adding a font.

5. Click the No button to stop adding a new font.

6. A file explorer window opens.

7. Browse for and select an open source font file.

8. Go to the Fonts table.

9. Go to the Custom Fonts section of the Fonts table.

10. The new font is added to the last row.

Related:

Project Settings 21

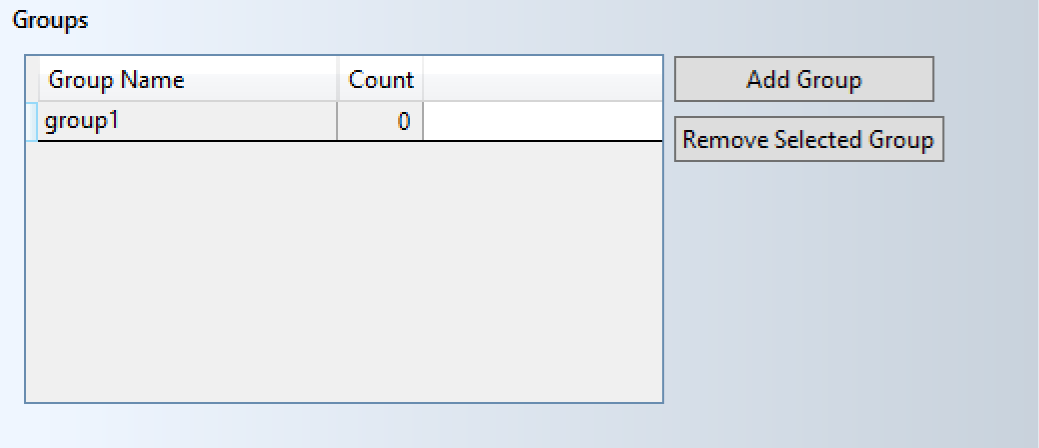
12.7.2.1 Remove Selected Row

1. Go to the Fonts table.
2. Click the row to select a font.
3. Click the Remove Selected Item button.
4. The font is removed from the fonts table.

Related:
Add Font 312

12.7.3 Add Group

1. Click the Add Group button.
2. Go to the Groups table.



Group Name	Count
group1	0

Groups Table

3. The new group is added to the last row.
4. Click the Group Name cell to change the name.

Related:
Remove Selected Group 314
Edit Project Settings 308
Project Settings 21

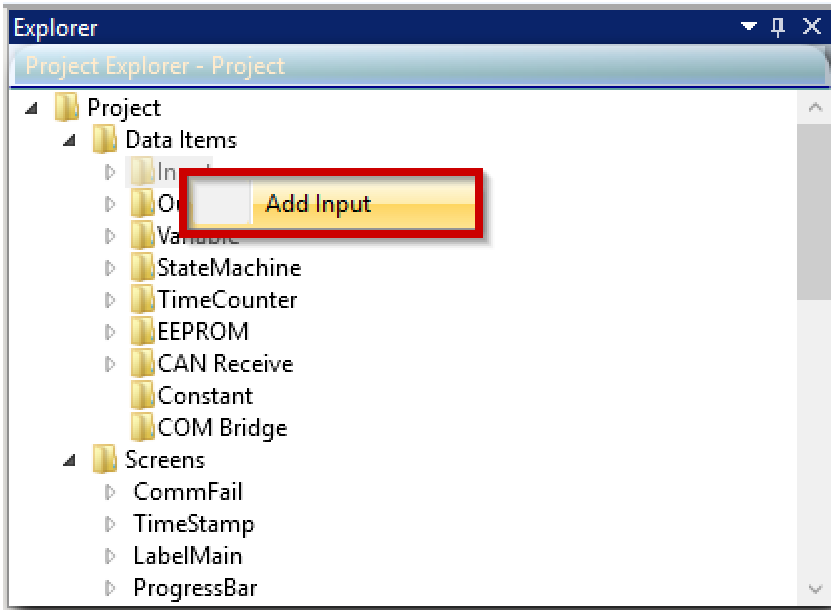
12.7.3.1 Remove Selected Group

1. Go to the Groups table.
2. Click the row to select a group.
3. Click the Remove Selected Item button.
4. The group is removed from the table.

Related:
Add Group 314

12.8 Add Input

1. Go to the Explorer pane.
2. Go to the Projects folder.
3. Go to the Data Items subfolder.
4. Go to the Input subfolder.
5. Right click the Input subfolder.
6. Select “Add Input”.



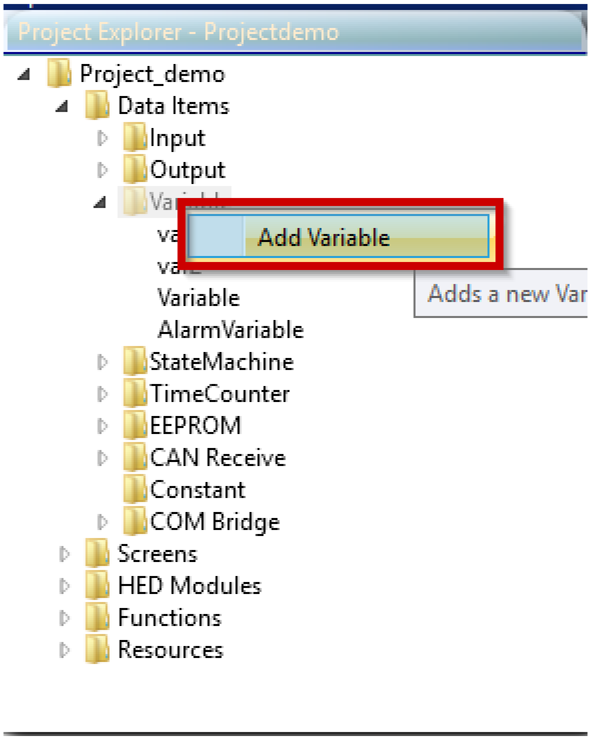
Add Input

7. A new Input is created within the subfolder.
8. View and edit the input properties within the Properties Pane.

Related:
Input 54

12.9 Add a Variable

1. Go to the Explorer pane.
2. Right click the Variable folder.
3. Click the Add Variable menu.



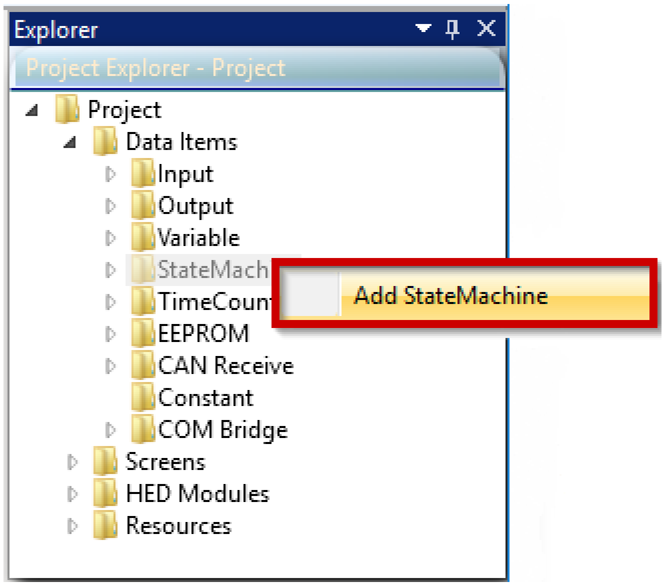
Add Variable Menu

4. The new variable is added to the bottom of the variable list.
5. Go to the Properties pane.
6. Set the variable properties.

Related:
Add a State 317
Set Variable Array Size 320
Variable 90

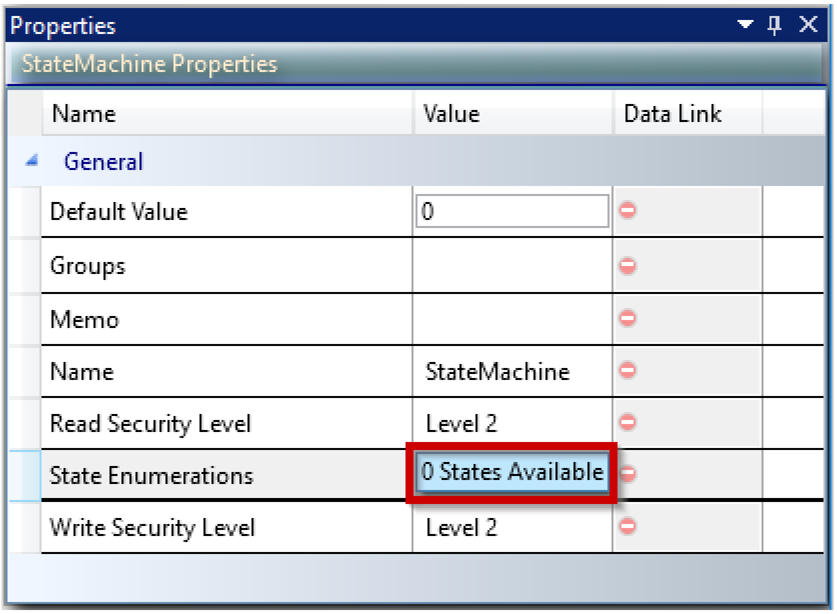
12.9.1 Add a State

1. Go to the Explorer pane.
2. Go to the Project folder.
3. Go to the Data Items folder.
4. Go to the State Machines folder.



Add State Machine

5. Right click the folder.
6. Click the Add State Machine button.
7. Go to the Properties pane.
8. Go to the State Enumerations Value cell.



Properties

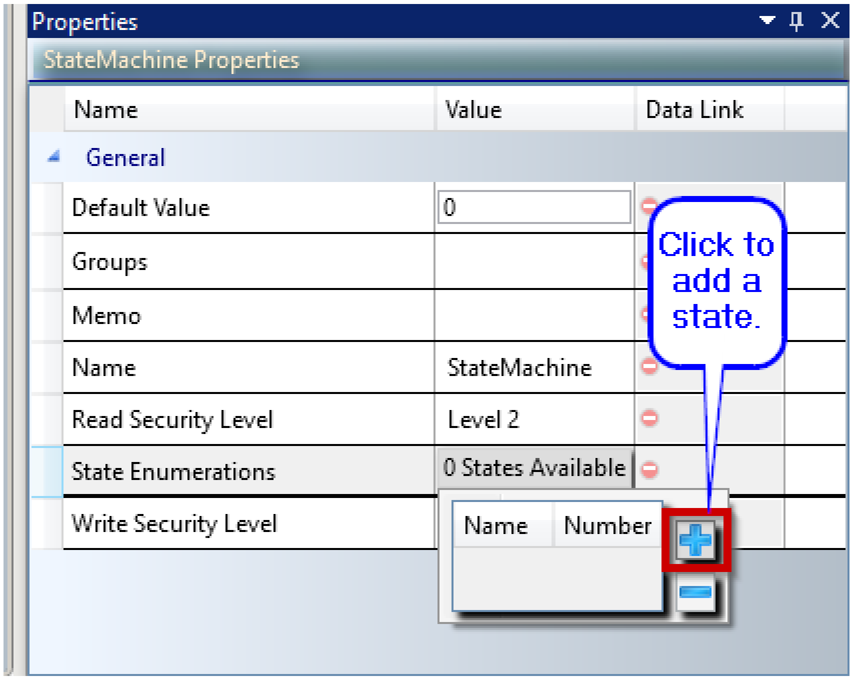
StateMachine Properties

Name	Value	Data Link
General		
Default Value	0	
Groups		
Memo		
Name	StateMachine	
Read Security Level	Level 2	
State Enumerations	0 States Available	
Write Security Level	Level 2	

Value Cell

9. Left click the Value button.

10. Click the "+" button to increase the number of states.



Properties

StateMachine Properties

Name	Value	Data Link
General		
Default Value	0	
Groups		
Memo		
Name	StateMachine	
Read Security Level	Level 2	
State Enumerations	0 States Available	
Write Security Level		

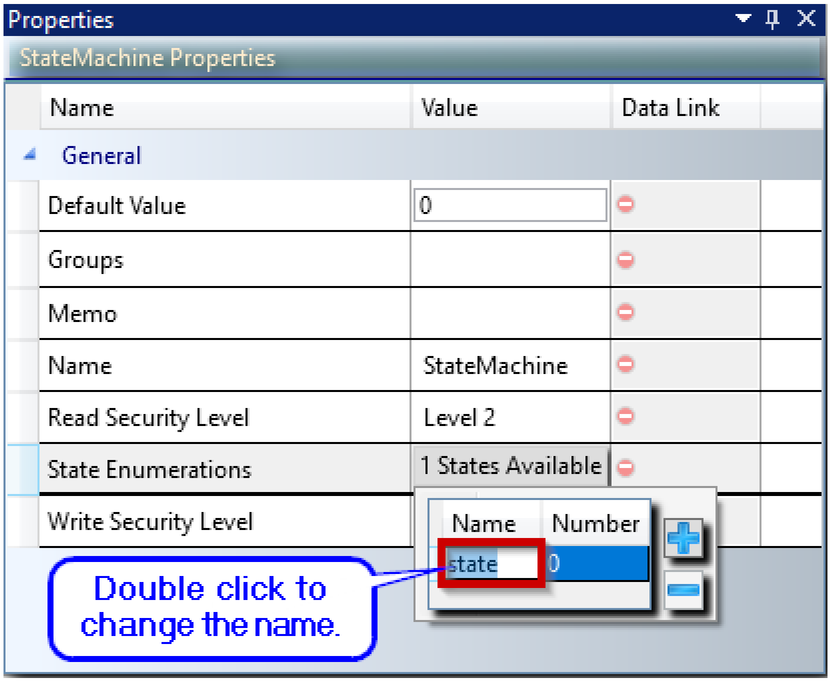
Add a State

11. Click the "-" button to decrease the number of states.

Related:
Edit State Name and Number 320
Set Variable Array Size 320

12.9.1.1 Edit State Name and Number

1. Left click on the number or name;
2. Once name/number is highlighted, type in a new name or number assignment.



Name	Value	Data Link
Default Value	0	
Groups		
Memo		
Name	StateMachine	
Read Security Level	Level 2	
State Enumerations	1 States Available	
Write Security Level		

Name	Number
state	0

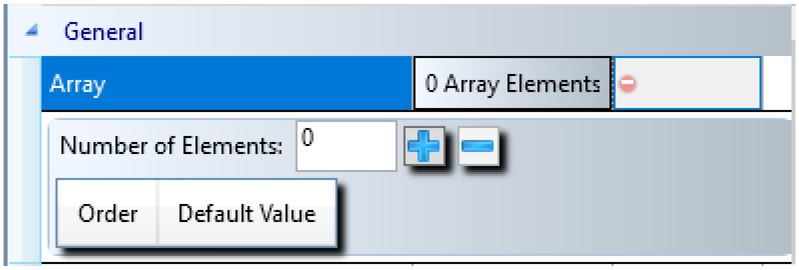
Double click to change the name.

Change State Name

Related:
[Add a State](317)

12.9.2 Set Variable Array Size

1. Left click on the Array property.

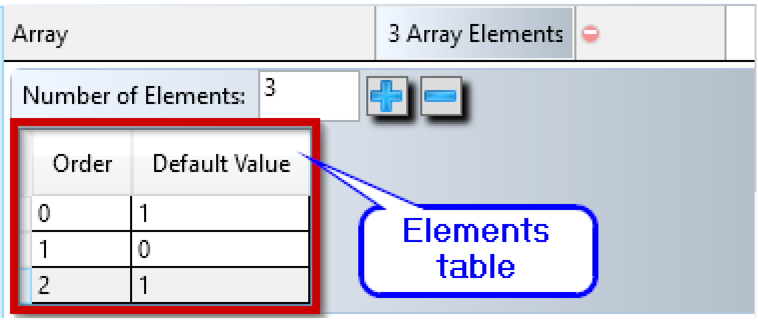


Name	Value
Array	0 Array Elements

Number of Elements: 0

Array Property

2. Either type in the size of the array or click the "+" button.
3. The elements are added to the table.



Array | 3 Array Elements

Number of Elements: 3

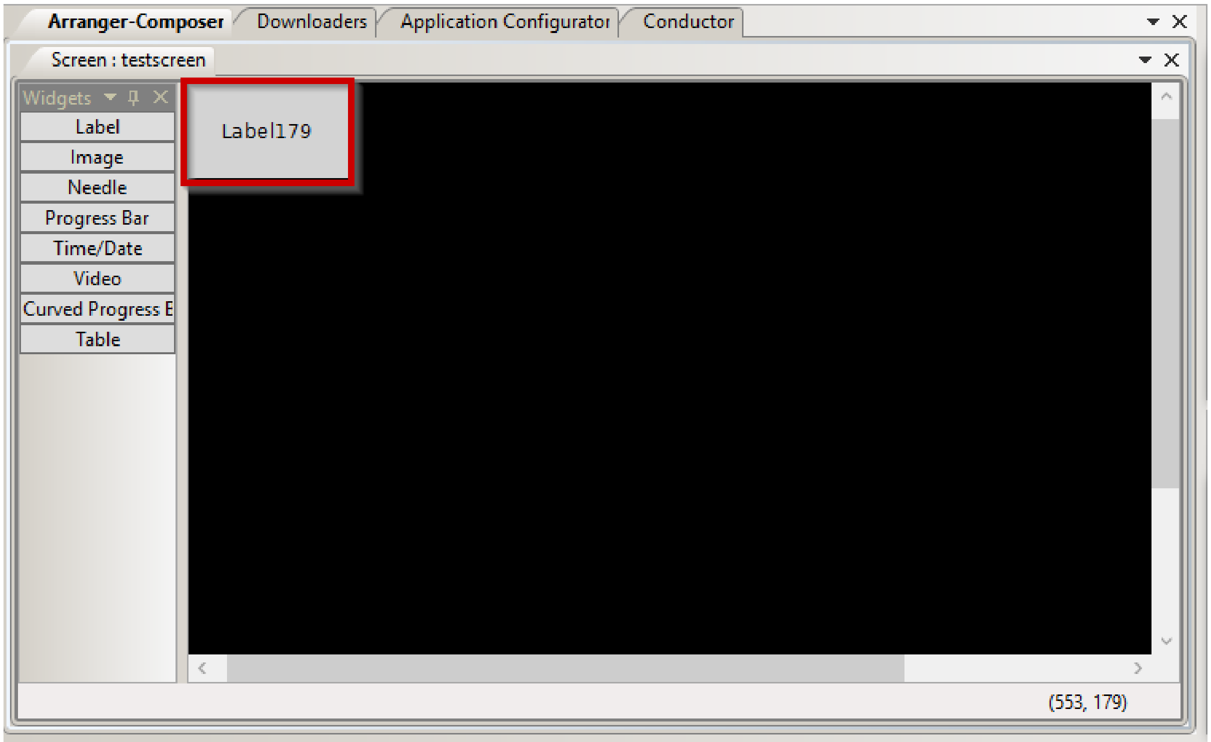
Order	Default Value
0	1
1	0
2	1

Elements Table

4. Change the default value.

12.10 Add Widgets

1. Left click on the desired Widget button from the Widget Panel on the left.
2. The widget appears in the far upper left corner of the screen (0, 0).
3. The widgets on your screen are also listed in the Explorer panel.



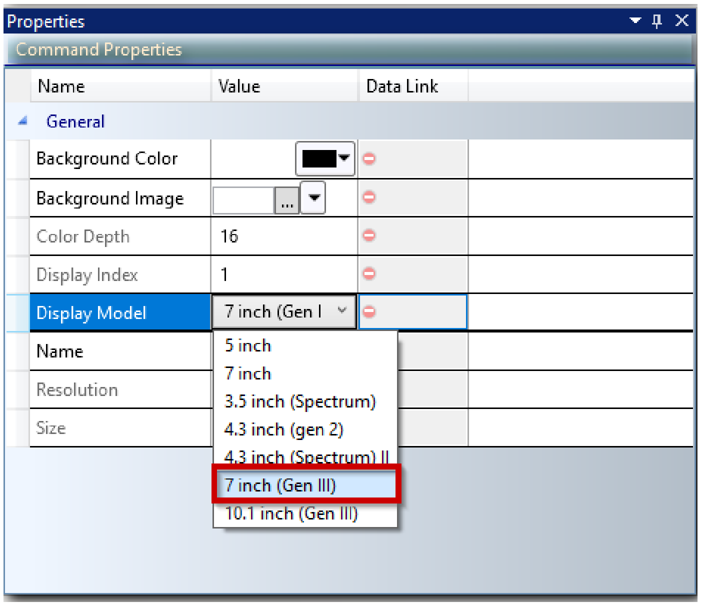
Widget Added to Screen

4. Click the widget to select its properties.
5. Go to the Properties pane.
6. Set the widget properties. Some properties are set by choosing from a drop-down box of possible values, and some properties are set by typing a new value.

Related:
Using Arranger Widgets 141

12.10.1 Activate Command Widget

1. Use Orchestra® 2.5.1 or higher to open or create a new project.
2. Add a CL-711-12 module.
3. Add a screen.
4. Go to the Explorer pane &#8594; Screens folder.
5. Select the screen that you want to add a Command widget to.
6. Go to the Properties pane&#8594; Display Model value cell.
7. Click the drop-down menu and choose 7 inch (Gen III).



Name	Value	Data Link
Background Color		
Background Image	...	
Color Depth	16	
Display Index	1	
Display Model	7 inch (Gen I	
Name		
Resolution		
Size		

Select 7 Inch (Gen III) Display Model

The Command widget is added to the Widgets pane.

Related:
Command Widget 165

12.11 Coding Within Orchestra® 2

Note: This section is for advanced users.

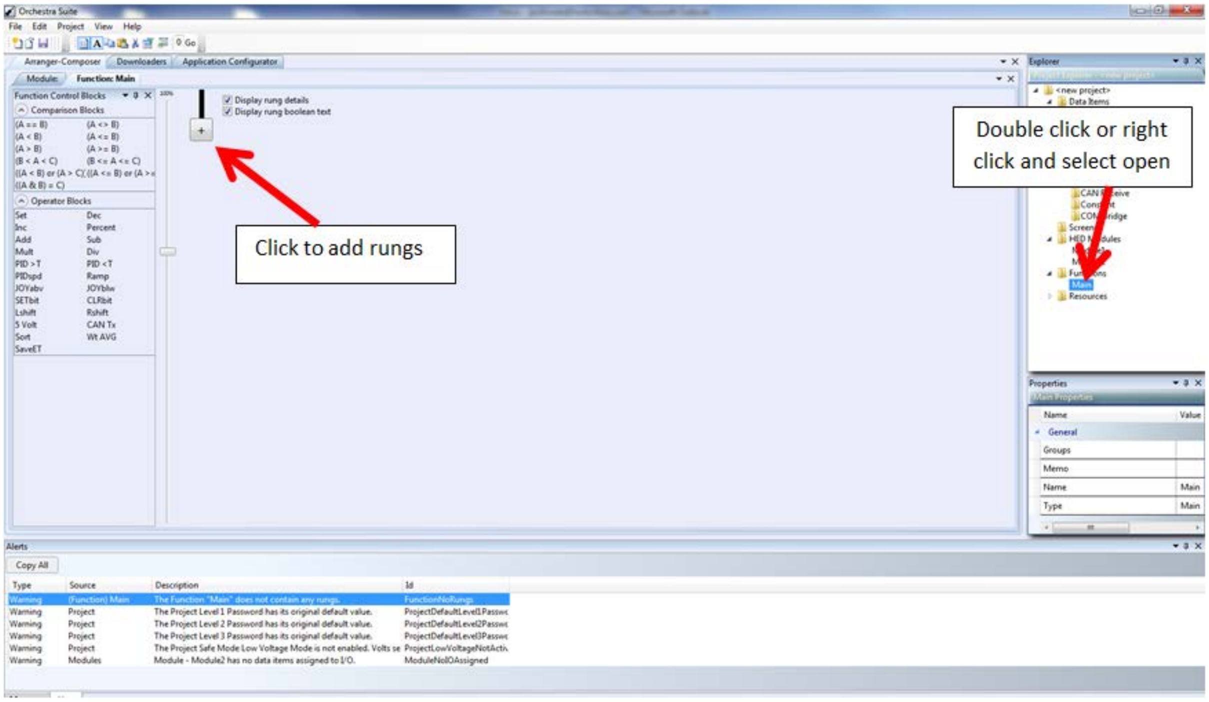
To code within Orchestra®, Rungs or Presto™ with Rungs must be selected as the compile option. Selecting one of those options activates the Functions folder in the Explorer Pane. The Functions folder is where the Main program will be stored.

Related:
Add Rungs 324
Add Function Control Blocks to Rungs 326
Comparison and Operator Blocks 326

12.11.1 Add Rungs

1. Double click Main to bring up a Function tab within the Designer pane where the ladder logic will be created.

2. Click the large "+" button.



Name	Value
General	
Groups	
Memo	
Name	Main
Type	Main

Type	Source	Description	Id
Warning	(Function) Main	The Function "Main" does not contain any rungs.	FunctionNoRungs
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passwc
Warning	Project	The Project Level 2 Password has its original default value.	ProjectDefaultLevel2Passwc
Warning	Project	The Project Level 3 Password has its original default value.	ProjectDefaultLevel3Passwc
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActi\
Warning	Modules	Module - Module2 has no data items assigned to I/O.	ModuleNoIOAssigned

Main Function Tab

3. A blank rung is added to the screen, with a generic name that can be changed within the Properties pane. That name appears under Main in the Functions folder, which allows the user to jump to specific rungs when needed.

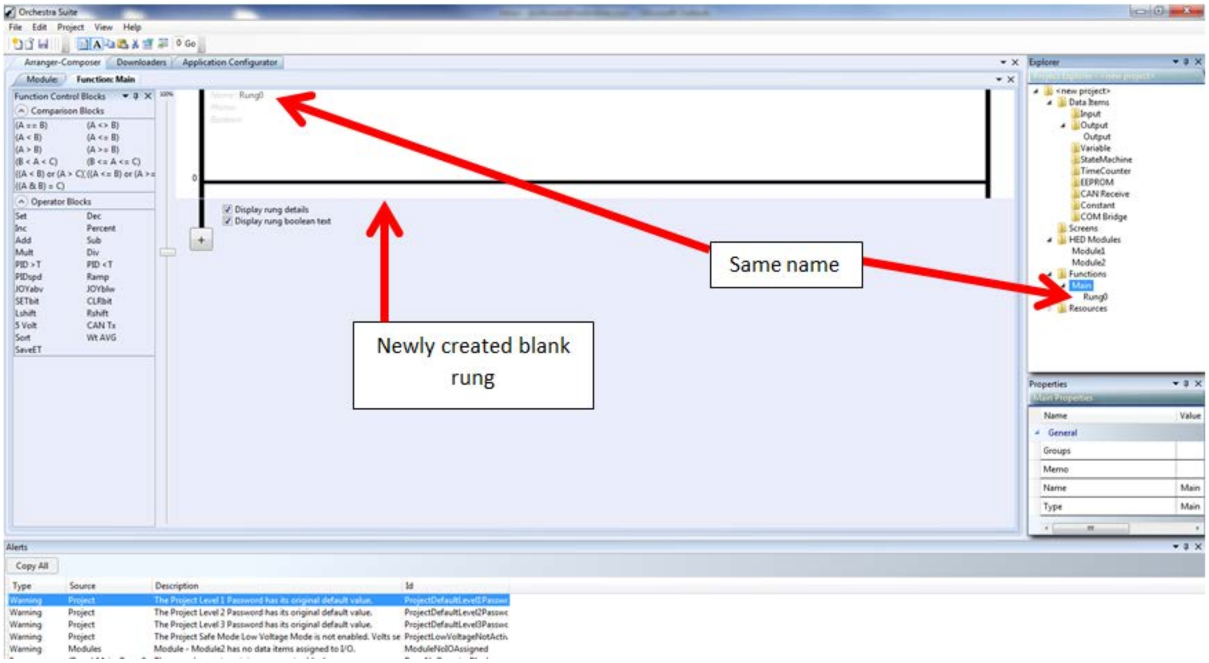
4. Right click on a rung to bring up options that allow the rung to be copied, cut, pasted, and other rungs to be inserted above or below.

Related:

Coding Within Orchestra 2 324
Add Function Control Blocks to Rungs 326
Comparison and Operator Blocks 326

12.11.2 Add Function Control Blocks to Rungs

1. Go to the Function Control Blocks panel.
2. Click and drag a comparison block into the rung.
3. Set the comparison block values.
4. Go to the Properties pane.
5. Set the comparison block properties.
6. Click and drag an operator block into the same rung.
7. Set the Operator block values.
8. Go to the Properties pane.
9. Set the operator block properties.



Same name

Newly created blank rung

Name	Value
General	
Groups	
Memo	
Name	Main
Type	Main

Type	Source	Description	Id
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultLevel1Passw
Warning	Project	The Project Level 2 Password has its original default value.	ProjectDefaultLevel2Passwc
Warning	Project	The Project Level 3 Password has its original default value.	ProjectDefaultLevel3Passwc
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActi
Warning	Modules	Module - Module2 has no data items assigned to I/O.	ModuleNoIOAssigned

Blank Rung

Related:

Add Rungs 324

Comparison and Operator Blocks 326

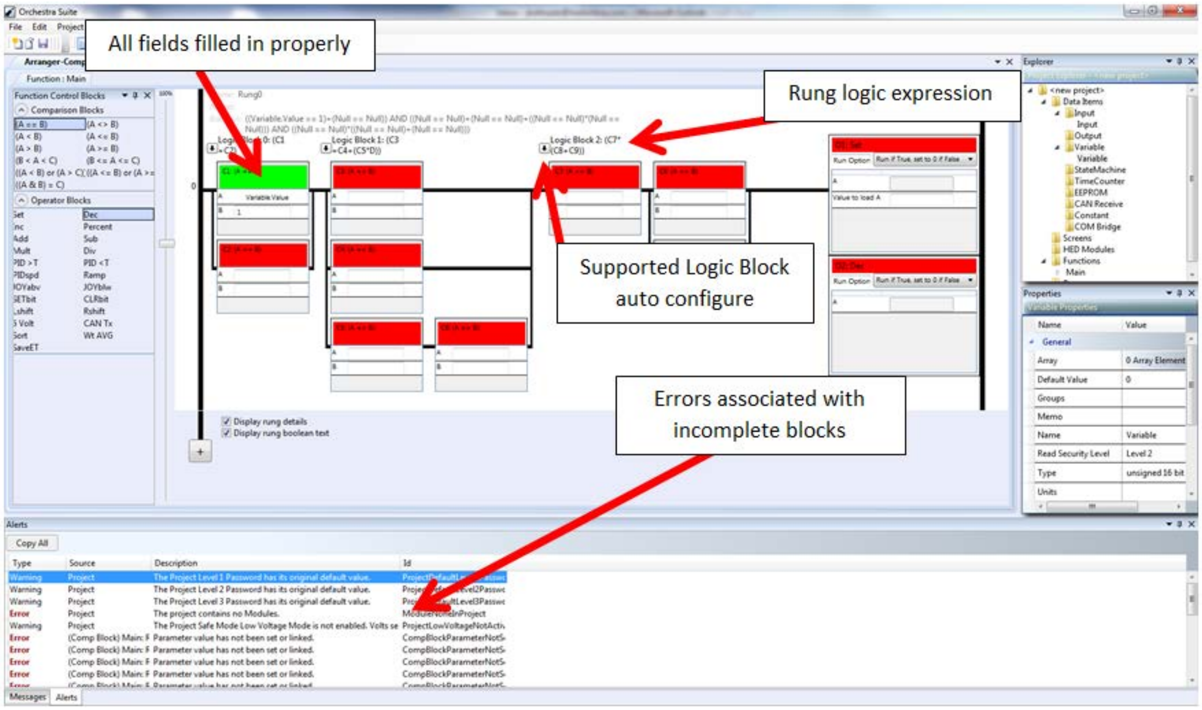
12.11.3 Comparison and Operator Blocks

Programming ladder logic in Orchestra® is done by clicking and dragging the Comparison Block(s) and Operator Block(s) to the rung.

Related:
[Comparison Logic Blocks](328)

12.11.3.1 Access Properties for Comparison Block Use

1. Take the mouse pointer and hover over the data item after it has been linked to a data field within the block.
2. A small downward pointing arrow appears to the left of the Data Item name.
3. Clicking that arrow will drop-down a small menu for the user to choose from. Most items only have “Value” within that drop-down, but if it has anything else such as a “Status” it will be shown within that menu.
4. Data items that have the ability to become arrays have a second drop-down menu that will allow the user to select a specific element within the array to do the comparison on, and the selected element will be indicated within the data item name in the Comparison Block.
5. All data items will have a red “X” next to those menus that will remove the selected data item from the Comparison Block.


All fields filled in properly

Rung logic expression

Supported Logic Block auto configure

Errors associated with incomplete blocks

Type	Source	Description	Id
Warning	Project	The Project Level 1 Password has its original default value.	ProjectDefaultL
Warning	Project	The Project Level 2 Password has its original default value.	Proje evel2Passwc
Warning	Project	The Project Level 3 Password has its original default value.	Pro ultLevel3Passwc
Error	Project	The project contains no Modules.	ModuleNoneInProject
Warning	Project	The Project Safe Mode Low Voltage Mode is not enabled. Volts se	ProjectLowVoltageNotActi\
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS
Error	(Comp Block) Main: F	Parameter value has not been set or linked.	CompBlockParameterNotS

Rung Example

Related:
[Add Function Control Blocks to Rungs](326)
[Comparison Logic Blocks](328)

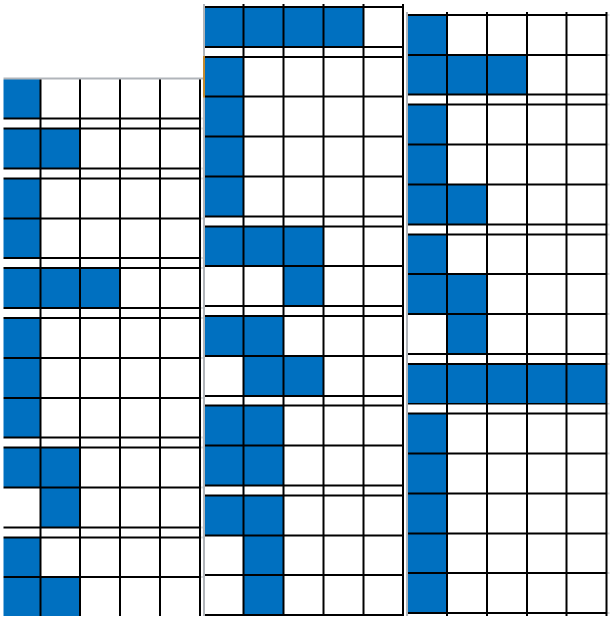
12.11.3.2 Comparison Logic Blocks

The Comparison Blocks are grouped together into logic blocks. Each logic block can have up to five Comparison Blocks in each logic block, and each rung can support up to five logic blocks for a max of 25 blocks pending configuration.

Related:
Supported Configurations 328
Comparison and Operator Blocks 326

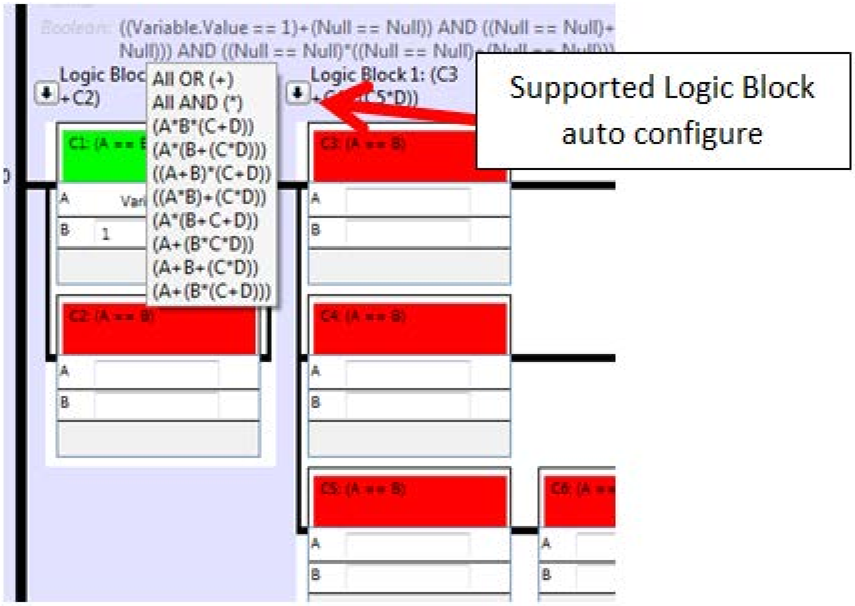
12.11.3.2.1 Supported Configurations

A way to determine supported configurations is to click the small downward arrow box to the left of the logic block. The drop-down will display all supported logic variations with that number of comparison blocks within the logic block, and rearrange the blocks automatically once selected.



Supported Comparison Block Configurations

The rung can also support up to a max of 25 Operator Blocks. Each block has a color band on it, either red or green. The block will remain red and an error will appear in the Alerts pane as long as the block does not have all of the require fields filled in properly. Once each field is filled in properly, the block will turn green and the errors will disappear from the Alerts pane.



Supported Logic Block Auto Configure Example

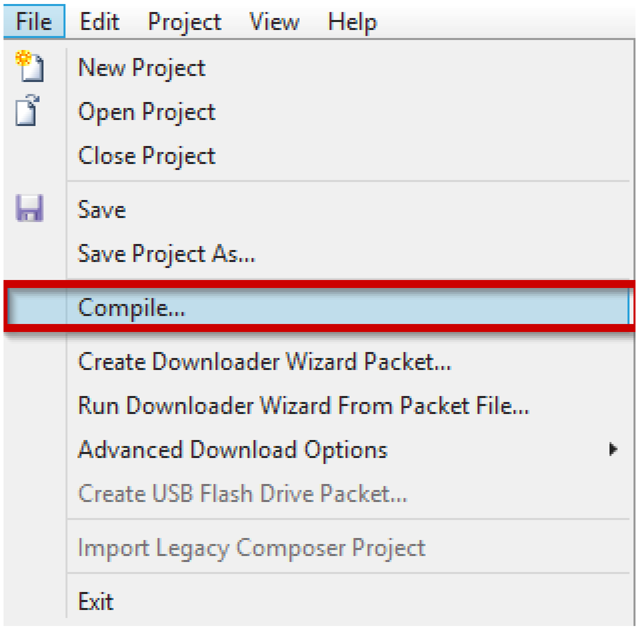
Related:

Comparison Logic Blocks 328
Comparison and Operator Blocks 326

12.12 Compile a Project

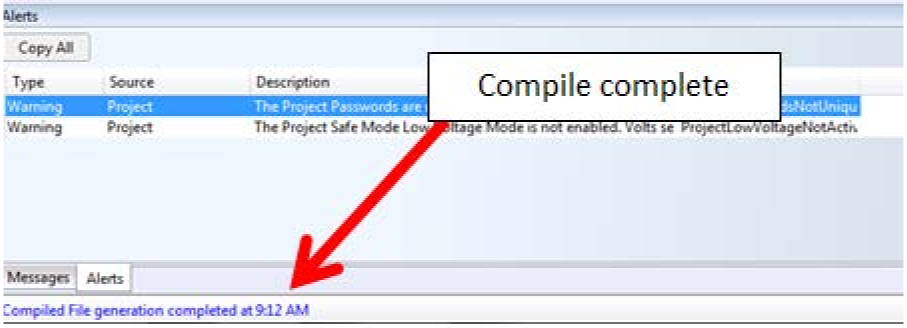
Once the application is written, it needs to be compiled before it can be downloaded to the system.

1. Save the project by clicking on the File drop-down menu and selecting Save.
2. After the application is saved, use the File drop-down menu and click the “Compile…” option.


File
Edit
Project
View
Help
New Project
Open Project
Close Project
Save
Save Project As...
Compile...
Create Downloader Wizard Packet...
Run Downloader Wizard From Packet File...
Advanced Download Options
Create USB Flash Drive Packet...
Import Legacy Composer Project
Exit

File-Compile

3. In the bottom left hand corner of the screen, Orchestra® will give the status of the compile.



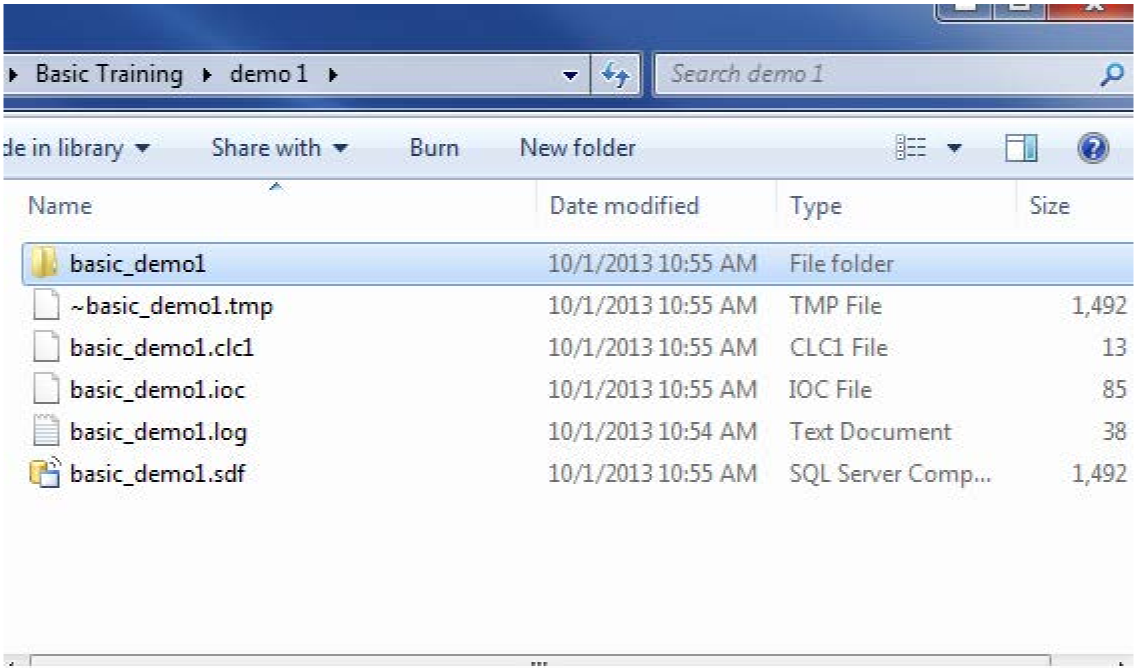
Type	Source	Description
Warning	Project	The Project Passwords are dsNotUniqu
Warning	Project	The Project Safe Mode Low ltage Mode is not enabled. Volts se ProjectLowVoltageNotActi\

Compile complete

Compiled File generation completed at 9:12 AM

Compile Status

4. From the compile, a number of files are created within the save directory that the project was saved in.



Name	Date modified	Type	Size
basic_demo1	10/1/2013 10:55 AM	File folder	
~basic_demo1.tmp	10/1/2013 10:55 AM	TMP File	1,492
basic_demo1.clc1	10/1/2013 10:55 AM	CLC1 File	13
basic_demo1.ioc	10/1/2013 10:55 AM	IOC File	85
basic_demo1.log	10/1/2013 10:54 AM	Text Document	38
basic_demo1.sdf	10/1/2013 10:55 AM	SQL Server Comp...	1,492

Generated Files

Related:

Compiled Project Folder 331
Set Compile Option 289
Compile Option 35

12.12.1 Compiled Project Folder

The file folder with the project name contains auto-generated application files that includes any fonts, images, display objects, etc. associated with the project. The .clc1 file is a file that pertains to a rung only project, and is what is downloaded to the Module using the Downloader tool.

Conductor™ uses the .ioc file to debug, calibrate, troubleshoot, etc. The .log file contains a record of each time the project was compiled, and the .sdf file is what is needed for Orchestra® to open and edit the project.

Note: When these files are generated the .ioc and .clc1, that is loaded into the Module, must match if the user would like to use Conductor™ with that particular software on the module.

Related:

Compile a Project 330

12.13 Valid and Invalid Orchestra Characters

There are several groups within Orchestra with different rules for the types of allowed characters.

- Windows file names
- Linux file names
- Orchestra/CANLink module and property naming
- Unicode widget text

HED® has several string validation functions for different Orchestra® elements.

From these functions, users can get the rules.

Name Validation Code

This is the HED® name validation code. These are the characters HED® validates for names: upper or lower case no spaces, numbers.

```
System.Text.RegularExpressions.Regex.Replace(name, "[a-z,_,0-9]", "",
System.Text.RegularExpressions.RegexOptions.IgnoreCase)
```

This is just for names, however.   Different areas, like string lists and labels, have different limitations and formats.

Characters Allowed in File Names

```
"^0-9A-Za-z _!#%',-;=@~$.+()[]{};
```

Using Presto™

13 Using Presto™

Based on the requirements of the application and the customer, Orchestra® allows the software to be written with rungs using ladder logic or coded within C or C++.

To program with Presto™, set the Properties Compile option to either Presto™ or Presto™ with Rungs.

Related:

Compile Option 35
Coding Within Orchestra 2 324

Troubleshoot Guide

14 Troubleshoot Guide

14.1 Conductor Troubleshoot Guide

Why doesnt the Conductor™ connect with my equipment's system?

- Check for proper connection between the master module and the PC.
- Verify that you are selecting the correct .ioc file configured for your system.
- Check for adequate power supplied to the master module on your system.

How can I verify that the Conductor™ is communicating with my equipment's system?

- Check that the **Connect/Disconnect** button shows connected.
- Check that the red Comm icon is flashing in the status bar at the bottom of the main window.

Some features are not available in my Conductor™ software. Why?

- Your password does not allow access to those features.
- The system was not configured to allow those features.
- This software release does not support these features at this time.

How can I obtain new/change passwords, obtain a new .ioc file or change access levels?

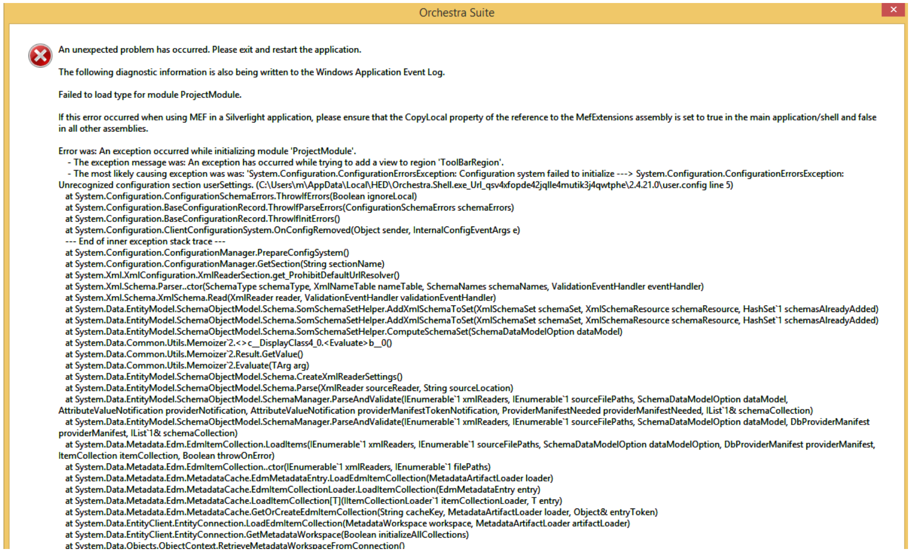
- Contact the Original Equipment Manufacturer (OEM) for assistance.

14.2 Known Start Up Issues

14.2.1 Start up error exception

Problem: Start up error exception

There are a few possible reasons for this error. The most common ones are listed below, followed by an example issue.



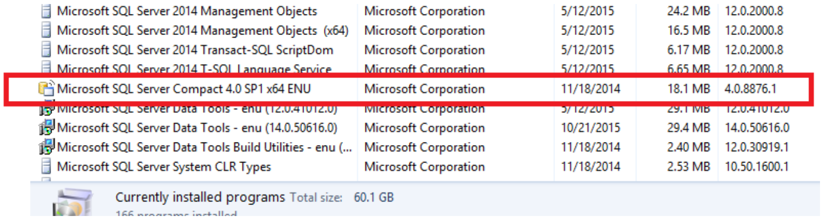
Orchestra Suite

```
An unexpected problem has occurred. Please exit and restart the application.

The following diagnostic information is also being written to the Windows Application Event Log.

Failed to load type for module ProjectModule.

If this error occurred when using MEF in a Silverlight application, please ensure that the CopyLocal property of the reference to the MefExtensions assembly is set to true in the main application/shell and false
in all other assemblies.

Error was: An exception occurred while initializing module 'ProjectModule'.
   - The exception message was: An exception has occurred while trying to add a view to region 'ToolBarRegion'.
   - The most likely causing exception was was: 'System.Configuration.ConfigurationErrorsException: Configuration system failed to initialize ---> System.Configuration.ConfigurationErrorsException:
Unrecognized configuration section userSettings. (C:\Users\m\AppData\Local\HED\Orchestra.Shell.exe_Url_qsv4xfopde42jqlle4mutik3j4qwtphe\2.4.21.0\user.config line 5)
   at System.Configuration.ConfigurationSchemaErrors.ThrowIfErrors(Boolean ignoreLocal)
   at System.Configuration.BaseConfigurationRecord.ThrowIfParseErrors(ConfigurationSchemaErrors schemaErrors)
   at System.Configuration.BaseConfigurationRecord.ThrowIfInitErrors()
   at System.Configuration.ClientConfigurationSystem.OnConfigRemoved(Object sender, InternalConfigEventArgs e)
   --- End of inner exception stack trace ---
   at System.Configuration.ConfigurationManager.PrepareConfigSystem()
   at System.Configuration.ConfigurationManager.GetSection(String sectionName)
   at System.Xml.XmlConfiguration.XmlReaderSection.get_ProhibitDefaultUrlResolver()
   at System.Xml.Schema.Parser..ctor(SchemaType schemaType, XmlNameTable nameTable, SchemaNames schemaNames, ValidationEventHandler eventHandler)
   at System.Xml.Schema.XmlSchema.Read(XmlReader reader, ValidationEventHandler validationEventHandler)
   at System.Data.EntityModel.SchemaObjectModel.Schema.SomSchemaSetHelper.AddXmlSchemaToSet(XmlSchemaSet schemaSet, XmlSchemaResource schemaResource, HashSet`1 schemasAlreadyAdded)
   at System.Data.EntityModel.SchemaObjectModel.Schema.SomSchemaSetHelper.AddXmlSchemaToSet(XmlSchemaSet schemaSet, XmlSchemaResource schemaResource, HashSet`1 schemasAlreadyAdded)
   at System.Data.EntityModel.SchemaObjectModel.Schema.SomSchemaSetHelper.ComputeSchemaSet(SchemaDataModelOption dataModel)
   at System.Data.Common.Utils.Memoizer`2.<>c__DisplayClass4_0.<Evaluate>b__0()
   at System.Data.Common.Utils.Memoizer`2.Result.GetValue()
   at System.Data.Common.Utils.Memoizer`2.Evaluate(TArg arg)
   at System.Data.EntityModel.SchemaObjectModel.Schema.CreateXmlReaderSettings()
   at System.Data.EntityModel.SchemaObjectModel.Schema.Parse(XmlReader sourceReader, String sourceLocation)
   at System.Data.EntityModel.SchemaObjectModel.SchemaManager.ParseAndValidate(IEnumerable`1 xmlReaders, IEnumerable`1 sourceFilePaths, SchemaDataModelOption dataModel,
AttributeValueNotification providerNotification, AttributeValueNotification providerManifestTokenNotification, ProviderManifestNeeded providerManifestNeeded, IList`1& schemaCollection)
   at System.Data.EntityModel.SchemaObjectModel.SchemaManager.ParseAndValidate(IEnumerable`1 xmlReaders, IEnumerable`1 sourceFilePaths, SchemaDataModelOption dataModel, DbProviderManifest
providerManifest, IList`1& schemaCollection)
   at System.Data.Metadata.Edm.EdmItemCollection.LoadItems(IEnumerable`1 xmlReaders, IEnumerable`1 sourceFilePaths, SchemaDataModelOption dataModelOption, DbProviderManifest providerManifest,
ItemCollection itemCollection, Boolean throwOnError)
   at System.Data.Metadata.Edm.EdmItemCollection..ctor(IEnumerable`1 xmlReaders, IEnumerable`1 filePaths)
   at System.Data.Metadata.Edm.MetadataCache.EdmMetadataEntry.LoadEdmItemCollection(MetadataArtifactLoader loader)
   at System.Data.Metadata.Edm.MetadataCache.EdmItemCollectionLoader.LoadItemCollection(EdmMetadataEntry entry)
   at System.Data.Metadata.Edm.MetadataCache.LoadItemCollection[T](IItemCollectionLoader`1 itemCollectionLoader, T entry)
   at System.Data.Metadata.Edm.MetadataCache.GetOrCreateEdmItemCollection(String cacheKey, MetadataArtifactLoader loader, Object& entryToken)
   at System.Data.EntityClient.EntityConnection.LoadEdmItemCollection(MetadataWorkspace workspace, MetadataArtifactLoader artifactLoader)
   at System.Data.EntityClient.EntityConnection.GetMetadataWorkspace(Boolean initializeAllCollections)
   at System.Data.Objects.ObjectContext.RetrieveMetadataWorkspaceFromConnection()
```

Orchestra® Start Up Error Exception

Causes and steps to Resolve

1. SQL Server Compact edition is not installed. This should get installed as part of Orchestra® but it should still be checked in Control Panel:



Microsoft SQL Server 2014 Management Objects	Microsoft Corporation	5/12/2015	24.2 MB	12.0.2000.8
Microsoft SQL Server 2014 Management Objects (x64)	Microsoft Corporation	5/12/2015	16.5 MB	12.0.2000.8
Microsoft SQL Server 2014 Transact-SQL ScriptDom	Microsoft Corporation	5/12/2015	6.17 MB	12.0.2000.8
Microsoft SQL Server 2014 T-SQL Language Service	Microsoft Corporation	5/12/2015	6.65 MB	12.0.2000.8
Microsoft SQL Server Compact 4.0 SP1 x64 ENU	Microsoft Corporation	11/18/2014	18.1 MB	4.0.8876.1
Microsoft SQL Server Data Tools - enu (12.0.41012.0)	Microsoft Corporation	5/12/2015	29.1 MB	12.0.41012.0
Microsoft SQL Server Data Tools - enu (14.0.50616.0)	Microsoft Corporation	10/21/2015	29.4 MB	14.0.50616.0
Microsoft SQL Server Data Tools Build Utilities - enu (...	Microsoft Corporation	11/18/2014	2.40 MB	12.0.30919.1
Microsoft SQL Server System CLR Types	Microsoft Corporation	11/18/2014	2.53 MB	10.50.1600.1

Currently installed programs Total size: 60.1 GB
166 programs installed

SQL Server Compact Installed

2. The Orchestra® install folder is not read/write. Even though the CANLink®Modules.sdf database is not written to SQL Compact requires read/write access to the database. Also, HED® writes log files to the install directory.

3. The user that does the install should have admin privileges. There may be somethings in Orchestra® that will not install if the user doesn't have admin rights.

4. The user.config file is corrupt or empty. HED® hasn't found how this can happen and only ran into it once or twice, but it should be checked. Deleting the existing file will fix the issue.

The user.config file is located here:

```
C:\Users\<username of logged on user>\AppData\Local\HED\Orchestra.Shell.exe_Url_xxxxxxxxxxxxxxxx\
```

The Orchestra.Shell.exe_ Url will differ by installation, but it should be obvious which directory it is.

5. The problem has to do with a configuration file that is used by Windows for the data base providers. The name of the file is machine.config. It is located here:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\Config\machine.config.
```

There should an entry for the data provider for SQL compact in the file. If it is missing, there will be an error:

```
<system.data>
         <DbProviderFactories><add name="Microsoft SQL Server Compact Data Provider 4.0" invariant
    </system.data>
```

Example

The problem in the example was that the company had a listing for a DB2 database and then the SQL Compact was added when that was installed. There is a bug in SQL install; it adds an additional <DbProvider/> tag if there is a database that exists. It looks like this:

```
     <system.data>
         <DbProviderFactories>
             <add name="IBM DB2 for i .NET Provider" invariant="IBM.Data.DB2.iSeries" description="
         <add name="Microsoft SQL Server Compact Data Provider 4.0" invariant="System.Data.SqlServe
          <DbProviderFactories/>
    </system.data>
```

In this instance, the machine.config file must be edited and the second tag removed so it looks like this:

```
     <system.data>
         <DbProviderFactories>
             <add name="IBM DB2 for i .NET Provider" invariant="IBM.Data.DB2.iSeries" description="
         <add name="Microsoft SQL Server Compact Data Provider 4.0" invariant="System.Data.SqlServe
     </system.data>
```

14.2.2 Licensing problem

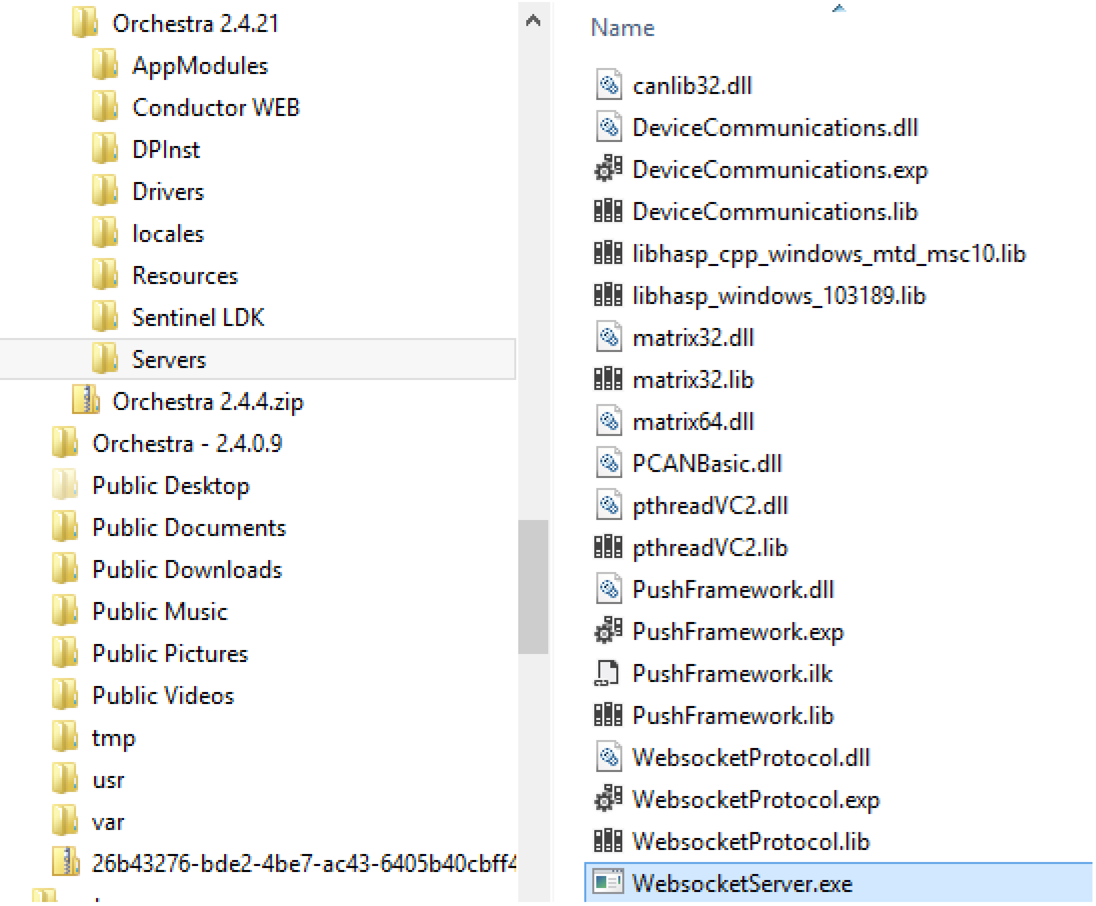
Problem

This problem has to do with Orchestra® coming up with the limited no license configuration.

Steps to Resolve

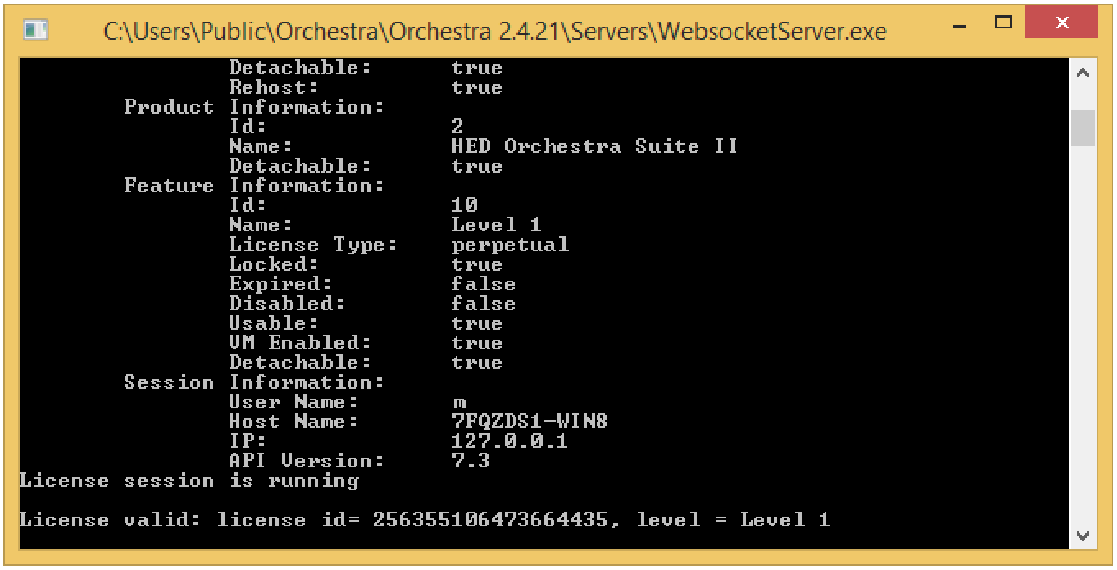
If the user has a legitimate license, then the first thing to check is if the web socket server shows the license.

To do this, start up the web socket server in console mode by itself:



Websocket Server.exe

The license information will show in the console:



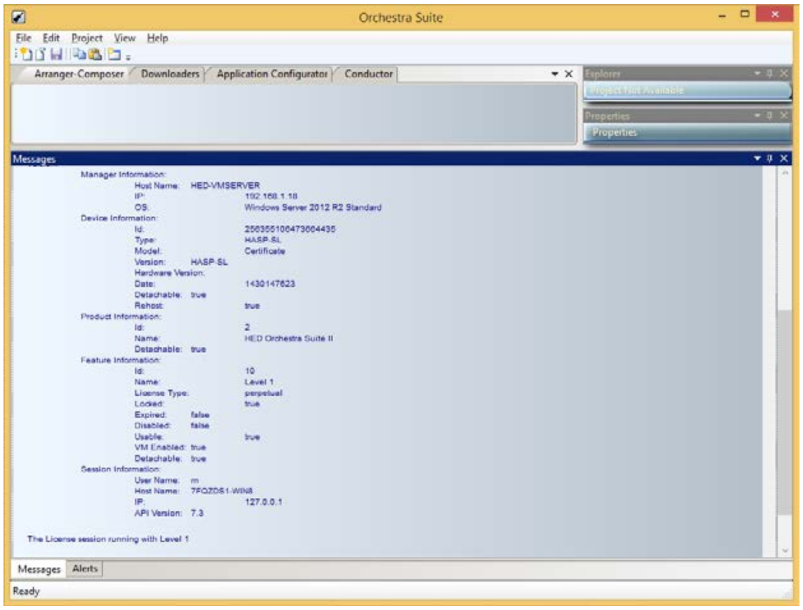
C:\Users\Public\Orchestra\Orchestra 2.4.21\Servers\WebsocketServer.exe

```
                Detachable:      true
                Rehost:          true
        Product Information:
                Id:              2
                Name:            HED Orchestra Suite II
                Detachable:      true
        Feature Information:
                Id:              10
                Name:            Level 1
                License Type:    perpetual
                Locked:          true
                Expired:         false
                Disabled:        false
                Usable:          true
                VM Enabled:      true
                Detachable:      true
        Session Information:
                User Name:       m
                Host Name:       7FQZDS1-WIN8
                IP:              127.0.0.1
                API Version:     7.3
License session is running

License valid: license id= 256355106473664435, level = Level 1
```

License Information in Console

If it doesn't, then there is something wrong with the license.

If it does, then run Orchestra® and see if the same information shows in the Orchestra® messages window:



```
        Manager Information:
                Host Name:   HED-VMSERVER
                IP:                  192.168.1.18
                OS:                  Windows Server 2012 R2 Standard
        Device Information:
                Id:                  256355106473664435
                Type:                HASP-SL
                Model:               Certificate
                Version:     HASP-SL
                Hardware Version:
                Date:                1430147623
                Detachable:  true
                Rehost:              true
        Product Information:
                Id:                  2
                Name:                HED Orchestra Suite II
                Detachable:  true
        Feature Information:
                Id:                  10
                Name:                Level 1
                License Type:        perpetual
                Locked:              true
                Expired:     false
                Disabled:    false
                Usable:              true
                VM Enabled:  true
                Detachable:  true
        Session Information:
                User Name:   m
                Host Name:   7FQZDS1-WIN8
                IP:                  127.0.0.1
                API Version: 7.3

The License session running with Level 1
```

Orchestra® Message Window

Additional Cause

This means that the web socket server and Orchestra® are not communicating. That means that probably there is a firewall blocking the socket communications. The user will have to check with their IT group to figure out the firewall issue.

The firewall that has been reported to HED® recently is the Kaspersky firewall.

Related:

Other Licensing Problems 344

14.2.2.1 Other Licensing Problems

There have been problems where the licensing does not work. These steps should help to solve the problem.

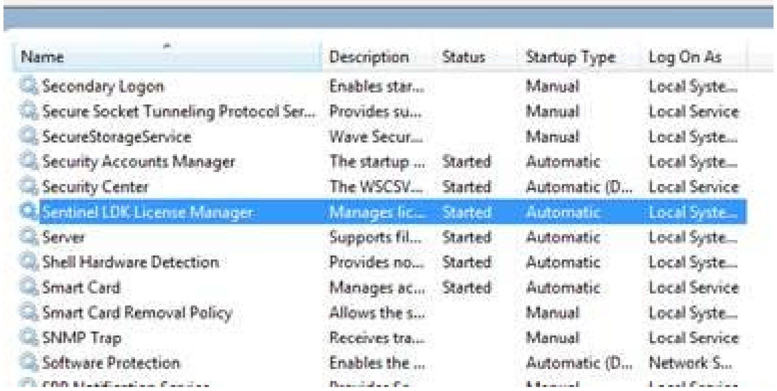
In some cases, HED® still does not know why the license install does work on some computers, either when installing Orchestra® or when manually running the install program.

Typically, the two reasons why the license does not install is because of user access problems or an anti-virus program running that blocks the install. Both these conditions should be checked.

Steps to follow when the license does not install:

Make sure the service (Sentinel LDK) is installed and started:

1. Go to: Control Panel\All Control Panel Items\Administrative Tools\Services
2. Scroll down to Sentinel LDK License Manager and check the status column for “Started”.



Name	Description	Status	Startup Type	Log On As
Secondary Logon	Enables star...		Manual	Local Syste...
Secure Socket Tunneling Protocol Ser...	Provides su...		Manual	Local Service
SecureStorageService	Wave Secur...		Manual	Local Syste...
Security Accounts Manager	The startup ...	Started	Automatic	Local Syste...
Security Center	The WSCSV...	Started	Automatic (D...	Local Service
Sentinel LDK License Manager	Manages lic...	Started	Automatic	Local Syste...
Server	Supports fil...	Started	Automatic	Local Syste...
Shell Hardware Detection	Provides no...	Started	Automatic	Local Syste...
Smart Card	Manages ac...	Started	Automatic	Local Service
Smart Card Removal Policy	Allows the s...		Manual	Local Syste...
SNMP Trap	Receives tra...		Manual	Local Service
Software Protection	Enables the ...		Automatic (D...	Network S...

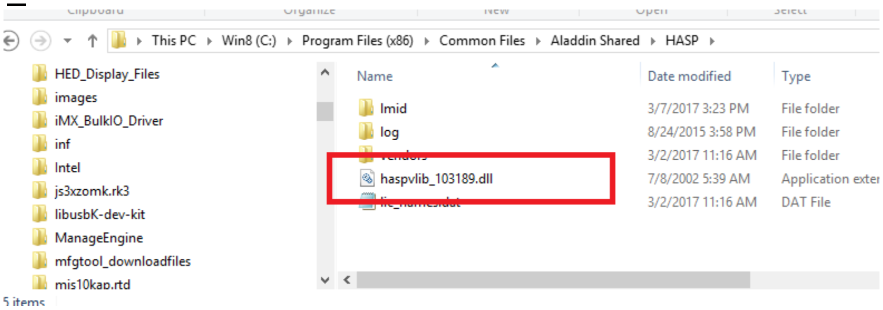
Sentinel LDK License Manager Status

Make sure this directory exists

```
C:\Program Files (x86)\Common Files\Aladdin Shared\HASP
```

The folder should have this file in it:

Haspvlib_103189.dll



This PC ▸ Win8 (C:) ▸ Program Files (x86) ▸ Common Files ▸ Aladdin Shared ▸ HASP ▸

Name	Date modified	Type
lmid	3/7/2017 3:23 PM	File folder
log	8/24/2015 3:58 PM	File folder
vendors	3/2/2017 11:16 AM	File folder
haspvlib_103189.dll	7/8/2002 5:39 AM	Application exter
lic_names.dat	3/2/2017 11:16 AM	DAT File

Haspvlib_102189 File

If either of these conditions exist, the license install program should be run to see if it can be installed:

The user should open the Command Prompt (admin) - must be admin and go to directory:
`C:\Users\Public\Orchestra\Orchestra 2.4.22\Sentinel LDK`

Run this program:

C:\haspdinst.exe /i

And see if it installs OK – no errors running program and the above two conditions exist.

If this does still not work, then HED® has to send the user the haspvlib_103189.dll and have the user manually copy the file in the directory.

Related:

Licensing problem 341

14.3 Linux Downloader Issues

14.3.1 RNDIS Error Occurs During Programming

Problem: RNDIS Error Occurs During Programming

Environment

- Windows 10
- Linux Device Downloader 1.1.3
- Between each programming attempt, power was cycled

Solution

1. Open Computer Management Device Viewer and search for Linux USB Ehternet/RNDIS Gadget.
2. If the device shows the warning icon, the driver needs to be reinstalled.

Steps to Resolve

1. Attempt to program:
2. Verified USB wiring
3. Pin 16 was grounded
4. Changed pin to floating for PC programming
5. Attempt to program: (RNDIS error occurs)
6. Open device manager, locate "Linux USB Ethernet/RNDIS Gadget"
7. Found under "Network Devices" with a warning indicator
8. Attempted to manually install drivers:
9. Manually locate file: C:\Users\Public\LinuxDeviceDownloader\LinuxDeviceDownloader\Drivers\USB_Network_wifi_win8_10
10. Device manager responded with: the best drivers were already installed
11. Right click, remove with "Delete the driver software for this device" checked
12. Attempt to program: (RNDIS error occurs)
13. Open device manager, locate "Linux USB Ethernet/RNDIS Gadget"
14. Found under "Other Devices" with a warning indicator

15. Attempted to manually install drivers:
16. Manually locate file: C:\Users\Public\LinuxDeviceDownloader\LinuxDeviceDownloader\Drivers\USB_Network_wifi_win8_10
17. Driver successfully installed
18. - Attempt to program: Success