SIEMENS

APPLICATION REFERENCE MANUAL

WCCT

JULY 2017

DOCUMENT NO. SIG-00-14-01
VERSION C

Siemens Industry, Inc. Rail Automation
9568 Archibald Ave., Suite 100,
Rancho Cucamonga, California 91730
1-800-793-SAFE

PRINTED IN THE U.S.A.

PROPRIETARY INFORMATION

TRANSLATIONS

The manuals and product information of Siemens Industry, Inc. are intended to be produced and read in English. Any translation of the manuals and product information are unofficial and can be imprecise and inaccurate in whole or in part. Siemens Industry, Inc. does not warrant the accuracy, reliability, or timeliness of any information contained in any translation of manual or product information from its original official released version in English and shall not be liable for any losses caused by such reliance on the accuracy, reliability, or timeliness of such information. Any person or entity that relies on translated information does so at his or her own risk.

WARRANTY INFORMATION

Siemens Industry, Inc. warranty policy is as stated in the current Terms and Conditions of Sale document. Warranty adjustments will not be allowed for products or components which have been subjected to abuse, alteration, improper handling or installation, or which have not been operated in accordance with Seller's instructions. Alteration or removal of any serial number or identification mark voids the warranty.

SALES AND SERVICE LOCATIONS

Technical assistance and sales information on Siemens Industry, Inc. products may be obtained at the following locations:

SIEMENS INDUSTRY, INC. RAIL AUTOMATION
2400 NELSON MILLER PARKWAY
LOUISVILLE, KENTUCKY 40223
TELEPHONE: (502) 618-8800
FAX: (502) 618-8810
SALES & SERVICE: (800) 626-2710
WEB SITE:www.siemens.com/rail-automation

SIEMENS INDUSTRY, INC.RAIL AUTOMATION
939 S. MAIN STREET
MARION, KENTUCKY 42064
TELEPHONE: (270) 918-7800
CUSTOMER SERVICE: (800) 626-2710
TECHNICAL SUPPORT: (800) 793-7233
FAX:(270) 918-7830

DOCUMENT HISTORY

Version	Release Date	Sections Changed	Details of Change
1			Rdb draft
A	OCT 2015	ALL	Initial Release
B	JULY 2016	ALL	**Global:** Change Wayside Controller to WayConneX Add WayTraX Module Add PSO Module **Glossary:** Add WayTraX **Section 1** No Change **Section 2** Pg 2-3 Sec 2.1 update Fig 2-3 Pg 2-4 Sec 2.2 update Fig 2-4 and Fig 2-5 Pg 2-5 Sec 2.2 update Fig 2-6 and Fig 2-7 Pg 2-6 Sec 2.2 update Fig 2-8 Pg 2-7 Sec 2.3 update Fig 2-10 **Section 3** Pg 3-22 Sec 3.6 WayTraX Editor add Section Pg 3-43 Sec 3.12 PSO Module Editor add Section Pg 3-49 Sec 3.13 BVPI8 Module Editor add Section Pg 3-51 Sec 3.14 BVPI10 Module Editor add Section **Section 4** Pg 4-16 Sec 4.2.9 add CTIO, LDIO, and LDIOST **Section 5** Pg 5-24 Sec 5.1.2 correct data Pg 5-26 Sec 5.2.2 correct data Sec 5.3.2 correct VPI data Pg 5-27 Sec 5.4 VPI Module add Table 5-3 Sec 5.4.2 correct data Pg 5-30 Sec 5.8 WayTraX Module add Section Pg 5-31 Sec 5.9 PSO Module add Section Pg 5-33 Sec 5.10 BVPI Module add Section Pg 5-34 Sec 5.11 BVPI10 Module add Section Pg 5-39 Sec 5.12.6 LDIO: Long Distance Coded Track Channels add Section Pg 5-41 Sec 5.12.7 LDIOST Long Distance Coded Track Send Twice Channels add Section Pg 5-42 PSOTX: PSO Transmitter Channel add Section

			Pg 5-43 PSORX: Receiver Channel add Section Pg 5-44 BVPI: Bipolar VPI Channel add Section **Section 6** Pg 6-2 Sec 6.2.1 update Fig 6-2 Pg 6-4 Sec 6.2.1 update Fig 6-4 Pg 6-5 Sec 6.2.2 update Fig 6-5 **Section 7** Pg 7-13 Sec 7.7.2 WayTraX Module add Section **Appendix A** No Change **Appendix B** No Change **Appendix C** No Change
C		ALL	Updates from RDB for new WCCT and section reorganization.

Table of Contents

Section **Title** **Page**

List of Figures

Section **Title** **Page**

List of Tables

Section **Title** **Page**

NOTES, CAUTIONS, AND WARNINGS

Throughout this manual, notes, cautions, and warnings are frequently used to direct the reader's attention to specific information. Use of the three terms is defined as follows:

WARNING

INDICATES A POTENTIALLY HAZARDOUS SITUATION WHICH, IF NOT AVOIDED, COULD RESULT IN DEATH OR SERIOUS INJURY. WARNINGS ALWAYS TAKE PRECEDENEC OVER NOTES, CAUTIONS, AND ALL OTHER INFORMATION.

CAUTION

REFERS TO PROPER PROCEDURES OR PRACTICES WHICH IF NOT STRICTLY OBSERVED, COULD RESULT IN A POTENTIALLY HAZARDOUS SITUATION AND/OR POSSIBLE DAMAGE TO EQUIPMENT. CAUTIONS TAKE PRECEDENCE OVER NOTES AND ALL OTHER INFORMATION, EXCEPT WARNINGS.

NOTE

Generally used to highlight certain information relating to the topic under discussion.

If there are any questions, contact Siemens Industry, Inc. Application Engineering

ELECTROSTATIC DISCHARGE (ESD) PRECAUTIONS

Static electricity can damage electronic circuitry, particularly low voltage components such as the integrated circuits commonly used throughout the electronics industry. Therefore, procedures have been adopted industry-wide which make it possible to avoid the sometimes invisible damage caused by electrostatic discharge (ESD) during the handling, shipping, and storage of electronic modules and components. Siemens Industry, Inc. has instituted these practices at its manufacturing facility and encourages its customers to adopt them as well to lessen the likelihood of equipment damage in the field due to ESD. Some of the basic protective practices include the following:

- Ground yourself before touching card cages, assemblies, modules, or components.
- Remove power from card cages and assemblies before removing or installing modules.
- Remove circuit boards (modules) from card cages by the ejector lever only. If an ejector lever is not provided, grasp the edge of the circuit board but avoid touching circuit traces or components.
- Handle circuit boards by the edges only.
- Never physically touch circuit board or connector contact fingers or allow these fingers to come in contact with an insulator (e.g., plastic, rubber, etc.).
- When not in use, place circuit boards in approved static-shielding bags, contact fingers first. Remove circuit boards from static-shielding bags by grasping the ejector lever or the edge of the board only. Each bag should include a caution label on the outside indicating static-sensitive contents.
- Cover workbench surfaces used for repair of electronic equipment with static dissipative workbench matting.
- Use integrated circuit extractor/inserter tools designed to remove and install electrostatic-sensitive integrated circuit devices such as PROM's (OK Industries, Inc., Model EX-2 Extractor and Model MOS-40 Inserter (or equivalent) are highly recommended).
- Utilize only anti-static cushioning material in equipment shipping and storage containers.

For information concerning ESD material applications, please contact the Technical Support Staff at 1-800-793-7233. ESD Awareness Classes and additional ESD product information are also available through the Technical Support Staff.

GLOSSARY

TERM	DESCRIPTION
AAR:	Association of American Railroads – An organization that establishes uniformity and standardization among different railroad systems.
AREMA:	American Railroad Equipment Manufacturing Association – An organization that supersedes AAR.
Aspect:	(Signal Aspect) The name given to a signal aspect used in PTC interface.
ATCS:	Advanced Train Control System – A set of standards compiled by the AAR for controlling all aspects of train operation.
ECD:	External Configuration Device – A serial EEPROM (Flash Memory) device mounted inside the chassis of the GEO unit. The ECD is used to store site-specific configuration data (MCF, SIN, UCN, and card parameters) for the CPU.
GEO Module:	GEO PCB I/O card. Modules installed in GEO Wayside Units and provide I/O channels to control GEO hardware.
GEO Wayside Unit:	GEO equipment used at a site location to operate railroad lights, switches or transmits information to other locations.
Group:	A sub-header under the Line entry, to organize the information created by the WCCT. The Group location is where the Chassis, NVLP and MCF's, of a track layout are stored. The Group entry has a number, name, state location and milepost that can be associated with ATCS addressing.
I/O Module:	Circuit cards contained within the WayConneX or GEO Wayside Unit. These modules pass input and output information to tracks, signals, and other railroad equipment.
Line:	A sub-header under Railroad to organize the information created by the WCCT. The Line entry has a number, name and description that can be associated with ATCS addressing.
MEF:	Module Executable File – The executive software running in the CPU. The user can download the MEF through the DTU port to update the software.
MCF:	Module Configuration File – The site-specific configuration information created by the WCCT and downloaded into the ECD via the CPU II+ and Diagnostic Terminal Utility (DT), or CPU III WebUI.
Menu Path:	The logical path from the WCCT menu bar to appropriate sub menus for completing a task.
PTC:	Positive Train Control
PTC UCN:	The Unique Check Number associated with the PTC logic in the MCF
Railroad:	The main header for the information created by the WCCT. The Railroad has a number, mnemonic and name that can be associated with ATCS addressing.

TERM	DESCRIPTION
SIN:	Site Identification Number – The 12-digit ATCS address for the WayConneX equipment downloaded through the Diagnostic serial port on the CPU and stored in the ECD. The SIN has the form 7.RRR.LLL.GGG.SS stored in binary coded decimal, with each digit in one nibble. The digit 0 is represented by "A" and 0 is used as a null byte.
Site Location:	The location where WayConneX unit is installed, usually within close proximity of the signal or switch that the GEO unit is controlling.
UCN:	Unique Check Number – A 32-bit CRC calculated over the MCF and stored in the ECD to detect file corruption. The UCN is created by the WCCT and entered into the CPU to verify proper configuration.
VPI:	Vital Parallel Input – A module input circuit the function of which affects the safety of train operation.
VRO:	Vital Relay Output – A module output circuit the function of which affects the safety of train operation.
WayTraX:	Microtrax Compatible Coded Track Module
WCCT:	WayConneX Configuration Tool

SECTION 1 GENERAL DESCRIPTION

1.0 INTRODUCTION

The Siemens WCCT is a Windows® based computer application used to create application programs for Siemens WayConneX and Boolean based applications for existing GEO™ systems.

This manual describes how this tool can be used to create application programs for WayConneX and GEO™ using traditional Boolean logic.

Section 1 in this manual will describe the part specification and requirements, Section 2 will walk the user through program setup, Section 3 will give a detailed explanation on editing each module in the chassis, Section 4 will give a quick overview of creating an application, then Sections 5 and 6 will contain more detailed information on builds, validation testing, and reports.

1.1 COMPUTER SYSTEM REQUIREMENTS

The list below indicates the minimum computer requirements for the WCCT Programming Application.

- VGA monitor
- 1.5GHz processor or faster
- 2GB of RAM
- 500MB Hard disk space
- CD-ROM drive for application installation (if installed via CD)
- Microsoft® Windows® Operating System, Windows 7 or 10

1.2 ORDERING INFORMATION

Siemens part numbers for the WCCT Application:

Part Number	Description
Z224-9VD24-A010	WCCT

1.3 MCF VERIFICATION

Verification and Validation (V&V) of the MCF, as with all software, requires code level and system level testing. The WCCT includes tools that facilitate lower level V&V activities and Siemens recommends a V&V approach, utilizing a combination of Simulation and Decompilation. A system level approach, such as rack testing, is a required V&V activity, based on the system safety analysis.

WARNING

BEFORE BEING PLACED IN LIVE-SERVICE, THE MCF GENERATED USING THE WCCT MUST BE VERIFIED AND VALIDATED TO ENSURE SAFETY.

1.3.1 Simulation (Recommended)

Interactive simulation may be used to test the logical operation of the application. All logical combinations of input states may be explicitly commanded and all resulting output states may be inspected using relay logic simulation to ensure expected operation of configuration data and vital logic.

1.3.2 Validation (Recommended)

Module Configuration Files are compiled from data entered by the user into the graphical user interfaces and stored as source files. The Validation step is used to re-generate the original source files from compiled Module Configuration Files. The tool compares the original project files with the decompiled project files and produces a difference report that indicates whether any problems were found.

1.3.3 System Test (e.g. Rack Test) (Required)

The system test verifies the resultant MCF against actual hardware and executive software. As part of the system test, the Program Report obtained from the DT / OCE and target GEO/WayConneX system must be generated and reviewed to ensure all configuration parameters and default settings are accurately represented.

The person performing the system test must be familiar with the application logic and the software application used.

SECTION 2 PROGRAM SETUP AND CONFIGURATION

2.0 WCCT FUNCTIONS

2.1 PROGRAM INSTALLATION

1. Insert WCCT installation disk into the computer's CD ROM Drive.
2. The installation startup screen should automatically appear. If the installation screen does not appear within a short time, browse for the CD using My Computer or Windows Explorer and locate the file named Setup.exe. Double-click the Setup file to begin the installation process.
3. Follow the on-screen instructions to install the applications.
4. Restart the computer to complete the installation.

2.2 MANAGING PROJECTS

The WCCT allows the user to open one project at a time.

2.2.1 Setting File Workspace

By default, the tool will store new projects in c:\Siemens\WCCT\Workspace.

If you want to use a different location for all project first set the new workspace using **File>Workspace**.

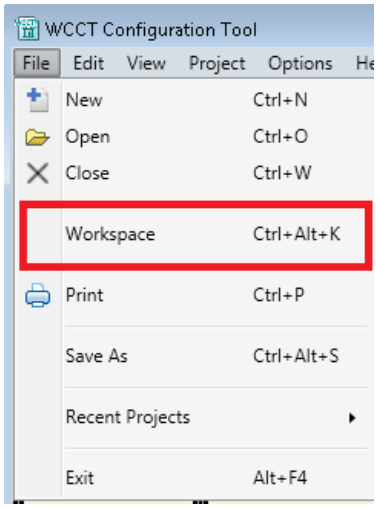


Figure 2-1 Setting Workspace

This will bring up a window where you can set the new workspace.

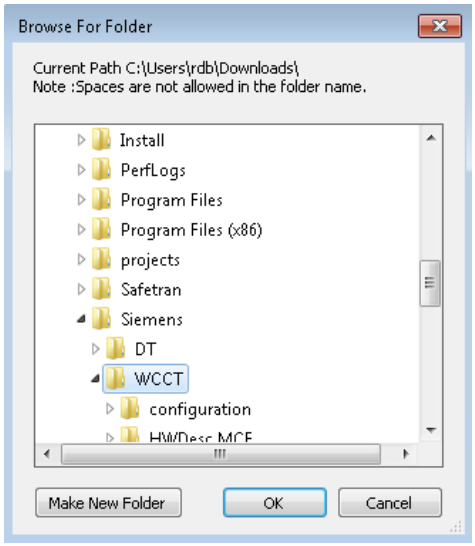


Figure 2-2 Select Workspace

Make sure that there are no spaces in the path that has been selected. If there are spaces the tool will give the following error message:

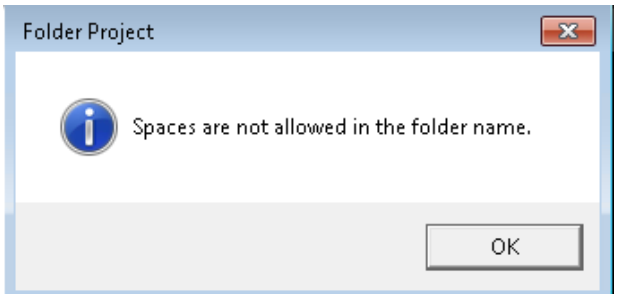


Figure 2-3 Spaces in Path Error

2.2.2 Creating a New Project

To create a new project, either select **File>New>Project**, select the **New** icon or use the **Ctrl N** shortcut.

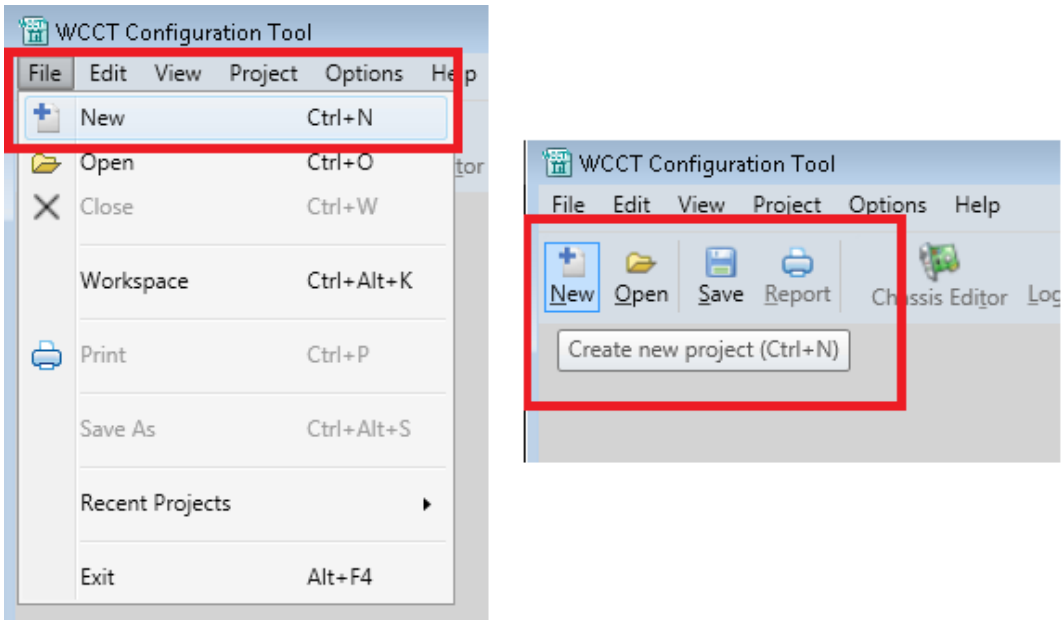


Figure 2-4 Creating a New Project

This brings up the **New Project Window**. From this, type in the project name (alphanumeric or underscore, 13 characters long, cannot include spaces or other punctuation).

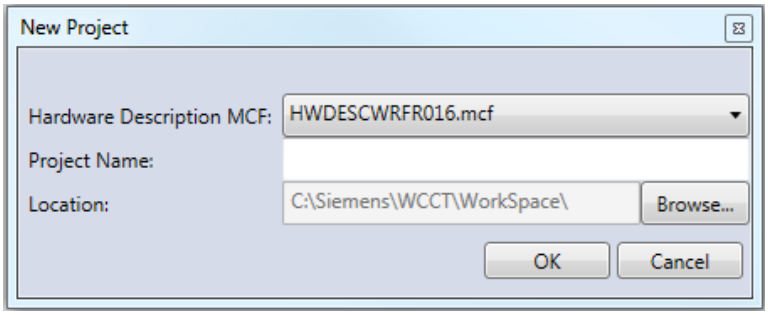


Figure 2-5 New Project Window

To change the location where you want to keep your projects, select the **Browse** button next the Location. The default Location is show above.

The Hardware Description MCF is used to define module properties used by the tool. Generally, you will use the latest version available.

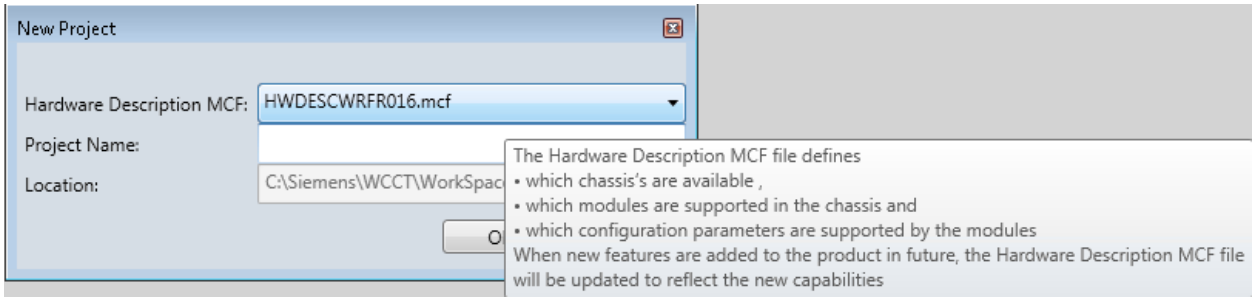


Figure 2-6 Hardware Description MCF

The new project will be created and will show up in the selected Location. The new status bar at the bottom of the main window will show the current project.

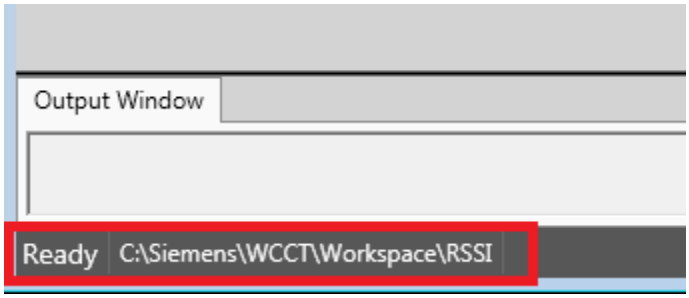


Figure 2-7 Status Bar

The project will open with the chassis editor showing the default 8Slot chassis with the VLP3 in slot 1.

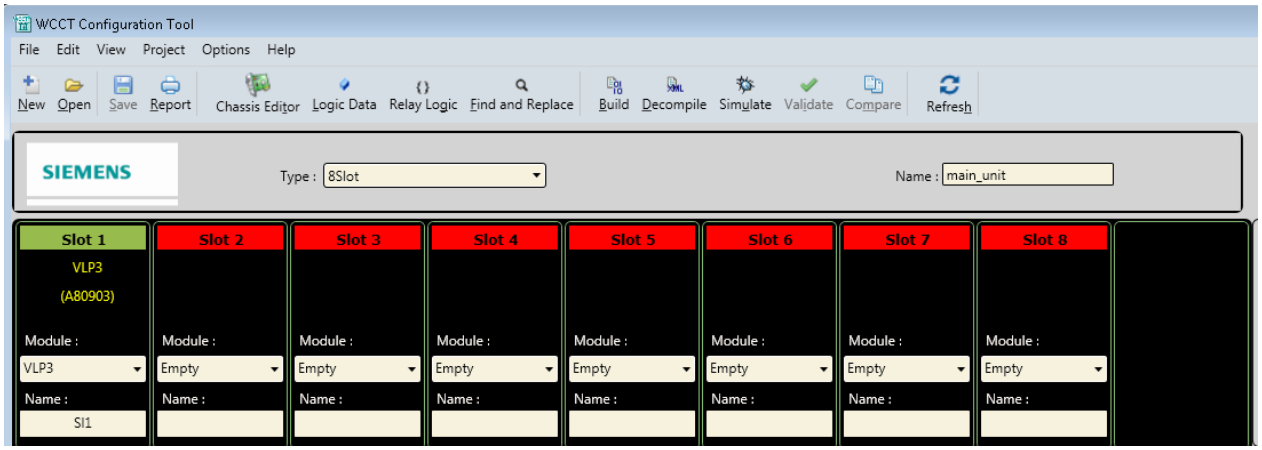


Figure 2-8 Chassis Editor Opening Screen

2.2.3 Opening a Project

To open a project, select **File>Open** or the **Open** icon or the **Ctrl O** shortcut.

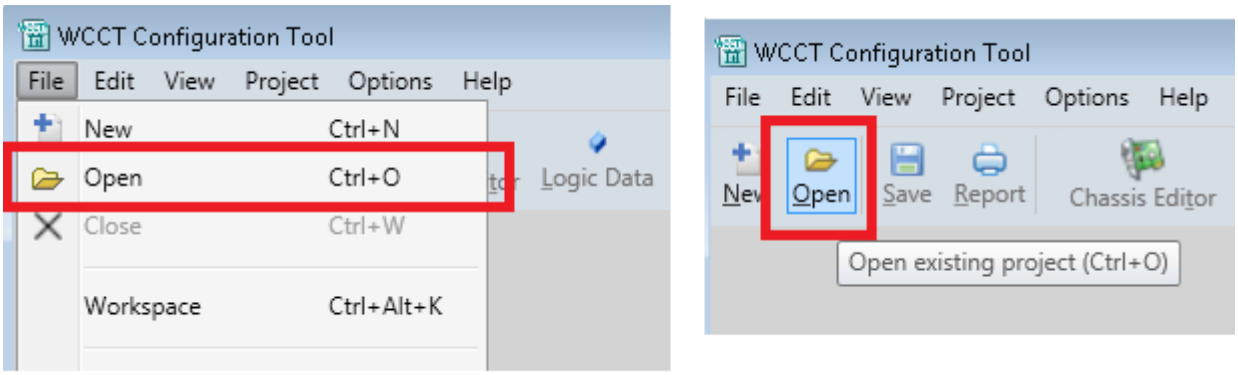


Figure 2-9 Select Open Project

This opens the **Open** window from which the user can select the folder and .xml file for the project. Double-click on the file or select **Open.**

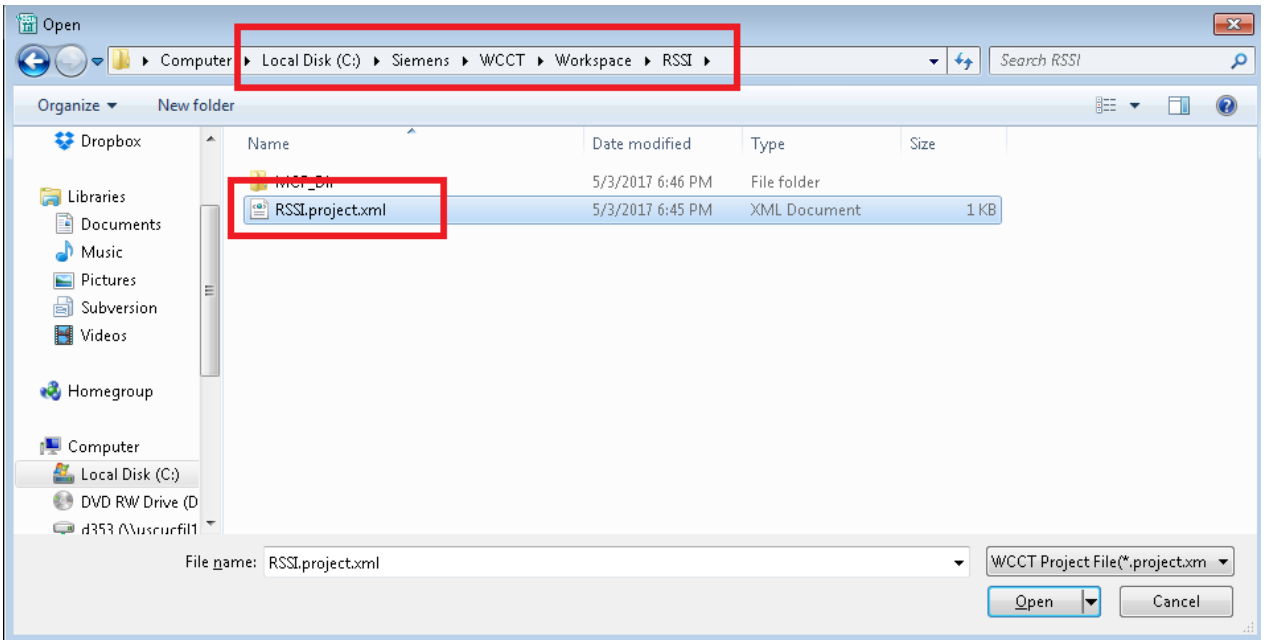


Figure 2-10 Opening a Project

This will open the project at the chassis editor. To open projects created with the older, Eclipse-based WCCT (version 1.3.6 and earlier) see Appendix E.

2.2.4 Closing a Project

Select the **File>Close** or the **Ctrl W** shortcut to close the current project.

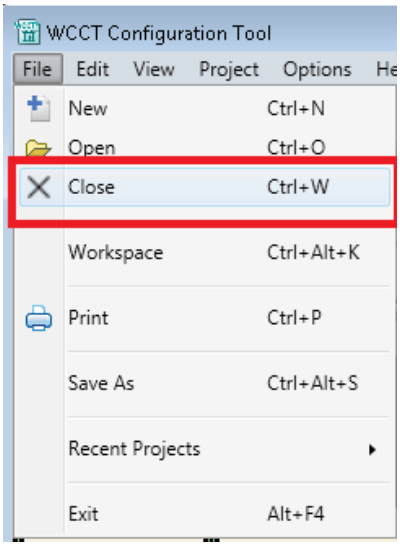


Figure 2-11 Closing Projects

2.2.5 Copying a Project

To copy a project, open the project to be copied, then select **File>Save As**.

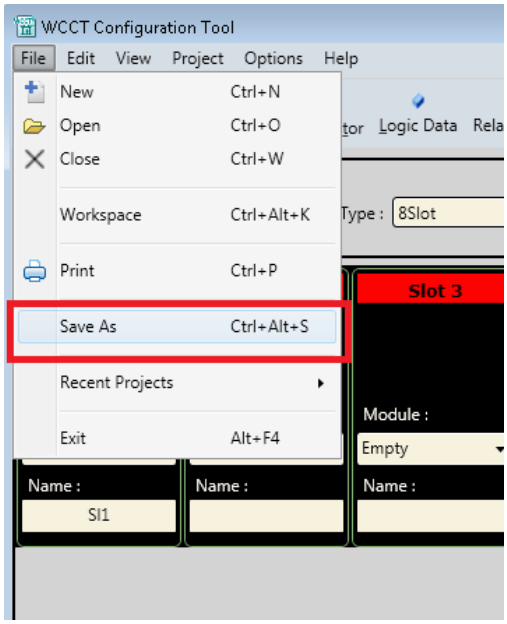


Figure 2-12 Save As Option

This will open the **Save As** window. Enter the new project name and, if required, select a different location using the Browse button. A copy of the project will be made and the WCCT will open the new project.

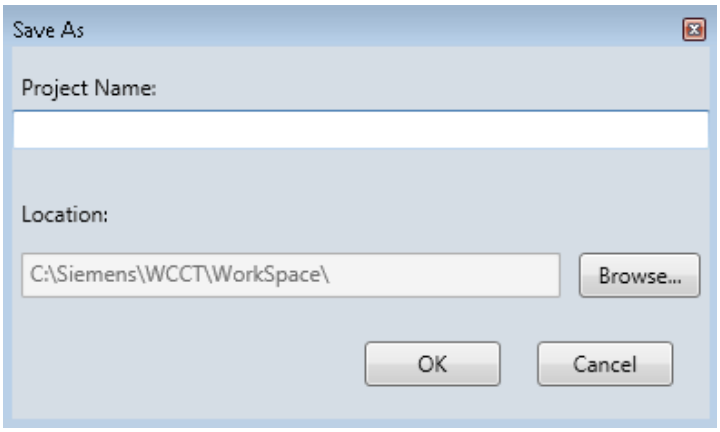


Figure 2-13 Save As Window

Note that doing the **Save As** only copies the source files for the project (.project xml, .gc and .gl files). It does not copy the output files under the MCF_Dir folder (i.e. the MCF, or the reports).

2.2.6 Deleting a Project

The tool does not provide a way to delete a project. To delete a project, find the project's location in Windows Explorer and delete it from there, for example to delete project test2, right-click on the file name and select **Delete**.

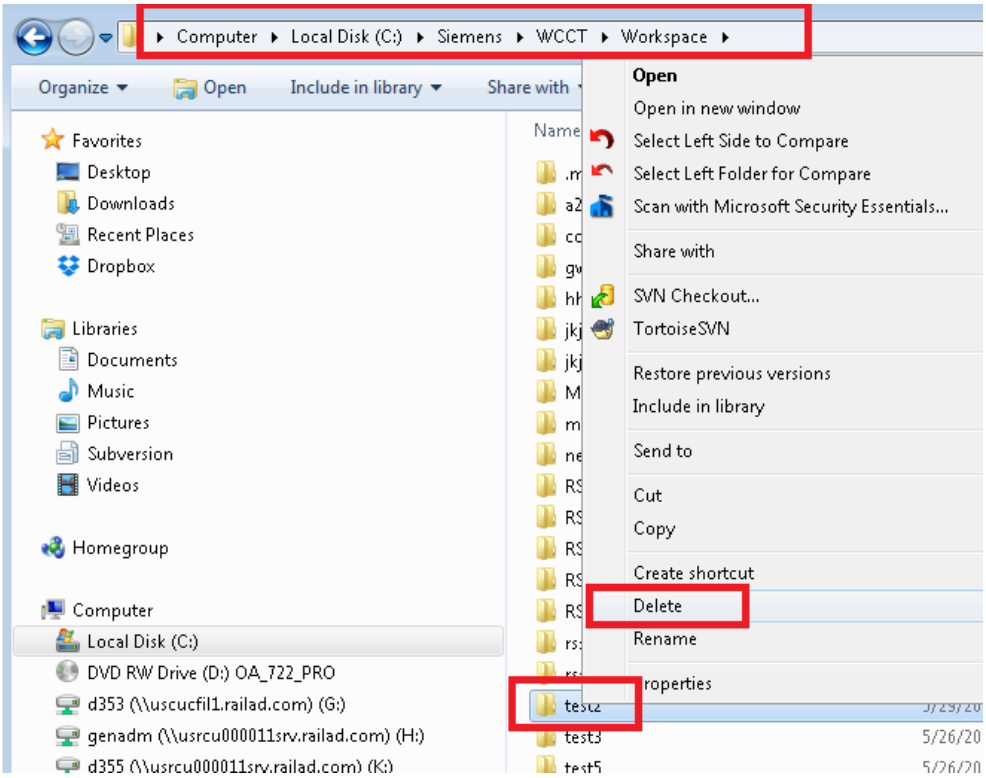


Figure 2-14 Delete Project from Windows

2.2.7 Renaming a Project

To rename a project do a **Save As** to save the project with a new name, then delete the old files using Windows.

2.2.8 Moving Projects

If you want to move a project to a different location, you can just copy the project folder from one place to another in Windows. Ensure that there are no spaces in the file path or else the tool will give the following error when you try and open it:

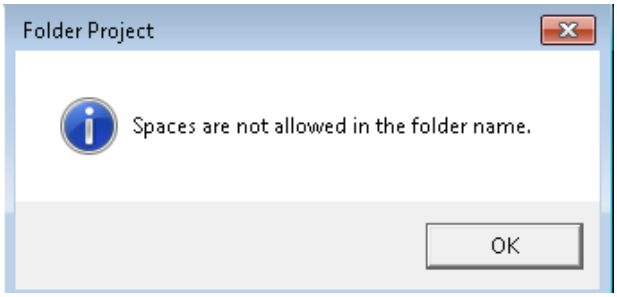


Figure 2-15 Space not Allowed Error

2.3 ENTERING DATA IN WCCT

The WCCT provides a graphical user interface in which to enter data.

Use the Chassis Editor (see section 3.3) to:

- Select the chassis
- Select the modules in the chassis
- Set the default configuration parameters for the module
- Set the names for input and output states used by the module

Use the Logic Data Editor (see section 3.17) to:

- Enter Interval Variables
- Enter Timers
- Enter configurable Property
- Define a menu system

Use the Relay Logic Editor (see section 3.19) to:

- Enter the relay logic

This page intentionally left blank

SECTION 3 EDITORS

3.0 EDITORS

3.1 SETTINGS EDITOR

The settings editor allows selecting options that apply to the WCCT for all projects. To open this editor, select the **Settings** submenu from the **Options** menu.

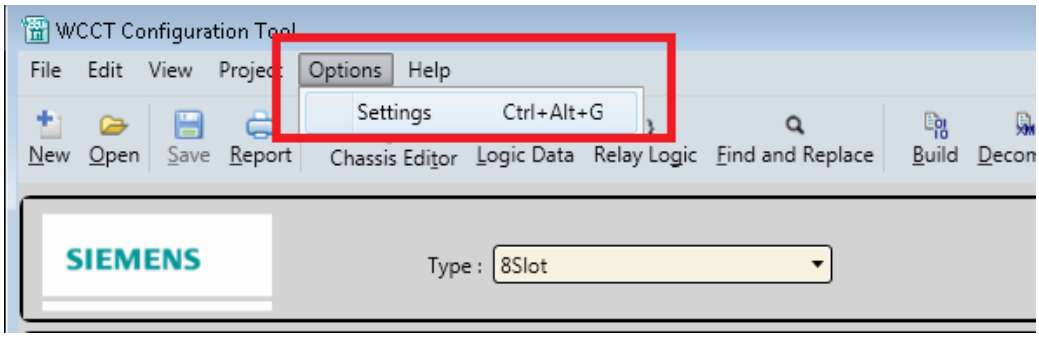


Figure 3-1 Selecting Settings

The settings editor is shown below.

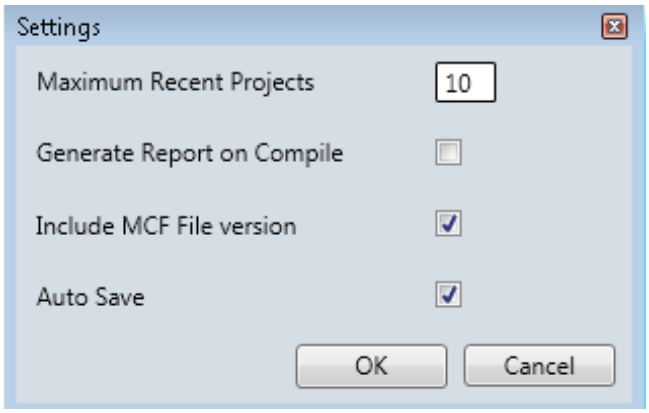


Figure 3-2 Settings Window

Maximum Recent Projects indicates how many recent projects are shown under **File>Recent Projects** menu. Note: the change doesn't take effect until the WCCT has been closed and re-opened.

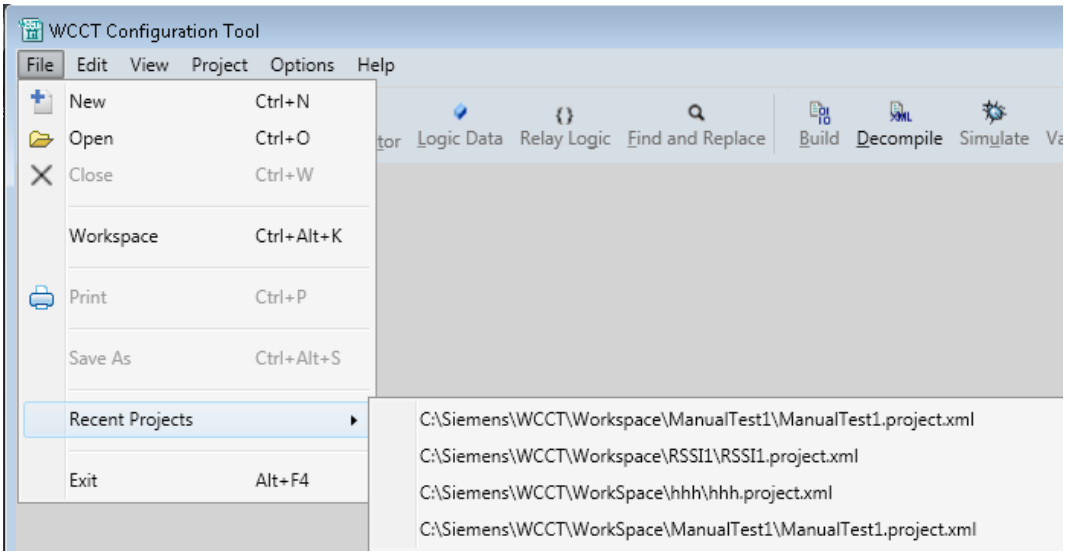


Figure 3-3 Recent Projects

When **Generate Reports on Compile** is checked, the WCCT will create an updated PDF for the project whenever the user builds the MCF. When this is not checked, the report is only generated when the user presses the **Report** button. When **Include MCF File Version** is checked, the MCF version set in the Project Properties window will be part of the file names for the build products (MCF, report, log file, txt file). For example, if the MCF version is 002, the file names will include the 002.

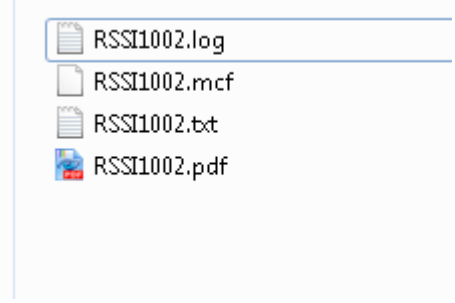


Figure 3-4 Output Files in MCF_Dir

If this is unchecked, the MCF Version is not included in the file name.

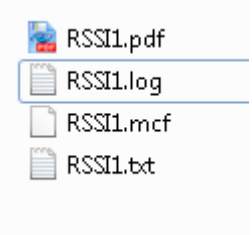


Figure 3-5 Output Files in MCF_Dir, with no version in the file name

When **Auto Save** is checked and the user makes changes to an equation in the Relay Logic Editor, and then selects to leave the equation, the WCCT will automatically save the changes to the last equation. When **Auto Save** is not checked and the user selects to leave an open equation, the WCCT will prompt the user to save changes before they go to the next equation.

3.2 PROJECT PROPERTY EDITOR

To open this editor, select the Project Properties submenu from the Project menu or use the **Ctrl K** shortcut.

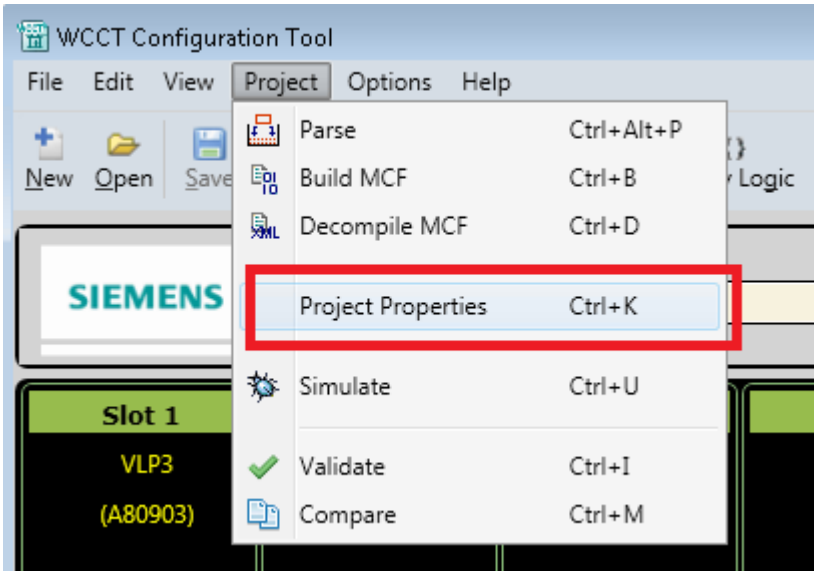


Figure 3-6 Selecting Project Properties

This window allows the user to:

- Change the HW Description MCF
- Change the MCF version
- Enter descriptive fields regarding the project
- Enter the ATCS Address for the system
- Enter revision information
- The HW Description MCF is used to define the capability of the modules used in the system. If a new module type is added, or the capability of a module changed, Siemens will release a new HWDESC MCF. If hardware is installed in the field without the new change, the user can elect to keep using the old version of the HWDESC to maintain compatibility.

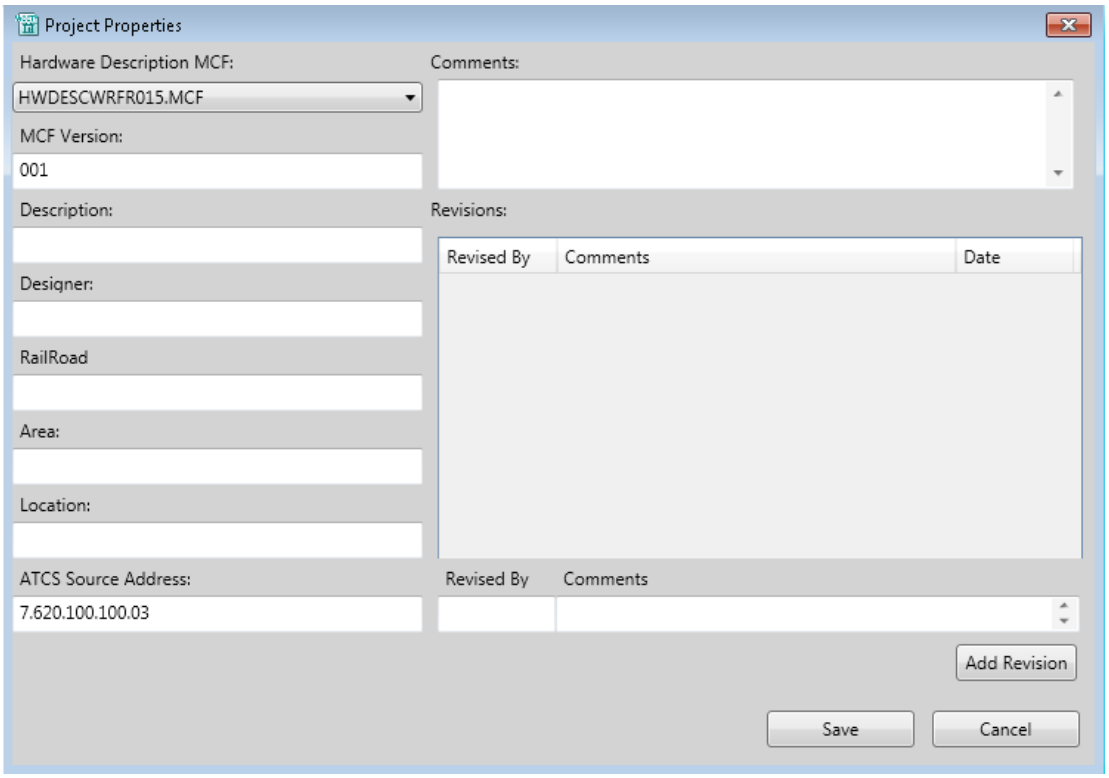


Figure 3-7 Project Property Editor

To add new revision information, enter data in the **Revised By** and associated **Comment** text box and press **Add Revision** button. The WCCT will add a date stamped entry in the **Revisions** text box.

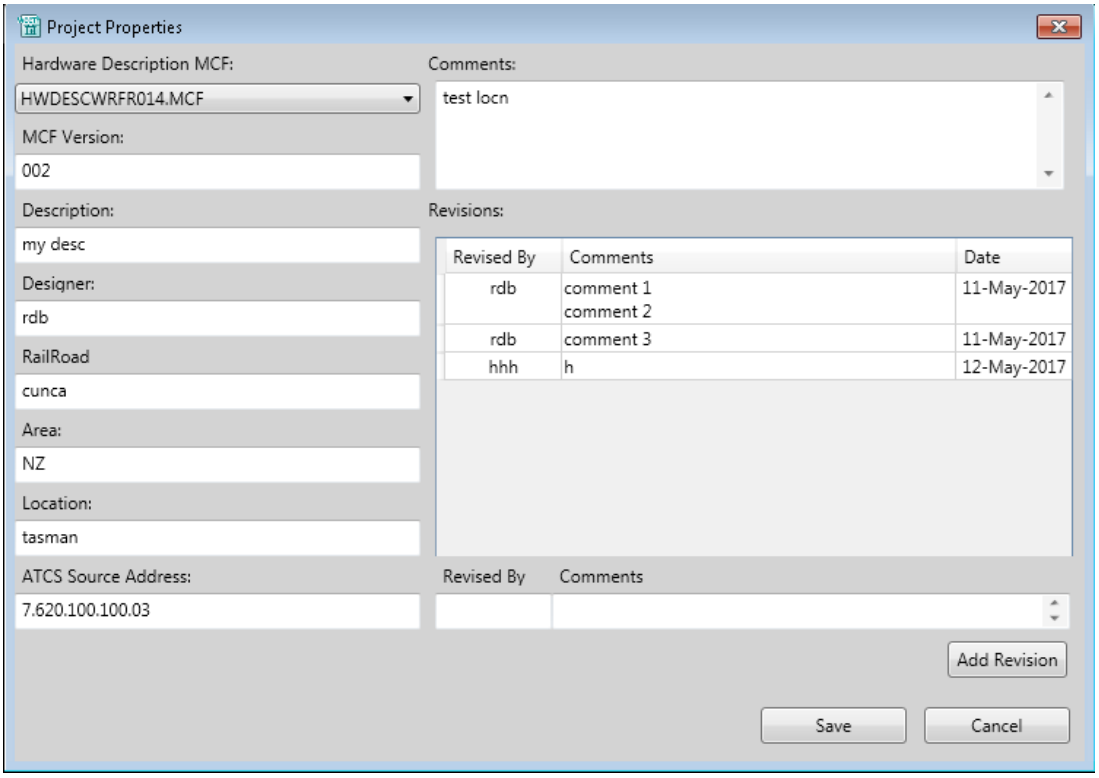


Figure 3-8 Project Property Editor - Adding Revisions

The ATCS Source address is the ATCS address of the WayConneX/GEO system. It has to be entered in the form 7.RRR.LLL.GGG.SS, where:

- RRR in range 1-999
- LLL in range 1-999
- GGG in range 1-998
- SS in range 03-98

The ATCS address is included in the UCN. The user has the opportunity to change the ATCS address later when they adjust the configuration settings using the OCE for specific sites in which the MCF is intended to be used.

3.3 CHASSIS EDITOR

The chassis editor allows selection of the chassis type and which modules are present in the chassis. The default chassis is the WayConnex 8Slot generic chassis.

Chassis	System	CPU required	Slot 1
8Slot	WayConneX	CPU3 (A80903)	VLP3
TwoTrackNE	WayConneX	CPU3 (A80903)	VLP3
TheTwoTrackExtended	GEO	CPU2+ (A80403)	VLP2
ThreeTrackShort	GEO	CPU2+ (A80403)	VLP2
ThreeTrackExtended	GEO	CPU2+ (A80403)	VLP2

The GEO chassis were initially designed to use the CPU II+ module (A80403) thus the only valid CPU choice in the Chassis Editor for these is the VLPI2. However, in practice, the CPUIII (A80903) can act as a drop in replacement to the CPUII+. In the field a CPUIII can be used in place of the CPUII+, however, its functionality is restricted to that of a CPUII+.

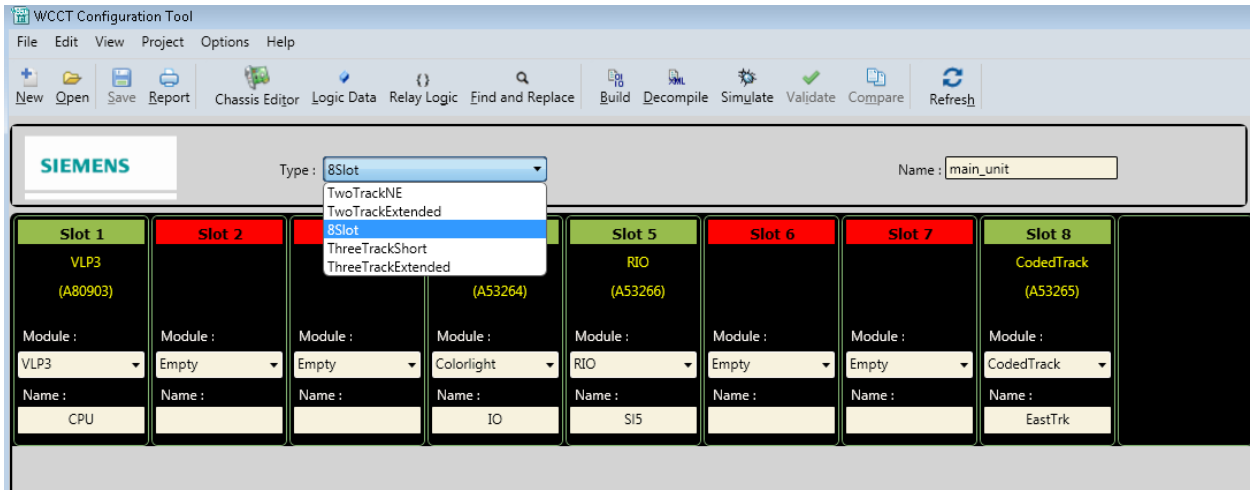


Figure 3-9 Selecting Chassis

In general, the chassis is selected once the project is created and does not change; however, in some cases, the user may want to change the chassis; for example, if a new project is created by copying another project and modifying it for a different application.

The WCCT allows the user to change the chassis on an existing project but in some cases this may result in inconsistencies. If the new chassis does not support a module in that particular slot, the WCCT will warn the user; for example, if slot 2 of 8Slot chassis has a Colorlight module in it and the user changes to a TwoTrackNE chassis, the following warning is shown:

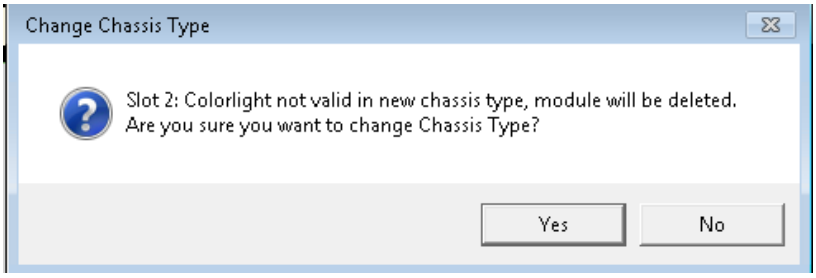


Figure 3-10 Deleting Module Confirmation Message

If the user accepts the change, the data for the invalid module is deleted.

If the user changes from a GEO chassis to a WayConnex type, the WCCT automatically changes the VLP2 to a VLP3. Since the VLP2 functionality is a subset of VLP3, there are no consistency issues here.

If the user changes from a WayConnex type Chassis to a GEO type chassis, the WCCT automatically changes the VLP3 to a VLP2. This may result in warning messages if the project has PTC defined, or Vital Comms channel using more bits than the VLP2 can handle.

The user may also want to change module types in an existing system. Since the majority of modules provide very different functionality, when you change one module to another, the module data from the old module will be deleted. However, there is one case, that of the Track module and Line Module, where there is significant commonality. Thus, if the user changes a track module to a line module, any common configuration parameter changes will be retained and the CTIO and Stk channel definitions will not be changed.

The WCCT allows the user to copy a module from one slot to another. Right-click on the module and select **Copy**, then select the slot to be copied to.

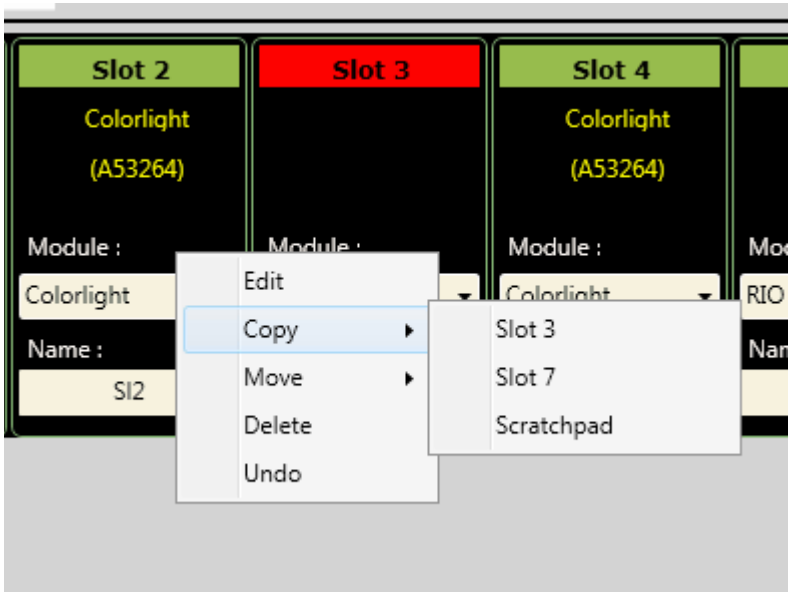


Figure 3-11 Copy Module

The module will appear with a red bar (i.e. unsaved): select the **Save** button (Ctrl S) to save the change.

Figure 3-12 Red Bar On Copied Module

When the copy is made, the module configuration properties channel and I/O states are copied. To avoid duplicating logic state names, all the logic states are prefixed with the new slot number as shown below. To edit the module, double-click and then change the logic state to the required names.

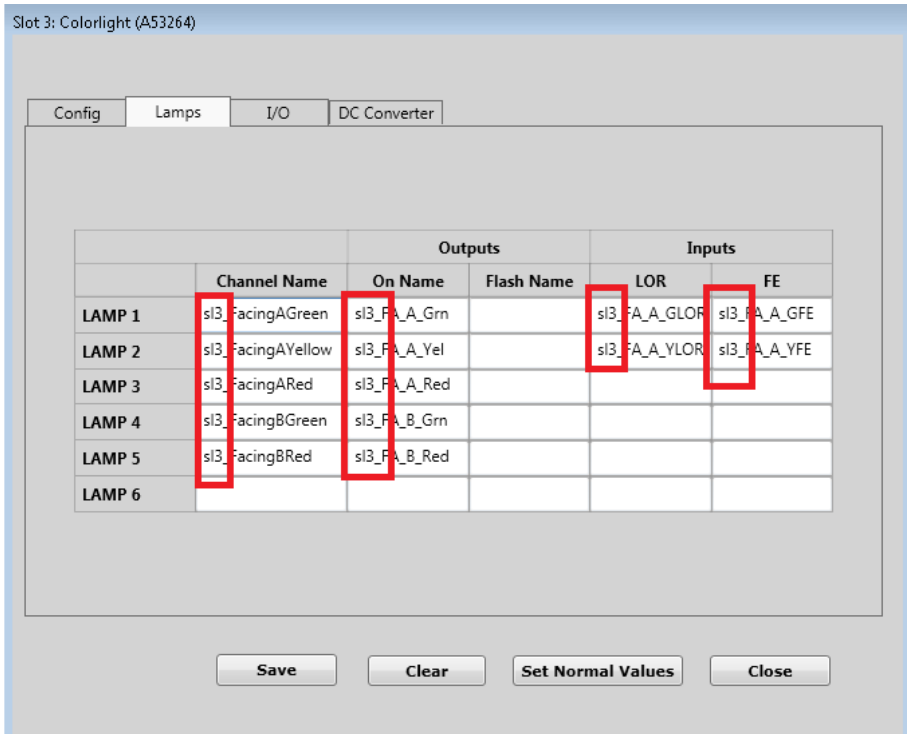


Figure 3-13 I/O Names after Copy

The WCCT allows the user to move a module from one slot to another. To do so, right-click on it and select **Move**, then the new slot.

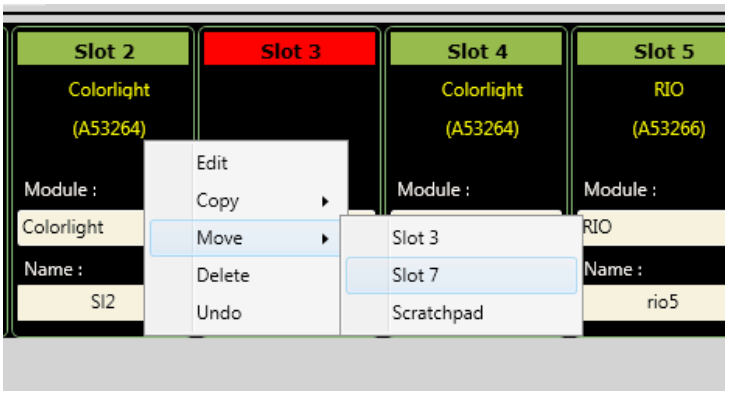


Figure 3-14 Move Module

When the module is moved, the previous slot is set to empty and the module moved to a new slot. If the current name of the module is still the default, e.g. SL2, then it will be changed to reflect the new slot. However, if the user has given the module a specific name, that name will be transferred to the new slot.

The Scratchpad slot is like a Cut/Paste buffer: a module can be copied into it temporarily, then moved/copied back to a real slot. For example, if all the slots on the current chassis are full and the user wants to swap modules around, he/she can first move one module to the Scratchpad; move the second module into the empty slot, then move the module from the Scratchpad.

Figure 3-15 Scratchpad

To delete a module, right-click and select the **Delete** option or set the slot to empty and press **Save**. To delete multiple modules, set them all to empty and press **Save**. The WCCT will ask for confirmation before the modules are deleted.

3.4 VLP3 MODULE EDITOR

The VLP3 module editor has four tabs

- Config
- Vital VComms
- NV Controller
- PTC Logic IF

3.4.1 Config

The Config tab allows the user to change the default configuration parameters for the VLP3 module. These parameters indicate whether an I/O module in a slot is enabled or not. If the slot is empty in the chassis, the enabled property will be set to No.

This feature would be used for an application where some modules may be optionally populated. Note that if a module is marked as not enabled, the CPU III will not try to establish communication with that module.

The status reported from an optional module, which is not populated in the chassis, will report its safe states, just as if the module was meant to be present, but is not plugged in. VPIs will report de-energized, but if there is an optional Colorlight module which will report back its lamp states as LOR/FE, it may be necessary to create a Property in the Logic Data Editor that can be used to mask these states.

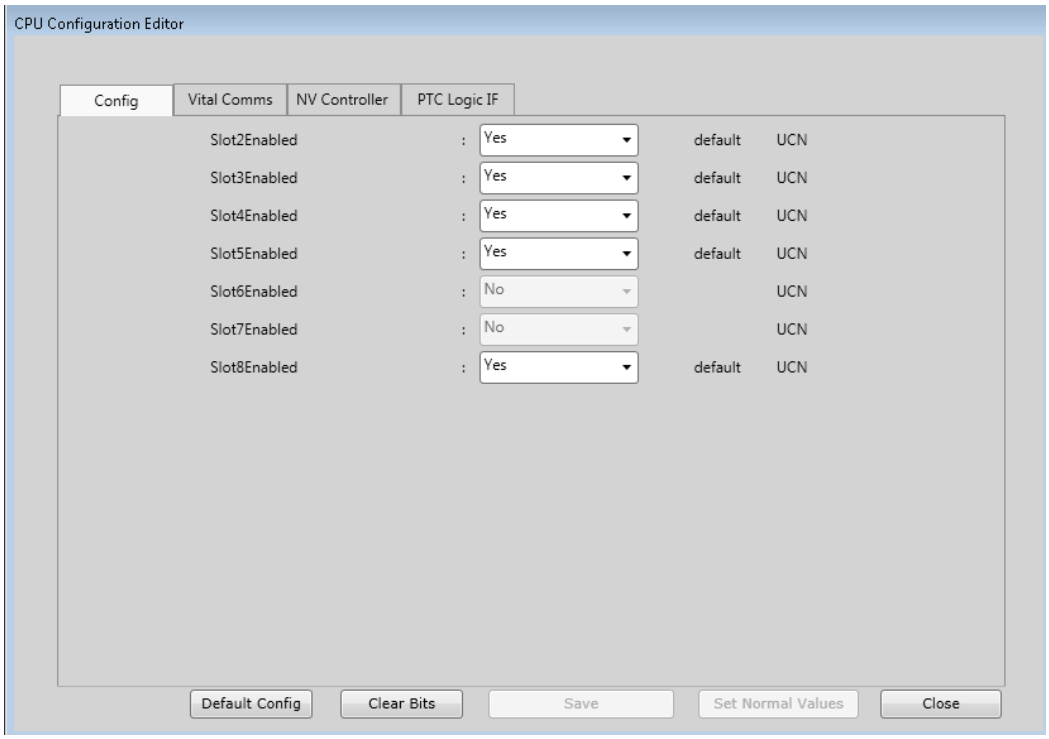


Figure 3-16 Slot Enabled Status Display

These parameters are all included in the UCN.

- **Default Config** button will set the values back to their default (Enabled = Yes).
- **Clear Bits** button does nothing on this screen
- **Set Normal Values** brings up the editor that allows the Normal Values to be set for CPU related I/O
- **Save** – Saves changes
- **Close** – Closes window

3.4.2 Vital Comms Editor

The Vital Comms Editor tab allows the user to define vital communication channels between GEO/WayConneX units. There are two tabs on the editor: Config and MsgBits.

3.4.2.1 Config

This tab allows the user to select default values for the Vital Comms channel configuration used in this MCF. Vital Comms links are ATCS message-based and thus the message will contain the ATCS address of this GEO/WayConneX unit as the source address, and the ATCS Address of the unit being communicated with as the destination address. The ATCS of the GEO/WayConneX is set in the Project Properties menu, see Section 3.2.

The ATCS of the neighboring unit can either be set directly by entering it in the Destination Address text box or by specifying the ATCS address of the remote unit relative to the ATCS address of the main unit via the RRR, LLL, GGG and SS Offset fields.

The general format for the full ATCS address is: 7.RRR.LLL.GGG.SS.DD

Where:

RRR – Railroad number (1-999)

LLL – Line number (1-999)

GGG – Group number (1-999)

SS – Subnode number (3-98)

DD – device number (2-90)

If the user updates the Destination Address, the WCCT will automatically recalculate and update the Offset fields. Similarly, if the user updates offset fields, the WCCT will automatically recalculate and update the Destination Address.

If more than one Vital Comms channel is used between the same Source and Destination addresses (i.e. if RRROffset, LLLOffset, GGGOffset and SSOffset are the same for more than one VComms channel) then the Local Device Numbers need to be made unique for each channel.

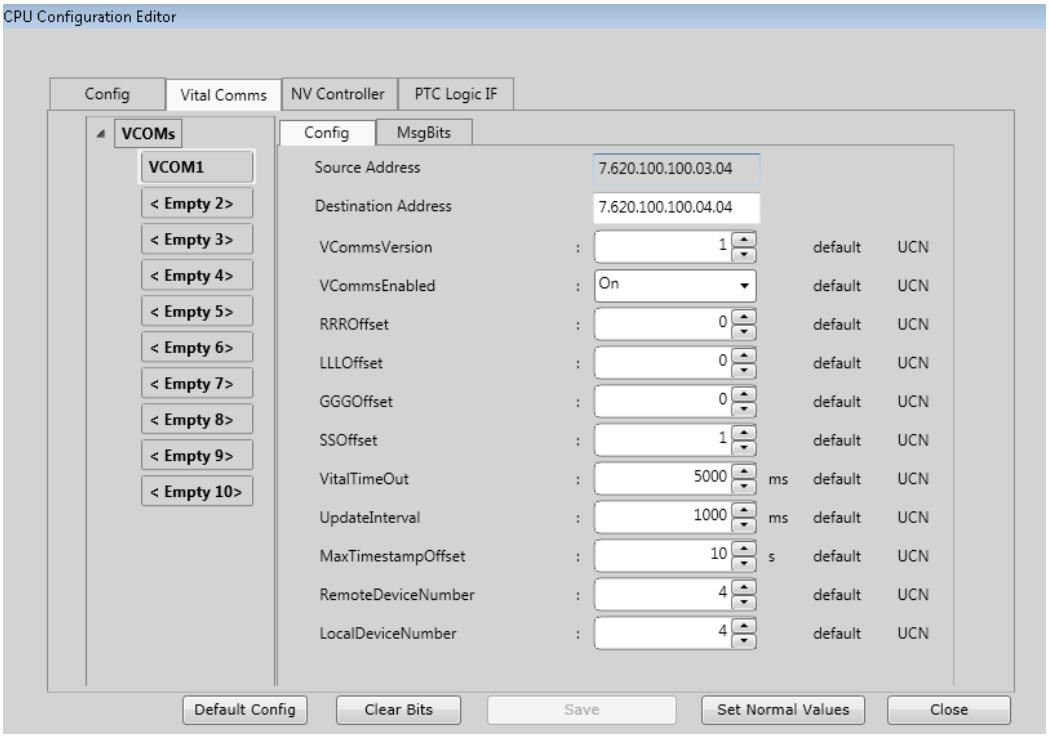


Figure 3-17 Vital Communications Configurations Parameters

Table 3-1 Vital Communications Parameters

Destination Address	Default is derived from ATCS address set in Project Properties with SS offset of 1	This is the ATCS address of the neighboring unit.
VCommsVersion	Default 1, Range 1-250	This is a version number for message data content for this channel. This should match the version number on the receiving end. The VComms version number can be used to ensure that if a change is made to the Vital message content for an installed location, that both sides of the link are updated in order for communication to be successfully established.
VCommsEnabled	Default On, Range On, Off	A Vital Comms link can be disabled by setting this to **Off**. This can be used in a case where the Vital Comms link is optional in the application.
RRROffset	Default 0, Range see Note below	This is the offset in the Railroad number part of the ATCS address from the ATCS address of this unit.
LLL Offset	Default 0, Range see Note below	This is the offset in the Line number part of the ATCS address from the ATCS address of this unit.
GGG Offset	Default 0, Range see Note below	This is the offset in the Group number part of the ATCS address from the ATCS address of this unit.
SS Offset	Default 0, Range see Note below	This is the offset in the Subnode number part of the ATCS address from the ATCS address of this unit.
VitalTimeOut	Default 5000 ms, Range 1,000-60,000 ms	If no valid ATCS message is received from the remote GEO/WayConneX for this time, then the link will go into a restrictive, out-of-session state, and all the received bits will report de-energized into the logic.
UpdateInterval	Default 1000 ms, Range 400-30,000 ms	Vital ATCS messages are sent out periodically at this time interval. It is recommended that the VitalTimeOut is at least three times the UpdateInterval + 200ms, so that several messages in a row won't be lost before the unit loses session.
MaxTimestamp Offset	Default 10 s, Range 5-30 s	This is used by the GEO/WayConneX to determine if a message is stale.
RemoteDevice Number	Default 4, Range 4-90	This is the Device number (DD) part of the ATCS address of the destination address. Each Vital Comms channel must have a unique Local and Remote Device number.
LocalDevice Number	Range 4-90	This is the Device number (DD) part of the ATCS address of the local address, each Vital Comms channel must have a unique Local and Remote Device number.

The full range of the RRR/LLL/GGG offset is -999 to +999, and SS -99 to +99; however, in practice, these are not used. The WCCT will not allow a value to be set for the offsets which results in an illegal ATCS address, thus, the range of these parameters depends upon the actual ATCS address set in the Project Properties window.

All these parameters are all included in the UCN.

Buttons:

- Default Config button will set the values back to their default values.
- Clear Bits button does nothing on this screen.
- Save – Saves changes.
- Set Normal Value: Bring up Normal Values window
- Close – Closes window.

3.4.2.2 MsgBits

The MsgBits Tab allows the user to define how many bits are used in the input and output Vital ATCS messages and the names of those bits.

To create a new Vital Comms channel, select the next empty channel and enter the number of IP and OP bits required, then enter names for the message bits.

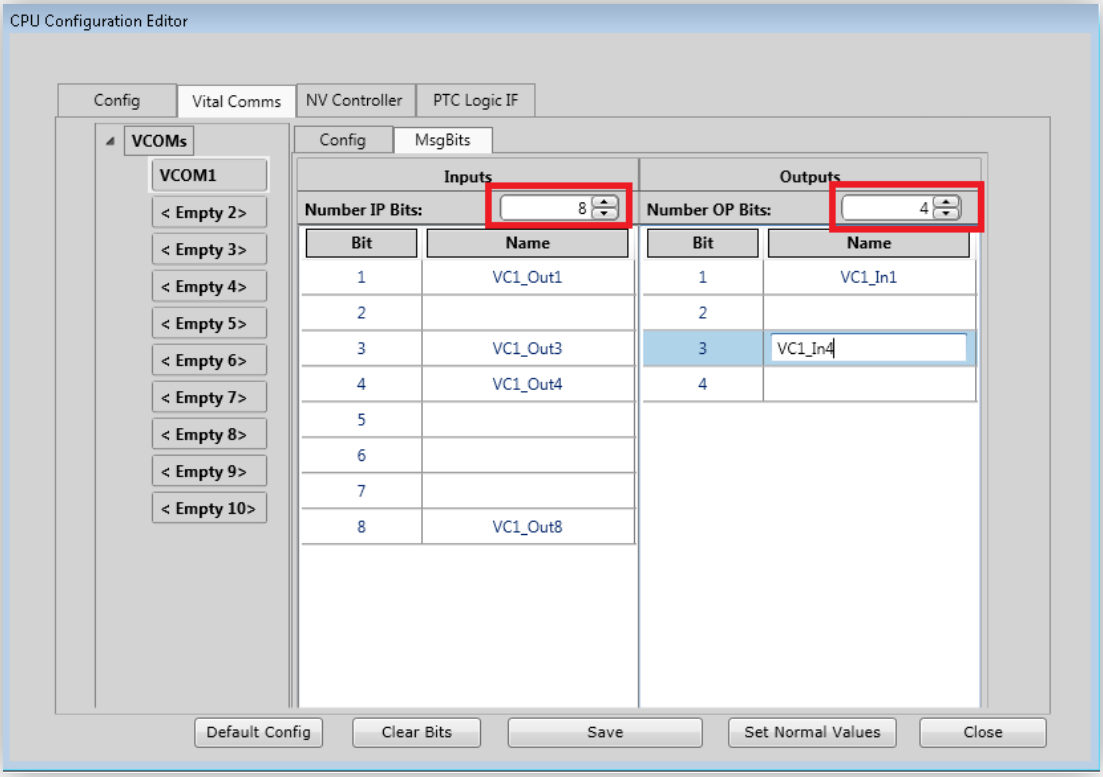


Figure 3-18 Select Number IP/OP bits in Vital Comms

For the VLP3, the maximum Number of OP bits is 255 and the maximum Number of IP bits is 255.

Not all the bits in the message need to be used, they can be left blank.

NOTE

The WCCT will check that the names entered on this screen are not duplicates of any other logic state names entered in the project, e.g. I/O module inputs and outputs and Boolean variables entered in Logic Data Editor. The user should write a Boolean equation to set the state of the output bits named here.

Buttons:

- **Default Config** will set the values on the Config screen back to their default values.
- **Clear Bits** will clear all the assigned bits on the Msg Bits screen.
- **Save** – Saves changes.
- **Set Normal Value** – Bring up Normal Values window
- **Close** – Closes window.

3.4.3 NV Controller Editor

The NV Controller Editor tab allows the user to define the interface to a non-vital controller such as the SEAR II. There are two tabs: Config and MsgBits.

3.4.3.1 Config

There are no configuration parameters applicable to this.

3.4.3.2 MsgBits

The MsgBits Tab allows the user to define how many bits are used in the input (control) and output (indication) ATCS messages to the NV Controller, and names of these bits.

Enter the number of IP and OP bits required. For the VLP3, the maximum number of OP bits is 255 and the maximum Number of IP bits is 255. Then enter the names for the required Control and Indication bits.

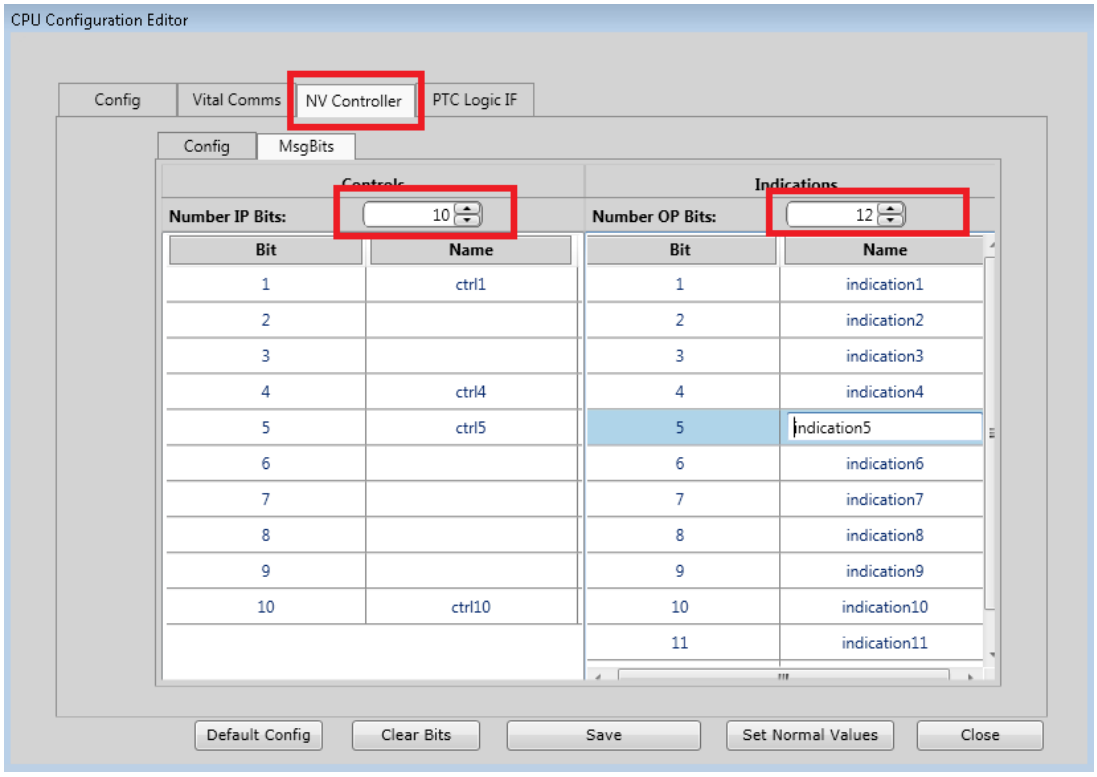


Figure 3-19 Configuring Message Bits

Not all the bits in the message need to be used, they can be left blank.

Buttons:

- **Default Config** does nothing
- **Clear Bits** will clear all the assigned bits on the Msg Bits screen
- **Save** – Saves changes.
- **Set Normal Value** – Bring up Normal Values window
- **Close** – Closes window.

NOTE

NOTE

The WCCT will check that the names entered on this screen are not duplicates of any other logic state names entered in the project, e.g. I/O module inputs and outputs and Boolean variables entered in Logic Data Editor. The user should write a Boolean equation to set the state of the output bits named here.

3.4.4 PTC Logic If Editor

The PTC Logic IF Editor tab allows the user to define the interface to the PTC system. There are three tabs: PTC Devices, Aspects, and MsgBits.

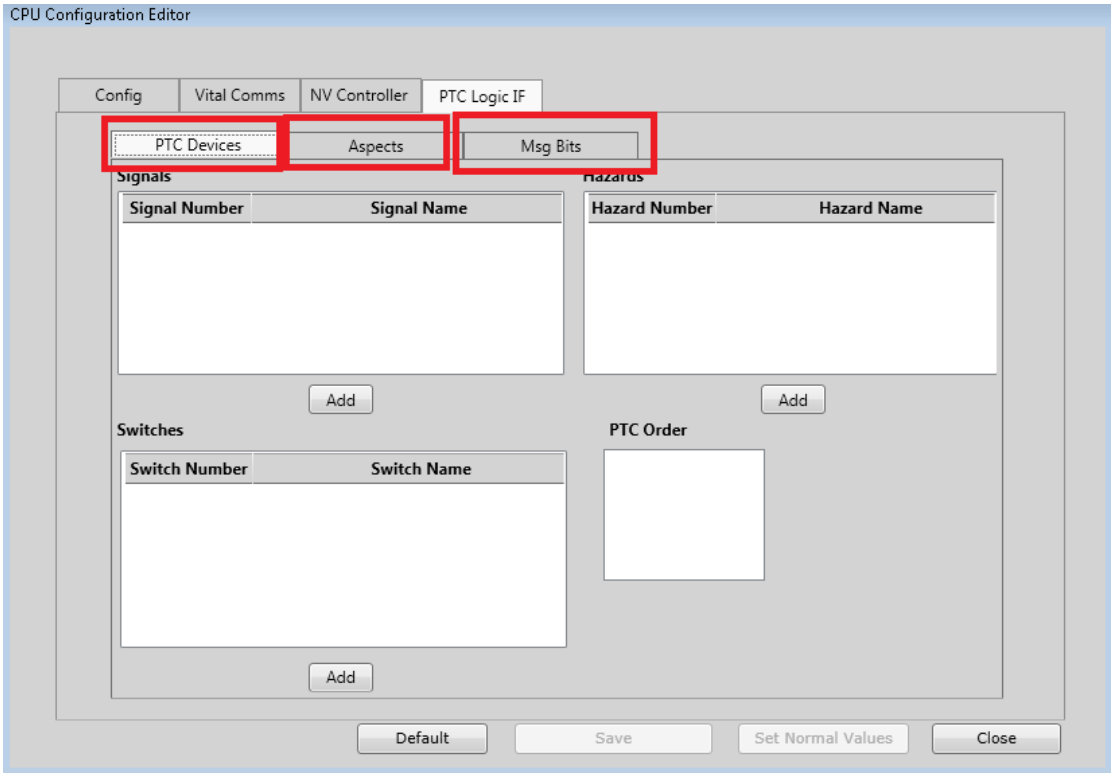


Figure 3-20 PTC Logic Tabs

3.4.4.1 PTC Devices

This screen is where you specify which PTC devices are used in the system. Use the **Add** button under Signals /Switches / Hazards boxes to add new devices.

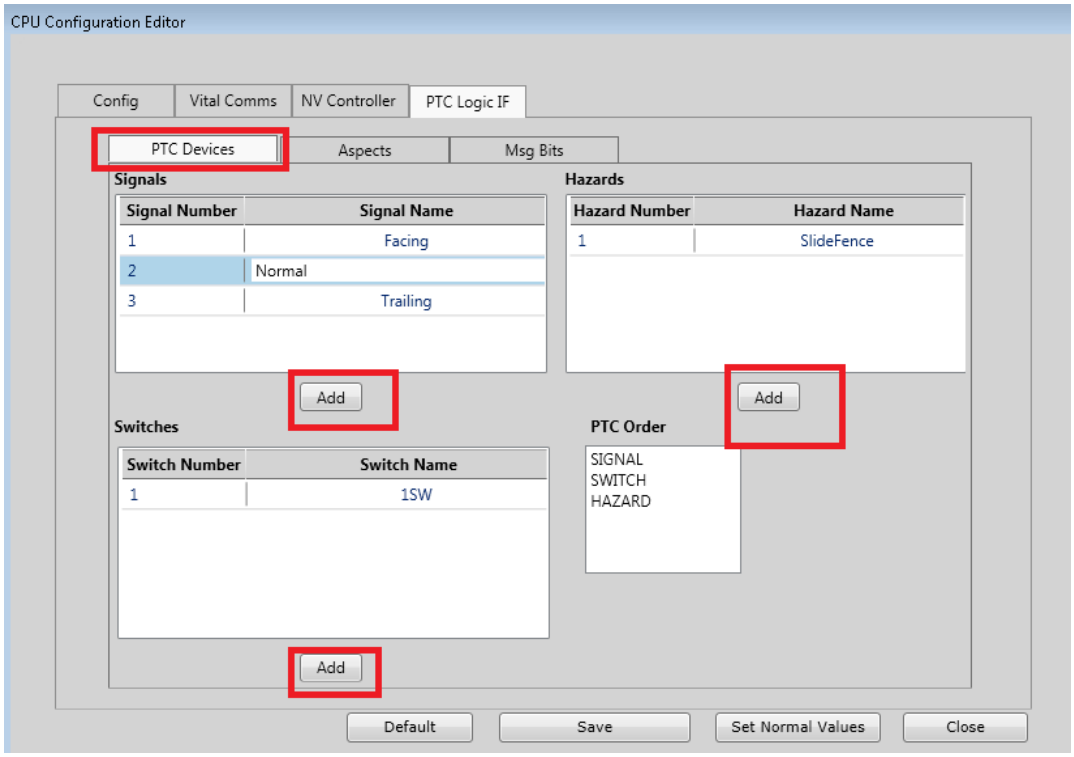


Figure 3-21 Adding Devices in PTC Logic IF Window

Select **Save** to save data.

The order in which the devices will be reported in the PTC message is determined by the PTC order first and then the order within the Signals, Switches, and Hazards boxes.

For example, in the above screen, the devices will be reported in the order:

- Facing Signal
- Normal Signal
- Trailing Signal
- 1SW
- SnowMelter

To change the order of whether signals, switches, or hazards are reported, right-click on the PTC order box and use the **Move Up/Move Down** to change the order.

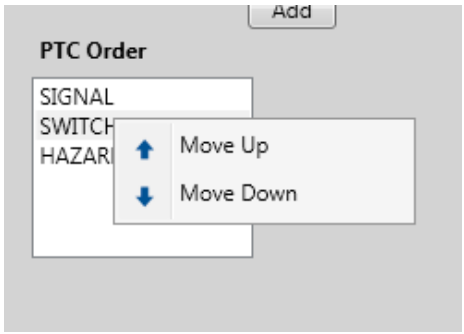


Figure 3-22 Changing Order of Signals, Switches, and Hazards

If the order of signals within the group of signals needs to be changed or a signal needs deleting, right-click in the signal number field and a drop-down menu will allow you to rearrange the signals, delete, or add one.

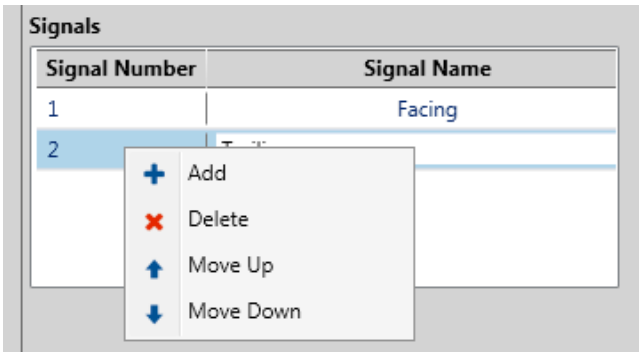


Figure 3-23 Deleting and/or Adding Signals, Switches, and Hazards

Similar drop-down menus exist for Switches and Hazard Detectors.

In the figure below, the changes in the order will result in the devices being reported in the following order:

- SlideFence
- 1SW
- Normal Signal
- Facing Signal
- Trailing Signal

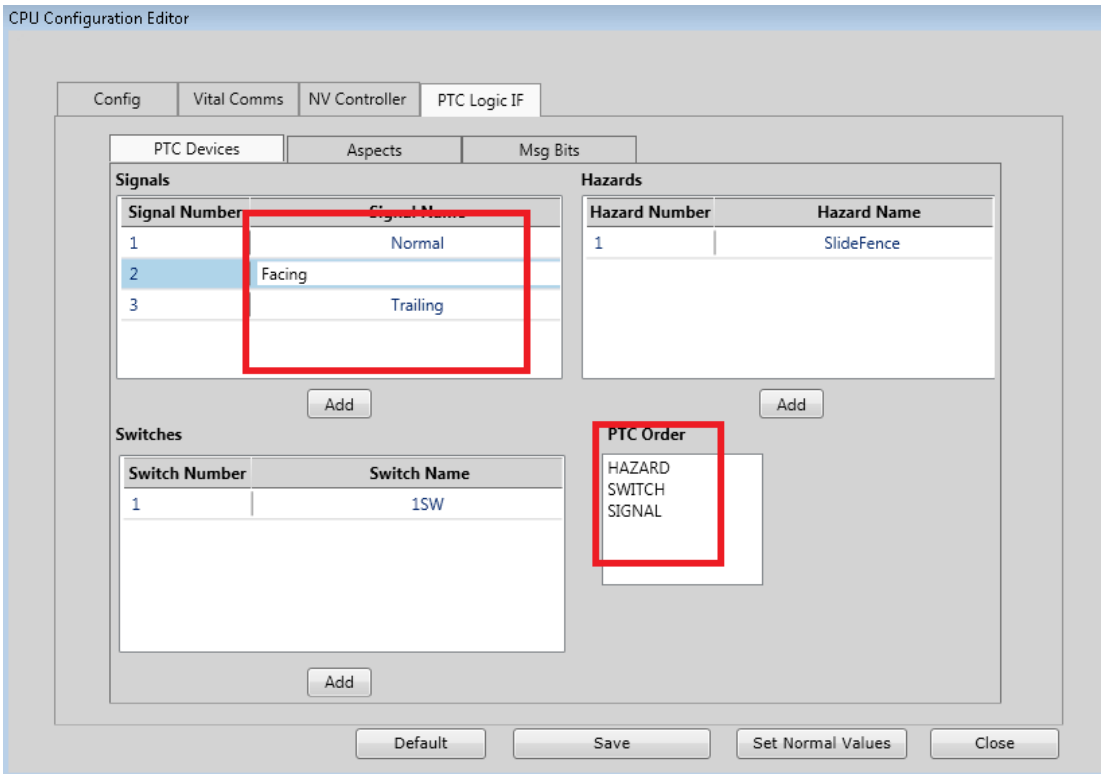


Figure 3-24 PTC Device Order Changed

3.4.4.2 Aspects

The aspects tab is used to specify which PTC aspects each signal will report and to assign a PTC Code to the aspect.

Two aspects are provided by default:

- The Illegal Aspect, with a default value of 0. This aspect will be reported by the CPU if aspect multiple bits are set at once by the logic.
- The All Dark Aspect, with a default value of 30. This aspect will be reported by the CPU if no aspect bits are set by the logic.

Use the **Add Aspect** button to add in new aspects, and change the aspect name and PTC Code as needed.

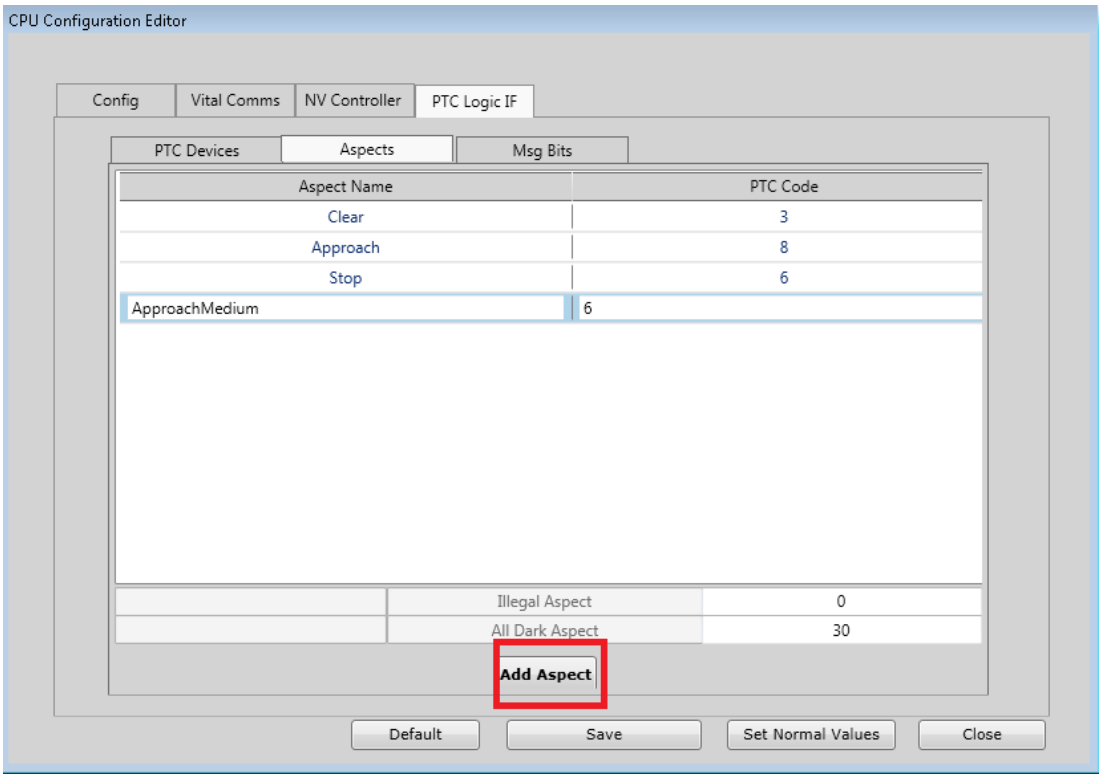


Figure 3-25 Aspects Tab

Aspects can be added with different Aspect Names but the same PTC Code. This setup may be used when creating an application that is used on territories that may have different names for the same signal aspect.

3.4.4.3 MsgBits

For Signals, the Msg Bits screen is used to assign logic states that can be set by the logic to select the desired aspect code for that signal.

The left window shows the list of PTC devices. Select each Signal in turn and assign a Logic state name to each aspect that the signal can display, as shown below.

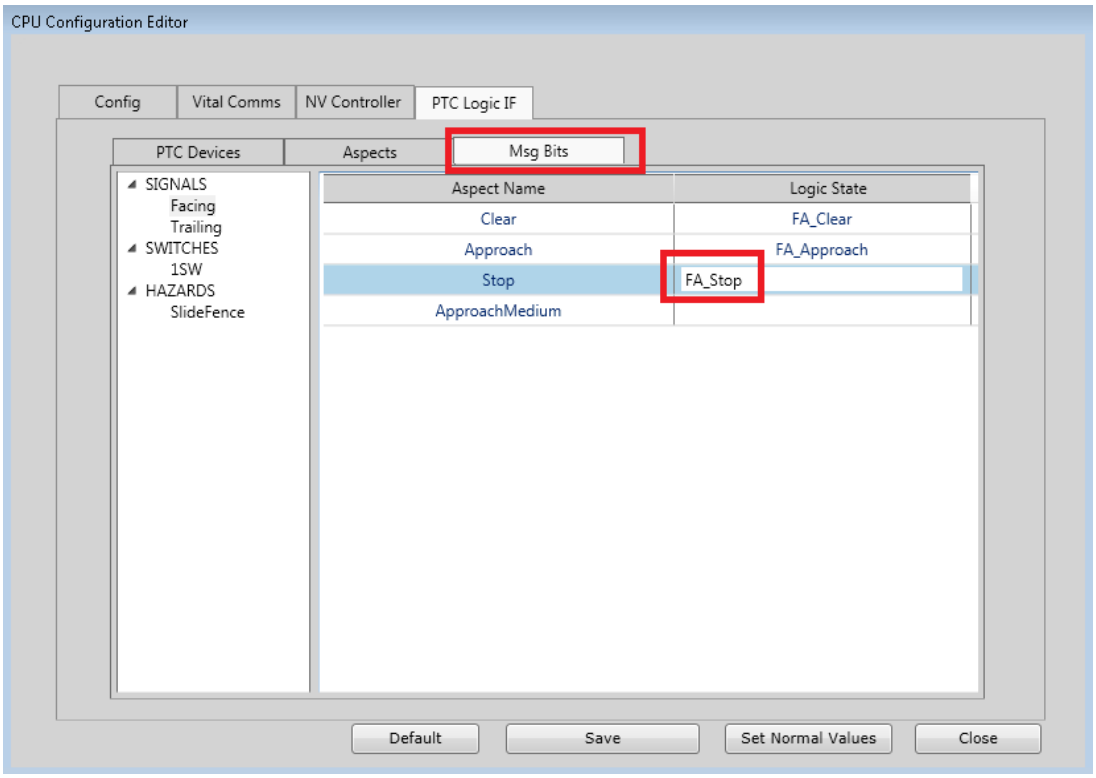


Figure 3-26 Assigning Msg Bit Names

If the signal doesn't display a particular aspect, leave the logic state field blank.

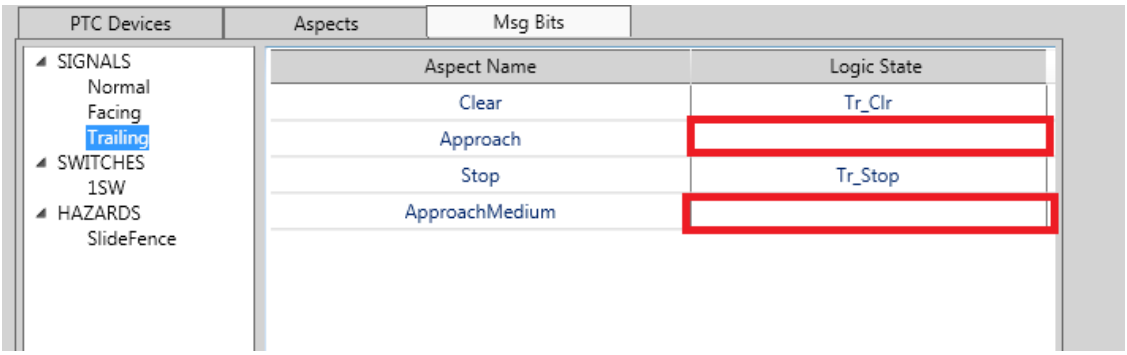


Figure 3-27 No Aspect to Display

In the logic section, write an equation for each of the logic state names in the list above to set its state to true when the desired PTC aspect is to be displayed. The signal may darken due to an approach lighting situation, but logically it can still have an aspect set.

To set the switch states, select the switch from the list shown, and enter logic state names for the reverse and normal positions. In the logic section write an equation to set the reverse and normal logic states.

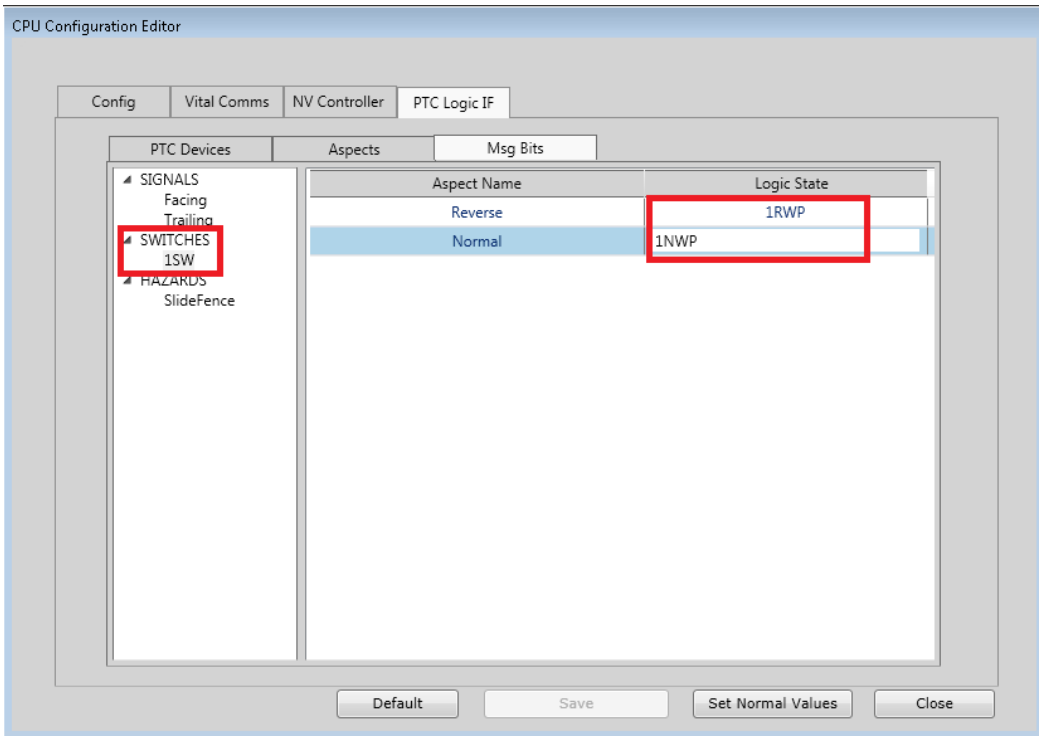


Figure 3-28 Setting Switch States

To set the hazard states, select the hazard from the list shown, and enter logic state names for the fault state. In the logic section, write an equation to set this state.

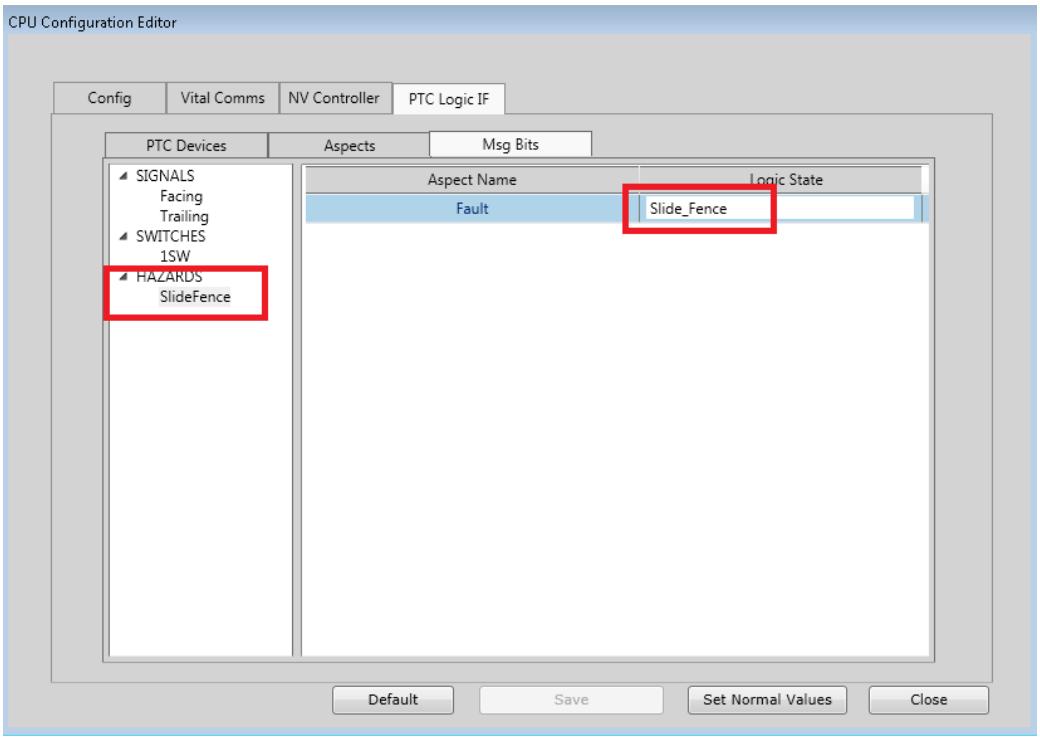


Figure 3-29 Setting Hazard States

3.5 VLP2 MODULE EDITOR

The VLP2 module editor has three tabs:

- Config
- Vital VComms
- NV Controller

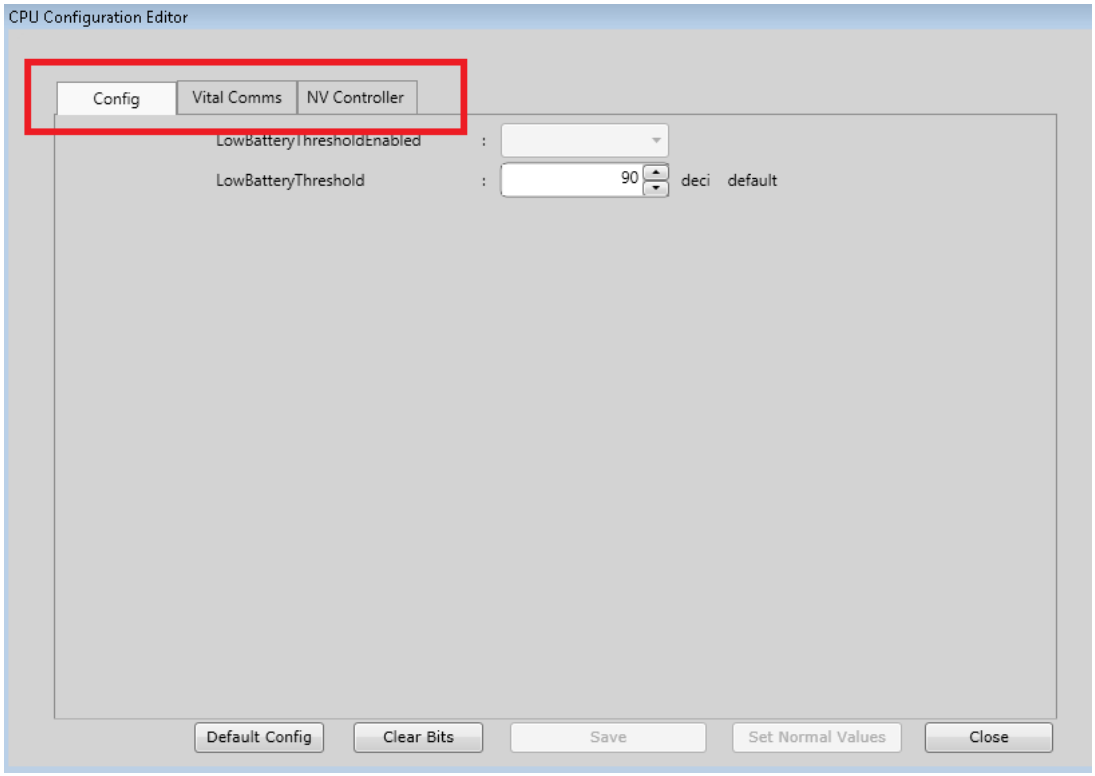


Figure 3-30 VLP2 Editor

3.5.1 Config

The Config tab shows two parameters relating to low battery detection.

If **LowBatteryThresholdEnabled** is set to **On** in the CPU II+, it will log entries in its status log when the battery voltage drops below the configured LowBatteryThreshold. Default value is 90dV (tenths of volts, i.e. 9V).

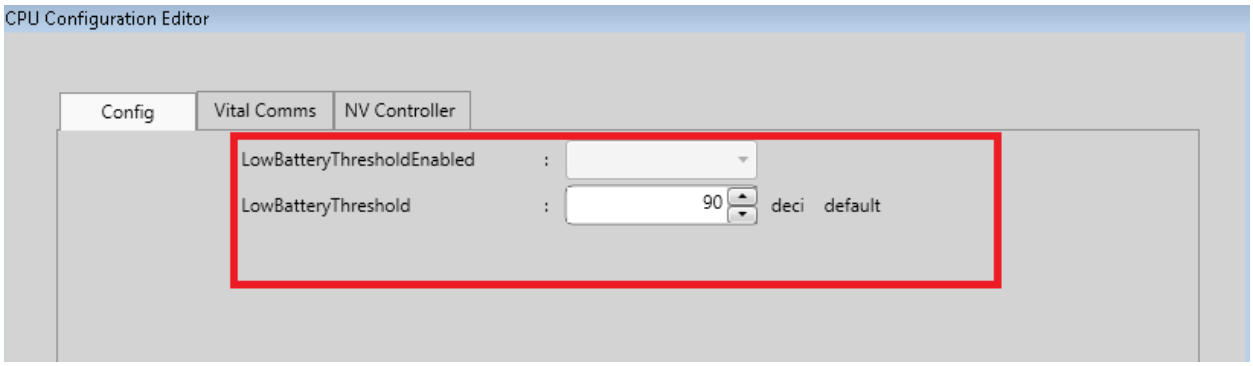


Figure 3-31 CPU Configuration Editor

3.5.2 Vital Comms Editor

The Vital Comms editor is the same as for the VLP3 described in Section 3.4.2. The only difference is that the maximum number of IP and OP bits supported by the CPUII + is 80.

3.5.3 NV Controller Editor

The NV Controller editor is the same as for the VLP3 described in Section 3.4.3. The only difference is that the maximum number of Control and Indication bits supported by the CPU II+ is 80.

3.6 CODED TRACK MODULE EDITOR

The Coded Track Module Editor has three tabs:

- Config
- CT I/O
- Other OPs

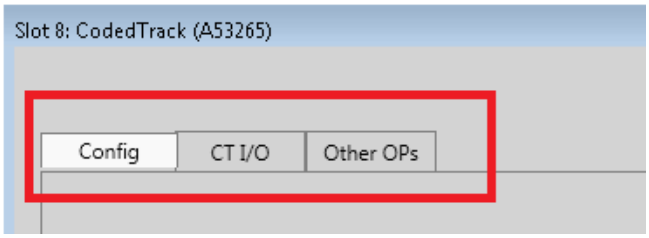


Figure 3-32 Coded Track Editor

3.6.1 Config

This tab allows the user to set values for track module configuration parameters. Generally, the UCN protected ones are set in the MCF. The track voltage is usually set in the field.

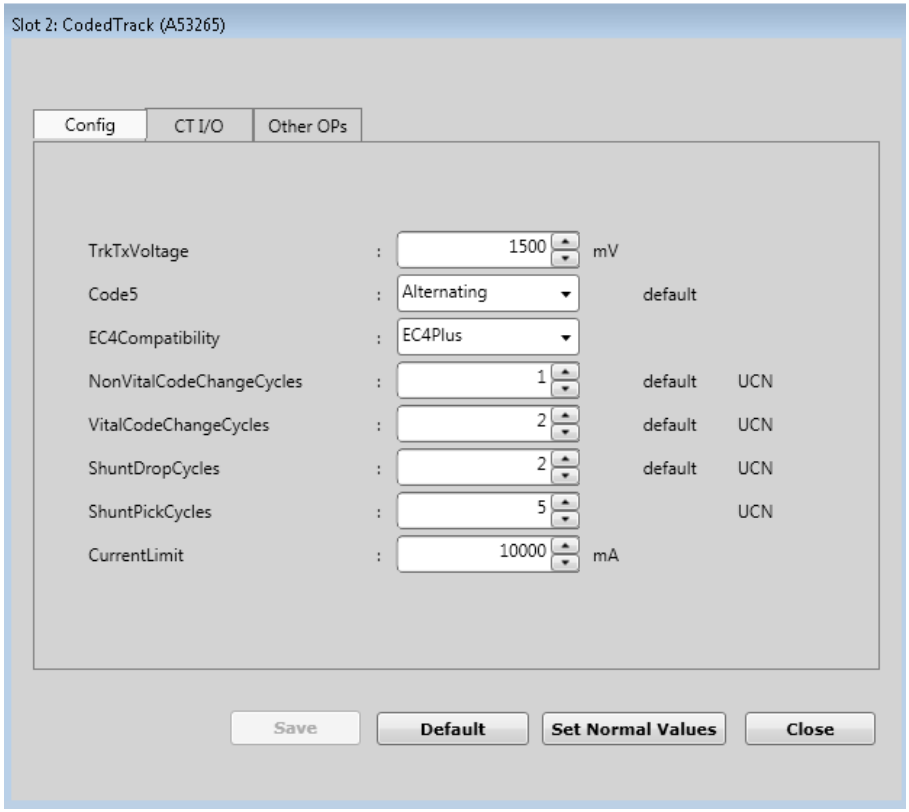


Figure 3-33 Coded Track Configuration Parameters

The following table describes the Coded Track Configuration parameters.

Table 3-2 CodedTrack Configuration Parameters and Descriptions

PARAMETER	DEFAULT	RANGE	COMMENTS	UCN
TrkTxVoltage	0 mV	0-4,000 mV in 20 ms steps	This sets the voltage for the track card transmitter. Its value is usually set in the field.	No
Code5	Alternating	Alternating Standard Long	This is used to specify which type of the code pattern is used for code 5	No
EC4 Compatibility	EC4	EC4Plus EC4	This is used to specify whether the code patterns are compatible with Electrocode 4 or Electrocode 4+.	No
NonVitalCode ChangeCycles	1	1 to 3	This is used to specify how many code cycles a non-vital code has to be received before the change is reported. Default usually set in MCF	Yes
VitalCode ChangeCycles	2	1 to 3	This is used to specify how many code cycles a vital code has to be received before the change is reported. Default usually set in MCF	Yes
ShuntDrop Cycles	2	1 to 8	This is used to specify how many code cycles the code has to be lost before reporting that the track is shunted. Required default usually set in MCF	Yes
ShuntPick Cycles	1	1 to 8	This is used to specify how many code cycles the code has to be seen before reporting that the track is not shunted. Default usually set in MCF	Yes
CurrentLimit	1,000 mA	1,000 to 10,000 mA in 50 mA steps	This is used to specify the maximum current that the track module can received before it turns off its transmitter. It is used to prevent the module sending codes when a train is shunting the rail. Usually set in the field	No

3.6.2 CT I/O

This tab is used to:

- Enter the logic states that are used to set specific track codes
- Enter logic state names that are set when track codes are received

NOTE

NOTE

On all screens where input and output bits names are entered:

a) the name can to be up to 30 characters long, contain only upper and lower case letters, numbers, and underscores (_)
b) the name must contain a letter or underscore, it cannot be all numbers
c) the name cannot be a duplicate of a logic state name entered anywhere else, i.e. I/O module editors, CPU editors, Logic Data Editor,
d) though upper and lower case names can be entered, the duplicate checking is not case sensitive, so the name Input1 is treated as a duplicate of INPUT1 and input1.

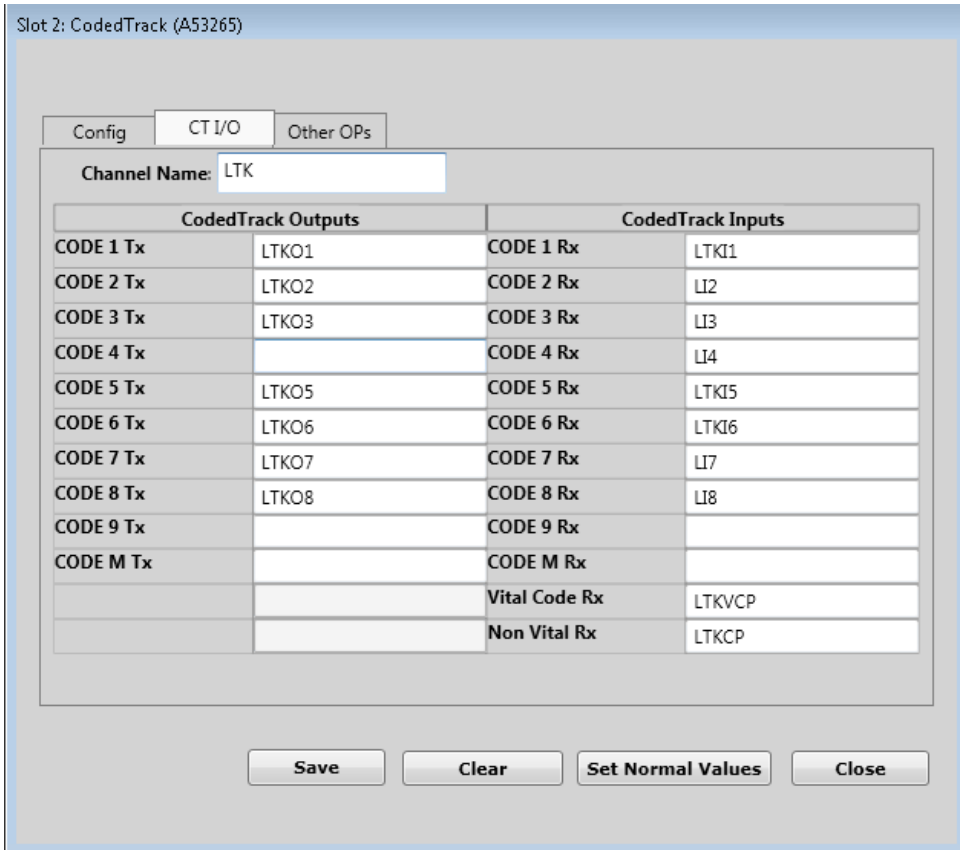


Figure 3-34 Coded Track I/O Parameters

NOTE

NOTE

a) When transmitting a vital code (2,3,4,7,8,9) or non-vital code (5,6,M), the code 1 bit must also be set.
b) When the code 6 bit is set, the track module will transmit 1 pulse of track 6 then go back to transmitting the previously commanded code. e.g. if code bits Code 1 and 7 then 6 bit is set as well, track card send 1 pulse of code 6, then goes back to transmitting a code 7.To send another code 6 pulse, the code 6 bit has to be cleared in the logic for at least 1 Electrocode code cycle then set again.
c) The Non-Vital Rx bit is set when any code has been received in the last code cycle.
d) The Vital Code Rx bit is set when any vital code has been received in the last code cycle.
e) When a shunt on the track is removed, the Non-Vital Rx and Vital Code Rx bits will be set before the non-vital code bits (1,5,6,M) or vital code bits (2,3,4,7,8,9) are set.
f) Once a code has been set for a number of cycles greater or equal to the configured value of Shunt pick cycles, the code bits will be set (provided the code has been seen for at least the values of Code change cycles)

Example:

Consider a shunted track circuit which is receiving a code 1,7 from transmit end (but not received at rx end while shunted), the shunt is then lifted.

ShuntPick = 4

VitalCodeChangeCycles = 2

NonVitalCodeChangeCycles = 1

Table 3-3 Shunt Code Reference 1

	Shunt	Remove Shunt	No Shunt	No Shunt	No Shunt	No Shunt
Code Cycle No.	1	2	3	4	5	6
Code Pulse Received	None	1,7	1,7	1,7	1,7	1,7
Rx Code bits set	None	NVC VC	NVC VC	NVC VC	NVC VC 1,7	NVC VC 1,7

The code 1 and 7 bits are not set until they have been received for 4 cycles.

Example:

Consider a non-shunted track circuit which is receiving a code 1,7 and is then shunted.

ShuntDrop = 2

VitalCodeChangeCycles = 2

NonVitalCodeChangeCycles = 1

Table 3-4 Shunt Code Reference 2

	No Shunt	Shunt	Shunt	Shunt	Shunt	Shunt
Code Cycle number	1	2	3	4	5	6
Code Pulse received	1,7	-	-	-	-	-
Rx Code bits set	NVC VC 1,7	1,7	-	-	-	-

The code pulse is lost, so the NVC and VC bits clear, but the vital code persists for the number of ShuntDrop delay cycles.

The timing of the shunt may result in the second pulse of the vital code being cut off, and leaving the first, so the logic may also see:

Table 3-5 Shunt Code Reference 3

	No Shunt	Shunt	Shunt	Shunt	Shunt	Shunt
Code Cycle number	1	2	3	4	5	6
Code Pulse received	1,7	1	-	-	-	-
Rx Code bits set	NVC VC 1, 7	NVC 1	-	-	-	-

When code 5 and M are used in alternating mode, the track module takes care of alternating the code, the logic just has to set the Code 5 Tx or Code M Tx bit to true.

3.6.3 Other OPs

This tab is used to set controls for the Vital Output on the track module and control its LED. The LED is usually used to indicate when a stick is set.

Enter variable names for the VRO On State and the Stick LED control.

The Vital Output can also be used to generate cab output signals. To use it in this mode, select the Cab check box, then enter logic state names that can be used to set the desired cab rates. Only one cab rate bit can be set at once by the Boolean logic, if multiple bits are set, the track card will reboot.

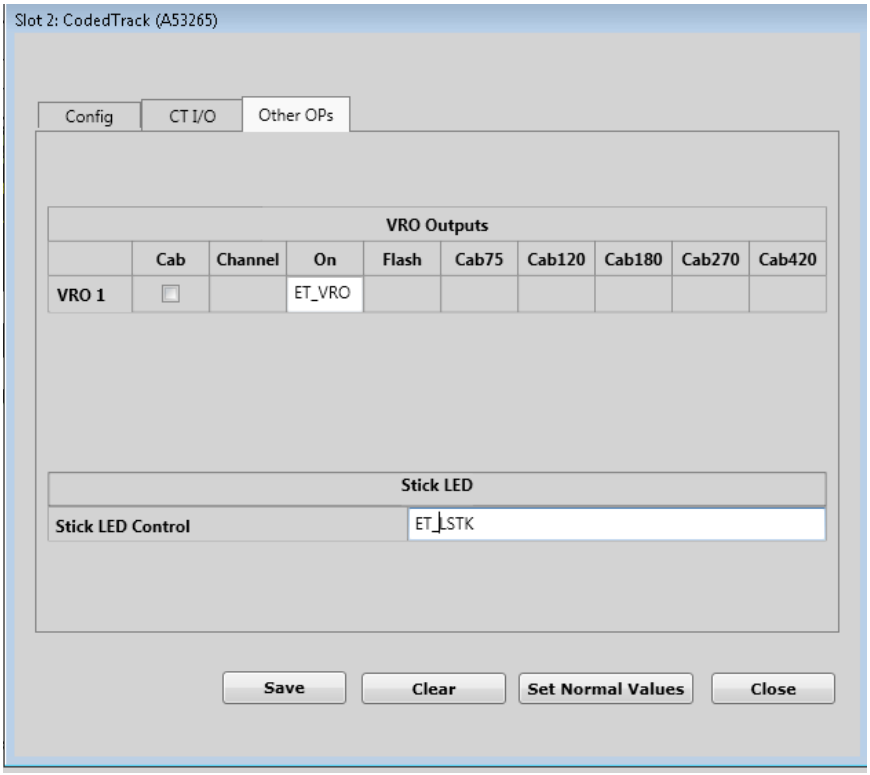


Figure 3-35 Configuring VRO Outputs

3.7 CODED LINE MODULE EDITOR

The Coded Line Module Editor has three tabs:

- Config
- CT I/O
- Other OPs

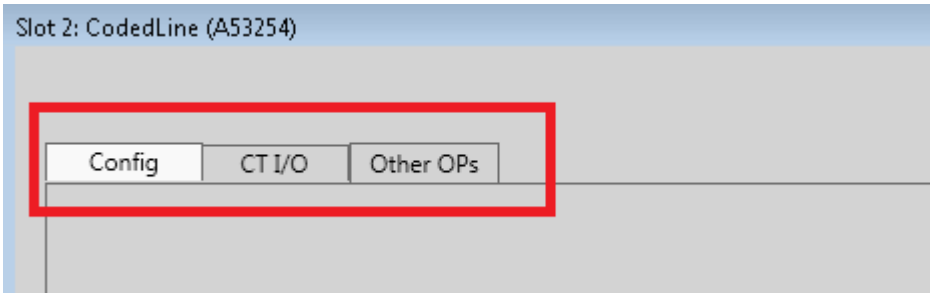


Figure 3-36 Coded Line Editor

3.7.1 Config

This tab allows the user to set values for line module configuration parameters.
Generally, the UCN protected ones are set in the MCF. The Line Tx Voltage is usually set in the field.

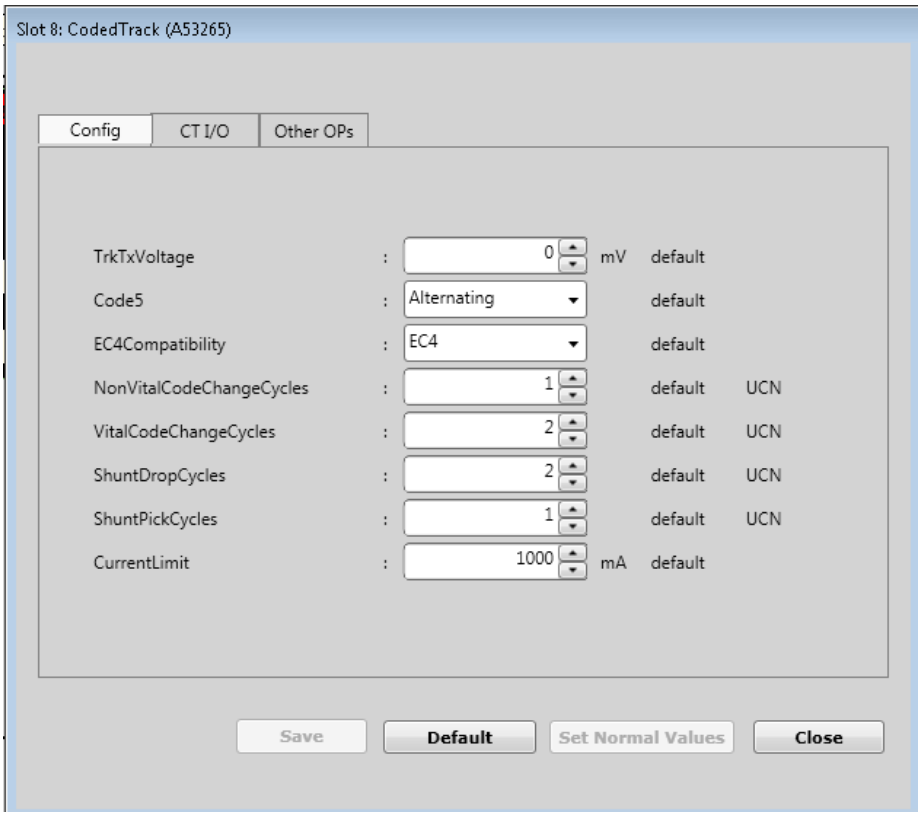


Figure 3-37 Coded Line Configuration Parameters

The following table describes the Coded Line Configuration parameters.

Table 3-6 CodeLine Configuration Parameters

PARAMETER	DEFAULT	RANGE	COMMENTS	UCN
Line TxVoltage	2,000 mV	2,000-1,5000 mV in 20 mV steps	This sets the voltage for the track card transmitter. Its value is usually set in the field.	No
Code5	Alternating	Alternating Standard Long	This is used to specify which type of the code pattern is used for code 5	No
EC4 Compatibility	EC4	EC4Plus EC4	This is used to specify whether the code patterns are compatible with Electrocode 4 or Electrocode 4+.	No
NonVitalCode ChangeCycles	1	1 to 3	This is used to specify how many code cycles a non-vital code has to be received before the change is reported. Default usually set in MCF	Yes
VitalCode ChangeCycles	2	1 to 3	This is used to specify how many code cycles a vital code has to receive before the change is reported. Default usually set in MCF	Yes
ShuntDrop Cycles	2	1 to 8	This is used to specify how many code cycles the code has to be lost before reporting that the track is shunted. Required default usually set in MCF	Yes
ShuntPick Cycles	1	1 to 8	This is used to specify how many code cycles the code has to be seen before reporting that the track is not shunted. Default usually set in MCF	Yes
Receive Threshold	2,000 mV	2,000 to 15,000 mV in 50 mV steps	Used to set the threshold voltage above which the module detects the code. Usually set in the field	No

3.7.2 CT I/O

This tab is used to:

- Enter the logic states that are used to set specific track codes
- Enter logic state names that are set when track codes are received

First enter a channel name for this channel.

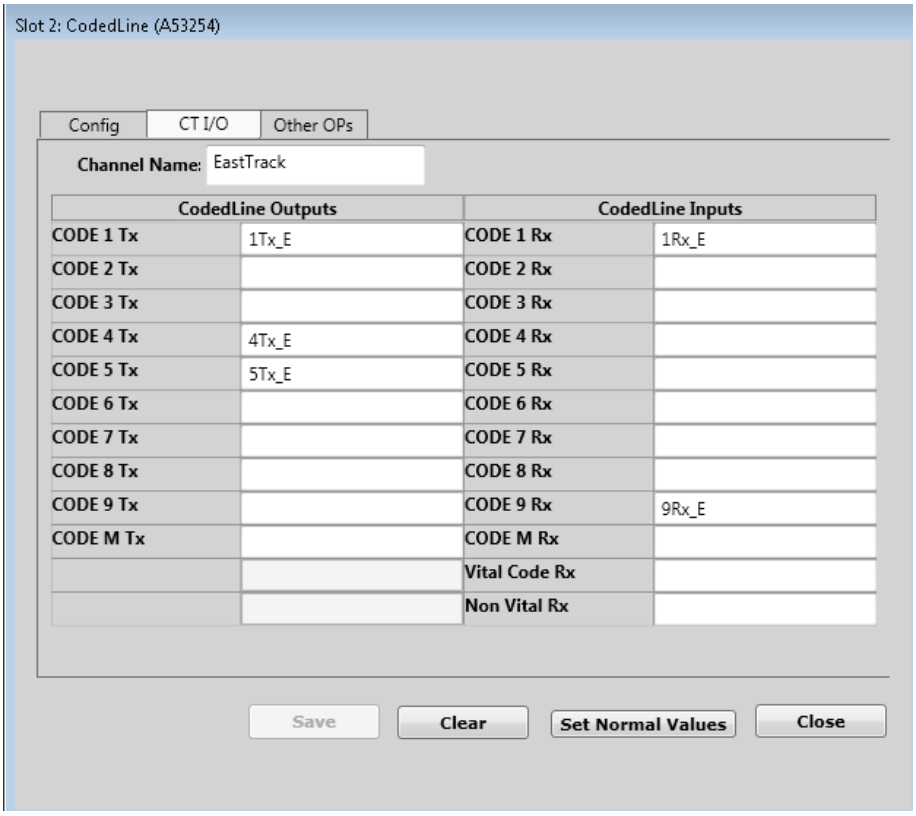


Figure 3-38 Entering Channel Name

The same rules about track code apply to the Line Module as the Coded Track Module.

3.7.3 Other OPs

This tab is used to set controls on the LED on the module. The LED is usually used to indicate when a stick is set. Enter a variable name for Stick LED control.

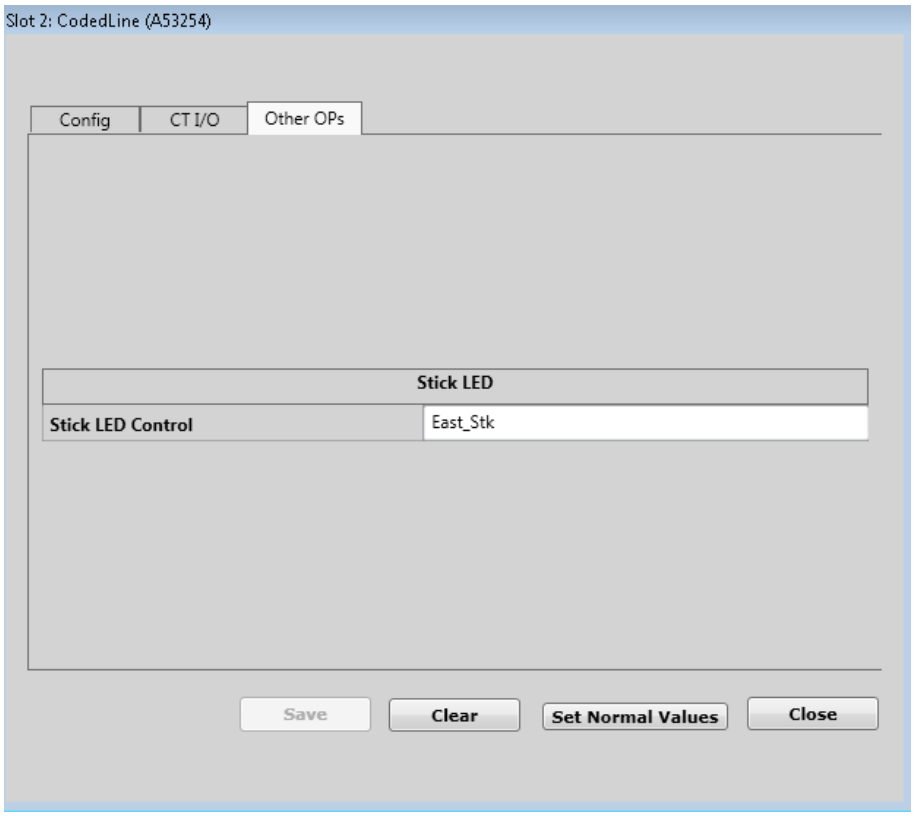


Figure 3-39 Other OPs

3.8 WAYTRAX EDITOR

The WayTraX Module Editor has three tabs:

- Config
- Codes
- VRO

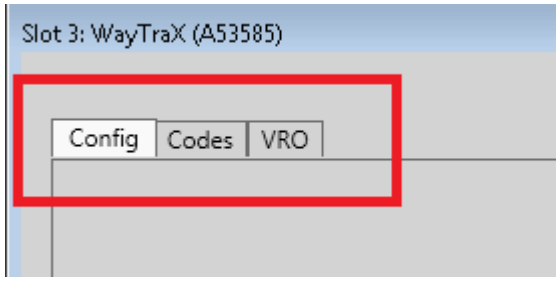


Figure 3-40 WayTraX Editor

3.8.1 Config

This tab allows the user to set values for WayTraX module configuration parameters. Generally, the UCN protected ones are set in the MCF. The GTTrkLength is usually set in the field.

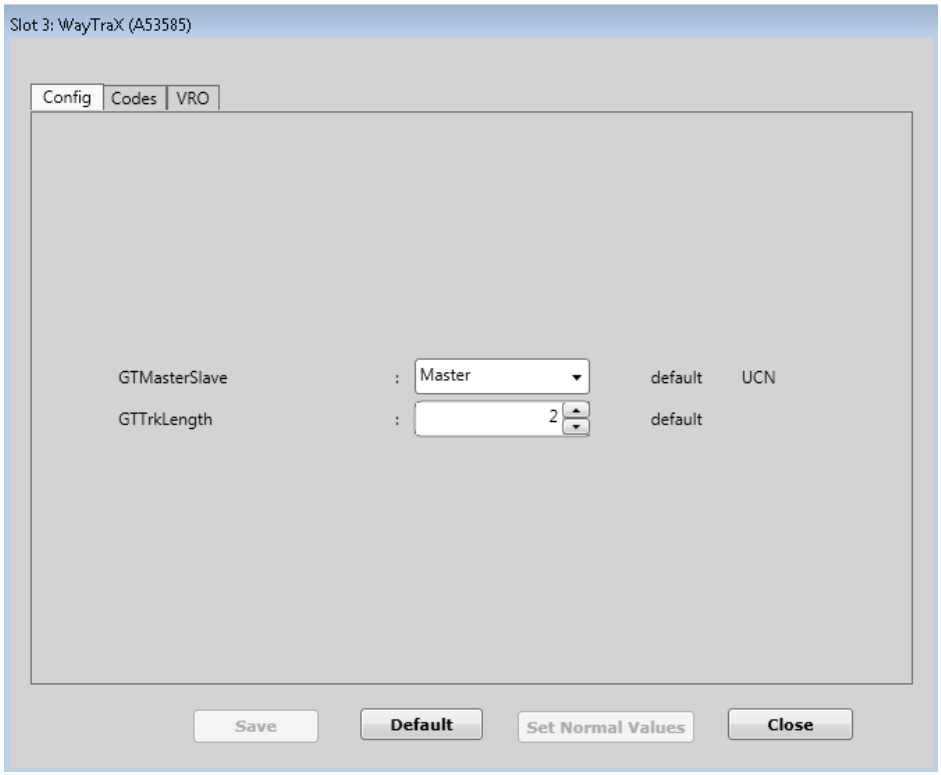


Figure 3-41 WayTraX Configuration Parameters

The following table describes the WayTraX Configuration parameters.

Table 3-7 WayTraX Configuration Parameters

PARAMETER	DEFAULT	RANGE	COMMENT	UCN
GTMasterSlave	Master	Master, Slave	This sets the operational mode for the WayTraX module. A module configured as Master must be connected to one configured as Slave	Yes
GTTrackLength	2	0-36 in 1000ft increments	This is the nominal track length. The actual setting will depend on the ballast conditions, so set this to the desired value in the field	No

NOTE

NOTE

A WayTraX module configured as a master can only communicate with a slave.

The Track length setting should be left to be set in the field, since the exact value to set depends on the ballast conditions as well as track length. See WayConneX Manual (SIG-00-16-06) for details.

3.8.2 Codes

The WayTraX Codes window is shown below.

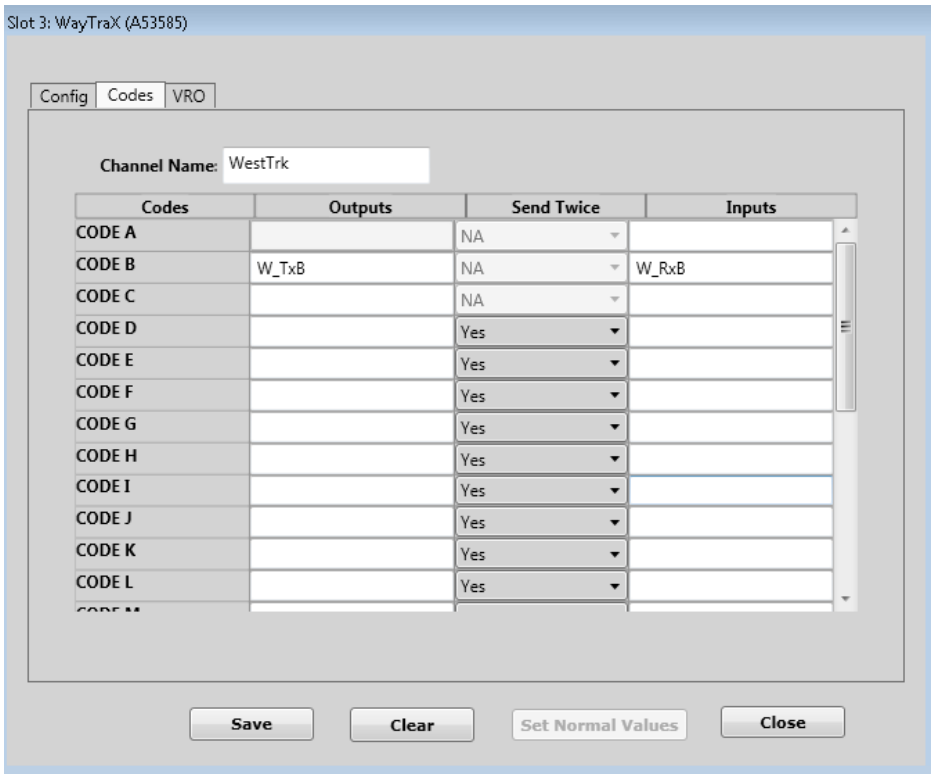


Figure 3-42 WayTraX Code Window

The Output column of the Codes screen allows the user to select the names of the variables which control the track codes sent by the WayTraX module. The Input column of the Codes screen allows the user to select the names of the variables which are set when the associated track codes are received by the WayTraX module.

The codes are identical and compatible with MicroTrax. Code A is equivalent to the link up code and so is automatically sent by the master after a shunt is removed; the slave will respond with a code A, the master will then send the code commanded by the application.

Note: when writing an application using the WayTraX codes, the application should always command a code; otherwise, the master and slave will just go in and out of Shunt/Remove Shunt mode.

Codes B and C differ from Codes D through W as they are received in 1 code cycle.

- the Input variable associated with code B or C will be set on receipt of Code B or C for one cycle by the WayTraX module
- the Input variable associated with code D (through W) will be set on receipt of two consecutive Code D (through W) by the WayTraX module
- the Input variable associated with code A will be set on receipt of Code A as part of the linkup by the WayTraX module

NOTE

NOTE

Unlike in MicroTrax, either code B or code C can be used as a tumble down code. WayTraX does not support sleep mode.

The WayTraX module allows the user to specify whether a code is guaranteed to be sent for two consecutive cycles. This only applies to code D-W.

If the following code sequence is received by the WayTraX: D D D E F F F, the module will report no received code on receipt of the first code F, as code D has not been seen for two cycles, and no new code has been received for two consecutive cycles. In order to prevent this from causing operational issues, the codes have the option of being guaranteed to be sent for two code cycles (default). If the user wants to use one of these codes as an additional non-vital code, they have the option to set **Send Twice** to **No**.

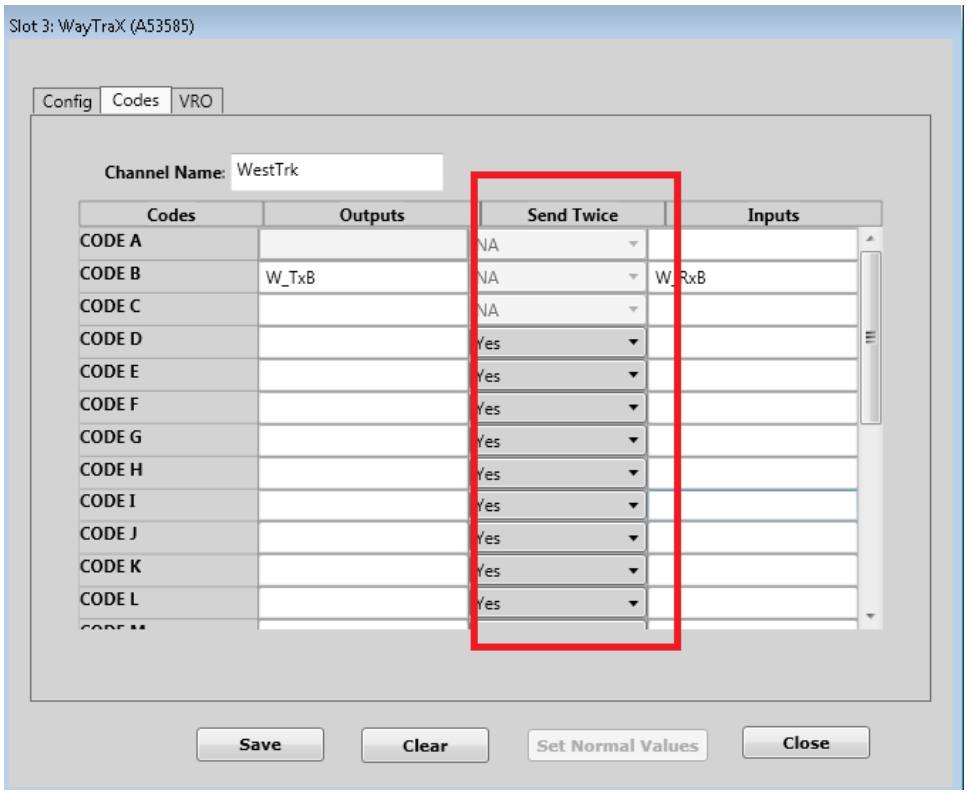


Figure 3-43 WayTraX Code Send Twice

3.8.3 VRO

This tab is used to set controls for the Vital Output on the WayTraX module. Enter variable names for the VRO On State. The Vital Output can also be used to generate cab output signals. To use it in this mode, select the **Cab** check box, then enter logic state names that can be used to set the desired cab rates. Only one cab rate bit can be set at once by Boolean logic, if multiple bits are set, the track card will reboot.

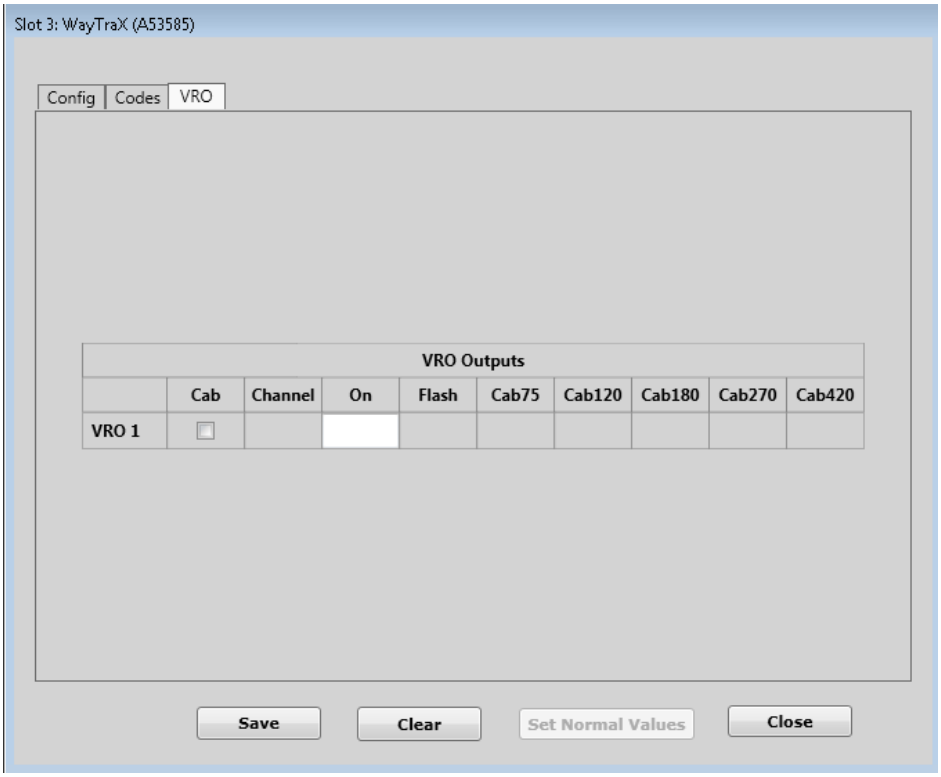


Figure 3-44 WayTraX VRO Parameters

3.9 COLORLIGHT MODULE EDITOR

The Colorlight Module Editor has four tabs:

- Config
- Lamps
- I/O
- DC Converter

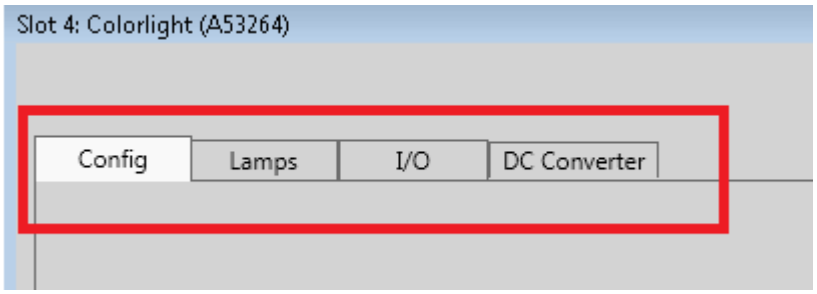


Figure 3-45 Colorlight Editor Tabs

3.9.1 Config

This tab allows the user to set values for Colorlight module configuration parameters. Generally, the UCN protected ones are set in the MCF.

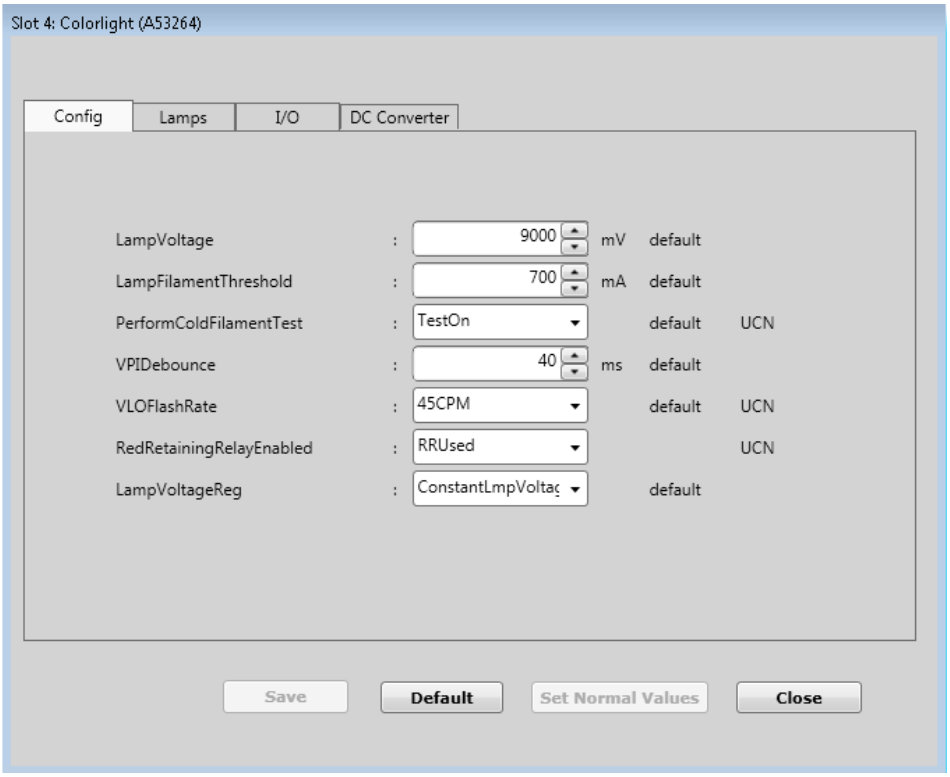


Figure 3-46 Colorlight Module Config Editor

If RedRetainingRelayEnabled is set to RRUsed, the user must write logic to control the converter channel. The WCCT will give a warning that the converter must be defined if the parameter is changed and no converter channel is defined.

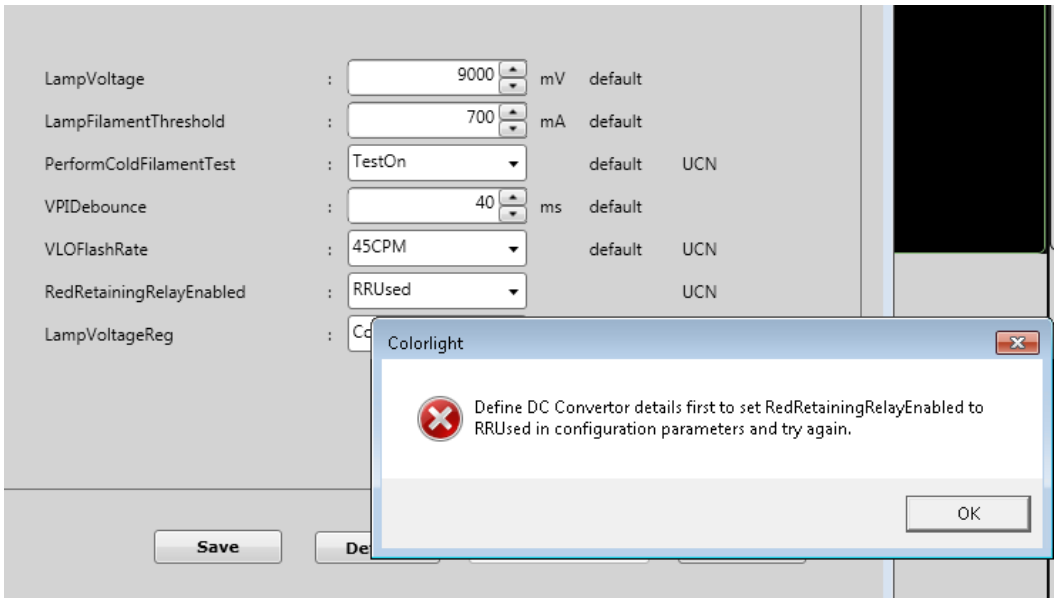


Figure 3-47 Colorlight Module RRUsed Warning Message

For usage of the RedRetainingRelayEnabled parameter, see the section 3.9.4.

The following table describes the Colorlight configuration parameters.

Table 3-8 Colorlight Configuration Parameters

PARAMETER	DEFAULT	RANGE	COMMENTS	UCN
LampVoltage	9,000 V	9,000 to 13,000 mV in 20 mV steps	Used to set the lamp voltage for all 6 lamps on the board Usually set in field	No
LampFilament Threshold	700 mA	150 – 2,500 mA in 10mA steps	Used to set the threshold for LOR detection current	No
PerformCold FilamentTest	TestOn	TestOn, TestOff	Used to indicate whether cold filament testing is done. Should be kept at TestOn. Usually set in MCF	Yes
VPIDebounce	40 ms	20 to 200 ms in 2 ms steps	Set VPI Debounce time. (milliseconds)	No
VLOFlashRate	45CPM	40CPM 45CPM 50CPM 55CPM 60CPM 65CPM	Used to select the flash rate for flashing lamps (CPM) with 50% duty cycle Usually set in MCF	Yes
RedRetaining RelayEnabled	RRNotUsed	RRNotUsed, RRUsed	See VSTOP section for how to set this Usually set in MCF	Yes
LampVoltage Reg	Constant Lmp Voltage	ConstantLmp Voltage VariableLmp Voltage	The module will try and maintain a constant lamp voltage. If AC power is off this can cause the battery voltage to drain more rapidly. By setting this to VariableLmpVoltage the module will gradually reduce the lamp voltage as the battery voltage drops.	Yes

3.9.2 Lamps

This tab is used to:

- Enter the logic states that are used to control the lamps on and flash states
- Enter the logic state names that are set when a lamp reports a light out condition (LOR) or foreign energy (FE)

First enter a channel name for each used lamp. To ensure that the lamps have the correct color displayed when shown on the CPU user interfaces (Web UI of the CPU III / DT for CPU II+) the channel name must contain the color of the lamp in the following form: Green, Grn, Yellow, Yel, Red, Lunar, Lun (lower or upper case). Enter names for output controls to turn lamps on or flash them. Enter names for states that will be set if lamps detect Light Out (LOR) or foreign energy (FE).

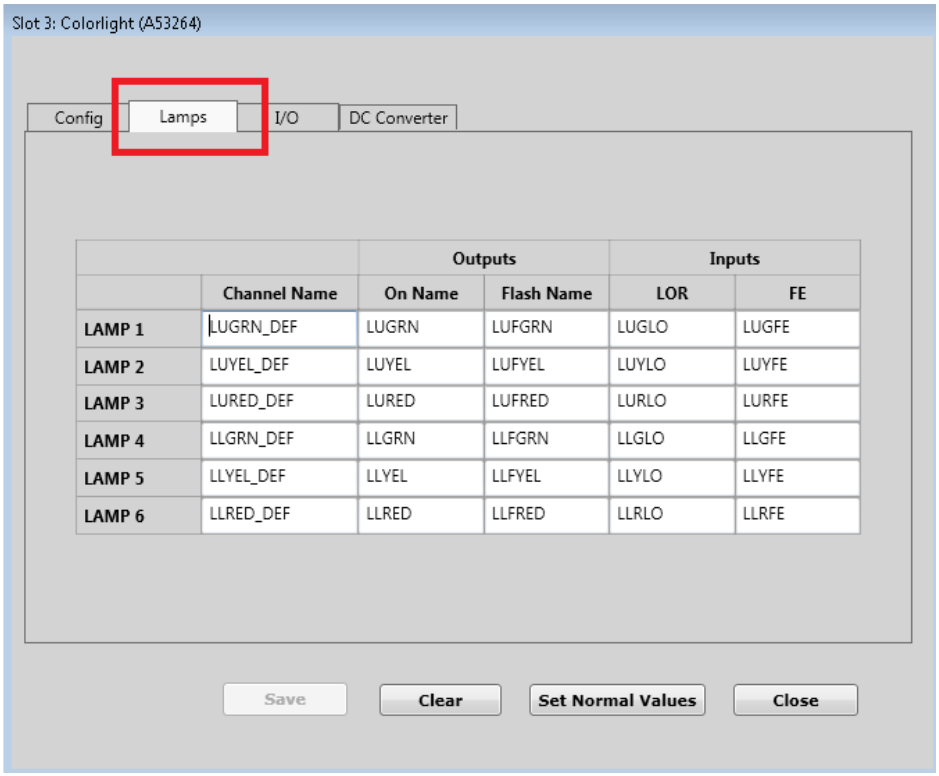


Figure 3-48 Entering Lamp Channel Names

NOTE

NOTE

- To flash a lamp, set the **Flash** state to true. The **On** state may be set to true or false.
- When a lamp detects it has a burned out filament, it will report the LOR state as true, and will turn off the lamp , even if the application logic still has the On state (or Flash) set true.
- When a lamp detects it has foreign energy applied to it, it will turn off the DC-DC convertor and all lamps will report LOR. If foreign energy is still detected on the lamp after all the lamps of this card have been turned off, it will report FE for that lamp.

Note: if the Colorlight module is not in session with the CPU (e.g. it has been removed, or has failed), the logic will set the reported states of the LOR and FE bits to true.

3.9.3 I/O

The Colorlight module has a Vital Relay Output and 2 Vital Parallel Inputs (VPI). Use the I/O screen to:

- set controls for turning on the VRO, or setting it to generate a Cab rate (see Coded Track section 3.6)
- set logic states that will indicate the state of the VPIs

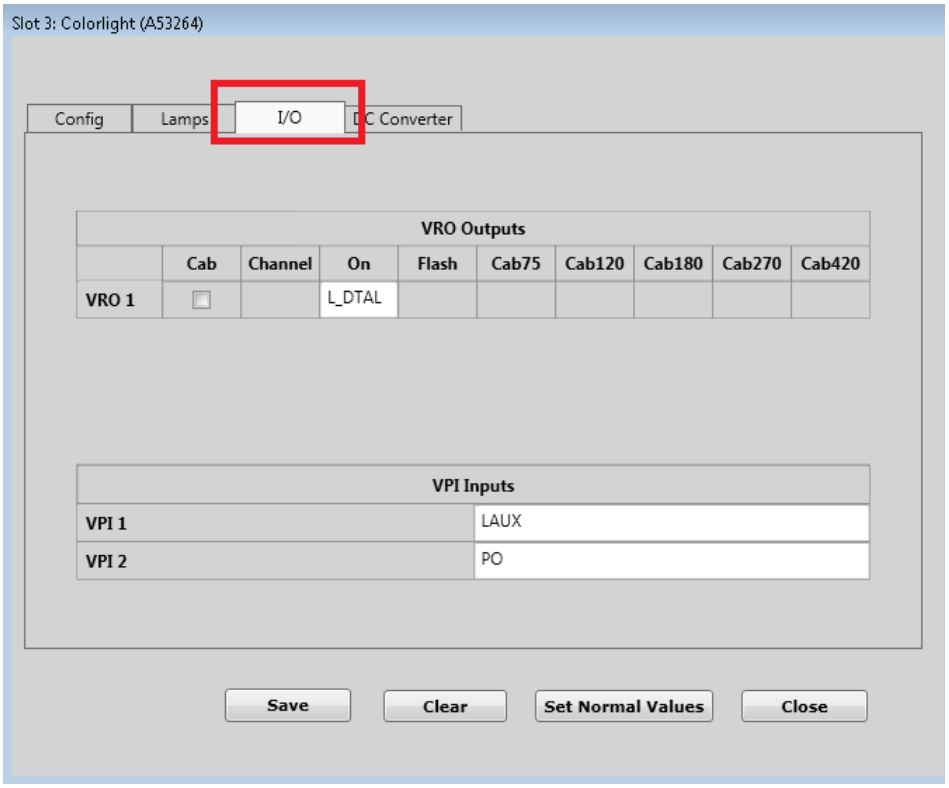


Figure 3-49 Colorlight VRO and VPI

3.9.4 DC Converter

Use this screen to set state to control the DC-DC converter to the Colorlight module and read its status. First enter a channel name, then enter names for commands and status.

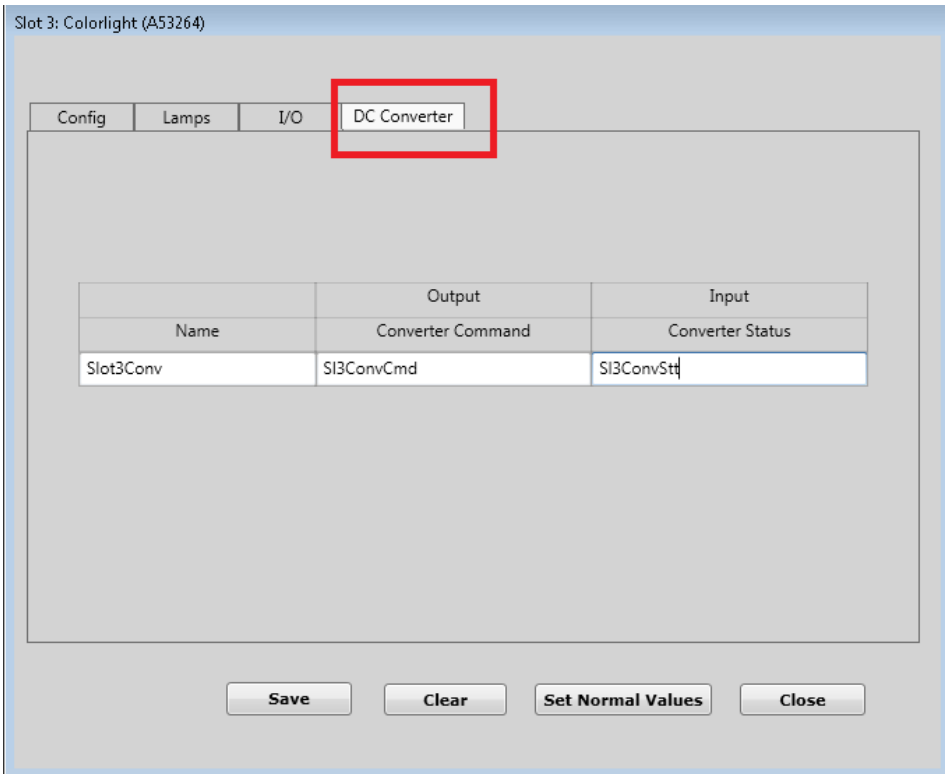


Figure 3-50 DC Converter States

Monitoring the state of the convertor and controlling it is only necessary when the configuration parameter RedRetainingRelayEnabled is set to RRUsed.

When RedRetainingRelayEnabled is set to RRNotUsed the DC-DC convertor on the Colorlight module is controlled purely by the module executive software. For example, if foreign energy is detected on a lamp, the DC-DC convertor is shut off and all lamps turned off. When the foreign energy is removed, the module executive software will automatically turn the convertor back on.

When RedRetainingRelayEnabled is set to RRUsed, the DC-DC convertor has to be turned on using the Convertor Command bit set from the logic (see previous page). The purpose for this is in case there is a signal whose heads are split across multiple Colorlight modules; because, if foreign energy is detected on one module and the DC-DC convertor turned off (as detected by the convertor Status bit) the application logic can be used to de-energize the DC-DC convertor on the other module. If a RedRetainingRelay is used to break the contacts on signal heads that are split across different modules, the DC-DC convertor on the two modules needs to be switched on at the same time, from the application logic.

3.10 SEARCHLIGHT MODULE EDITOR

The Seachlight Module Editor has four tabs:

- Config
- Lamps
- Mech
- I/O

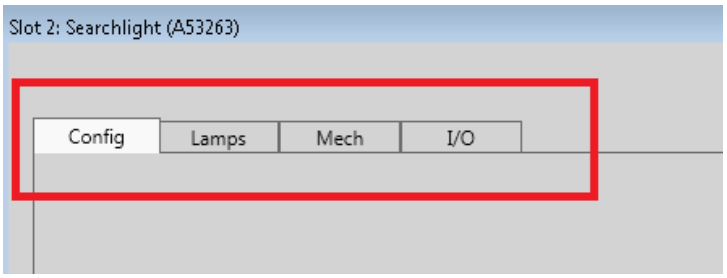


Figure 3-51 Searchlight Editor Tabs

3.10.1 Config

The Config tab allows the user to set values for Searchlight module configuration parameters. Generally the UCN protected ones are set in the MCF.

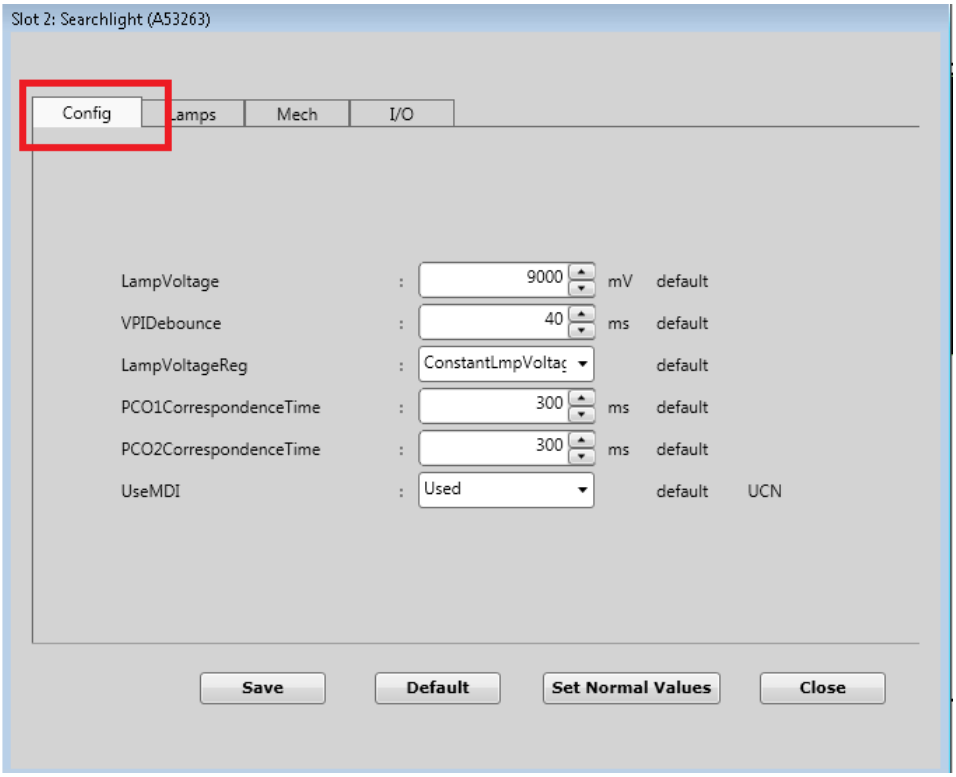


Figure 3-52 Searchlight Module Editor

The following table describes the Searchlight Configuration parameters.

Table 3-9 Searchlight Configuration Parameters

PARAMETER	DEFAULT	RANGE	COMMENTS	UCN
LampVoltage	9,000 V	9,000 to 13,000 mV in 20m V steps	Used to set the lamp voltage for all six lamps on the board Usually set in field	No
VPI Debounce	40 ms	20 to 200 ms in 2 ms steps	Set VPI Debounce time	No
LampVoltage Reg	ContantLmp Voltage	ConstantLmp Voltage VariableLmp Voltage	The module will try and maintain a constant lamp voltage. If AC power is off this can cause the battery voltage to drain more rapidly. By setting this to VariableLmpVoltage the module will gradually reduce the lamp voltage as the battery voltage drops.	Yes
PCO1 Correspondence Time	300 ms	300 – 3,000 ms in 15 ms steps	When the mechanism 1 moves the executive software will wait this long before checking that the mechanism feedbacks are consistent with the new mechanism position	No
PCO2 Correspondence Time	300 ms	300 – 3,000 ms in 15 ms steps	Time for mechanism 2 - See above for explanation	No
UseMDI	Used	Used, NotUsed	Set this to NotUsed if you have the very usual case of a Searchlight that has no feedbacks	Yes

3.10.2 Lamps

The Lamps tab is used for the following parameters:

- To enter the logic states that are used to control the lamps' On and Flash states
- To enter logic state names that are set when a lamp reports a Light Out condition (LOR) or Foreign Energy (FE)

First, enter a channel name for each lamp used. Enter names for output controls to turn lamps on or flash them. Enter names for states that will be set if lamps detect Light Out (LOR) or Foreign Energy (FE).

Note: if the Searchlight module is not in session with the CPU (e.g. it has been removed, or has failed), the logic will set the reported states of the LOR and FE bits to true.

The logic will also set the reported states of MechFail to true and Red feedback to true.

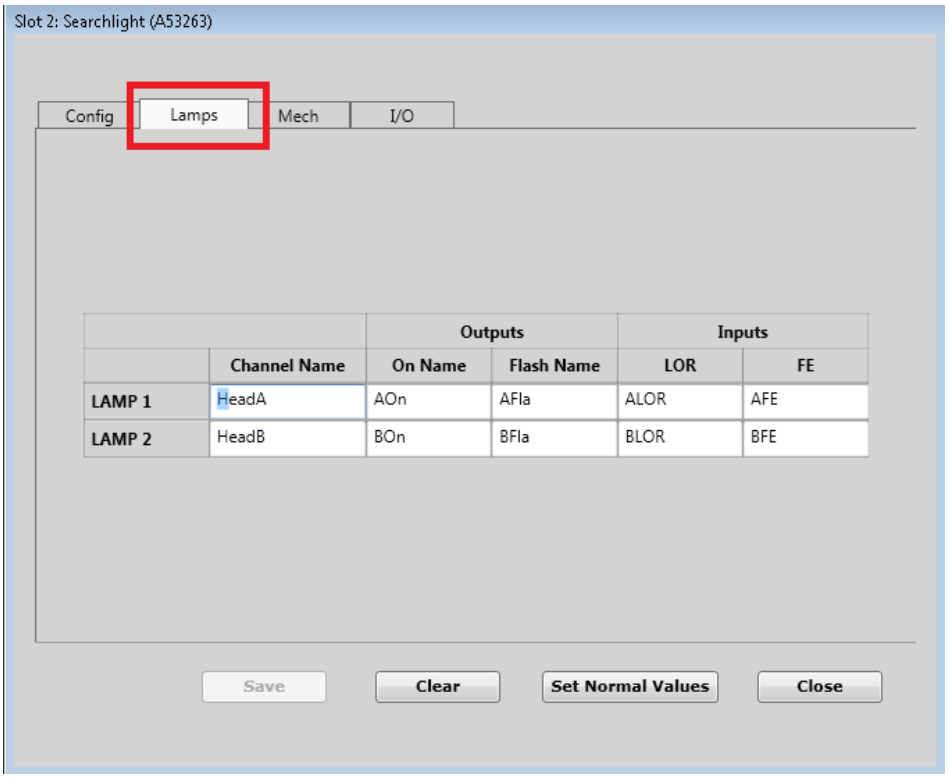


Figure 3-53 Searchlight Lamp Configuration

NOTE

- To flash a lamp, set the **Flash** state to true. The **On** state may be set to true or false.
- When a lamp detects it has a burned out filament, it will report the LOR state as true, and will turn off the lamp , even if the application logic still has the On state (or Flash) set true.
- When a lamp detects it has foreign energy applied to it, it will turn off the DC-DC convertor and all lamps will report LOR. If foreign energy is still detected on the lamp after all the lamps of this card have been turned off, it will report FE for that lamp.
- If the mechanism on the card has failed and the executive software on it cannot determine that the Searchlight is really in the red state, the module will turn off the lamp and ignore the lamp command from the logic.

3.10.3 Mech

The Mech tab is used for the following parameters:

- To enter the logic states that are used to control the mechanism position of the Searchlight
- To enter logic state names that are set depending on the mechanism feedbacks and mechanism fail state

First, enter a channel name for each used mechanism. Enter names for controls that will be used to drive the Searchlight mechanism to the positive (green) and negative (yellow) states. Enter names for states that will be set containing the mechanism feedback states. Enter a name for the fail state.

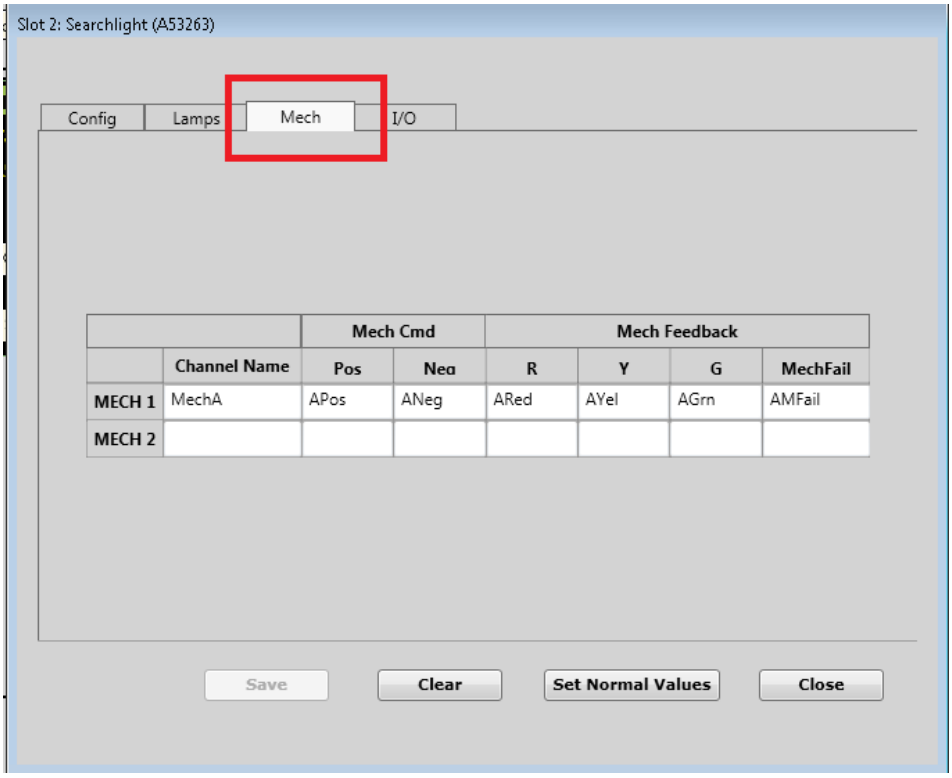


Figure 3-54 Searchlight Mechanism Configuration

NOTE

NOTE

- For a properly wired Searchlight mechanism, when the positive and negative controls are false, the mechanism will be at red and the red feedback bit will be set and the mech fail bit will be false.
- When the positive control is set to true, the mechanism will move to the green state and the green feedback bit will go true and the red one will go false. The mech fail bit will stay false.
- When the negative control is set true and the positive control is false, the mechanism will move to the yellow state and the yellow feedback bit will go true and the green and red one go false. The mech fail bit will stay false.

On start-up, the Searchlight performs a mechanism test where it will:

a) Keep the lamp off
b) Check that red feedback is true and green/yellow is false
c) Apply positive energy to the mechanism (i.e. move it to green position), and, after the configured PCO correspondence time, check that green feedback is true and red and yellow are false
d) Apply negative energy to the mechanism (i.e. move the mechanism to yellow), and, after the configured PCO correspondence time, check that yellow is true and red and green are false
e) Remove energy from mechanism, and, after the configured PCO correspondence time, check that red is true and green and yellow are false
f) If all the checks pass, the mech fail bit is set to false. The Searchlight is functional and the executive software will move the mechanism according to the command from the logic and turn on the lamp according to the lamp command (provided lamp is not LOR or FE detected)
g) If any of these checks fail, the mech fail bit is set true. If the red feedback can be shown to be working, i.e. that it can correctly indicate red when no energy is applied to mech, and not red when the mechanism moves away from red, the executive software will turn on the lamp if the logic is commanding it on, so the signal can show red. If the red feedback is not present, or cannot be trusted, as it won't move away from red, then the lamp will remain off
h) If a failure occurs with the mechanism drive correspondence after startup, the mech fail bit is set, and the Searchlight will remove energy from the mech drive to put it back to red (even if the logic is still commanding it positive or negative). Once in failure, the above startup test is rerun and has to pass before the module is set back to a healthy state and the mech fail cleared.

3.10.4 I/O

The Searchlight module has a Vital Relay Output and two Vital Parallel Inputs (VPI). Use the I/O screen to:

- Set controls for turning on the VRO, or set it to generate a Cab rate (see Coded Track section)
- Set logic states that will indicate the state of the VPIs

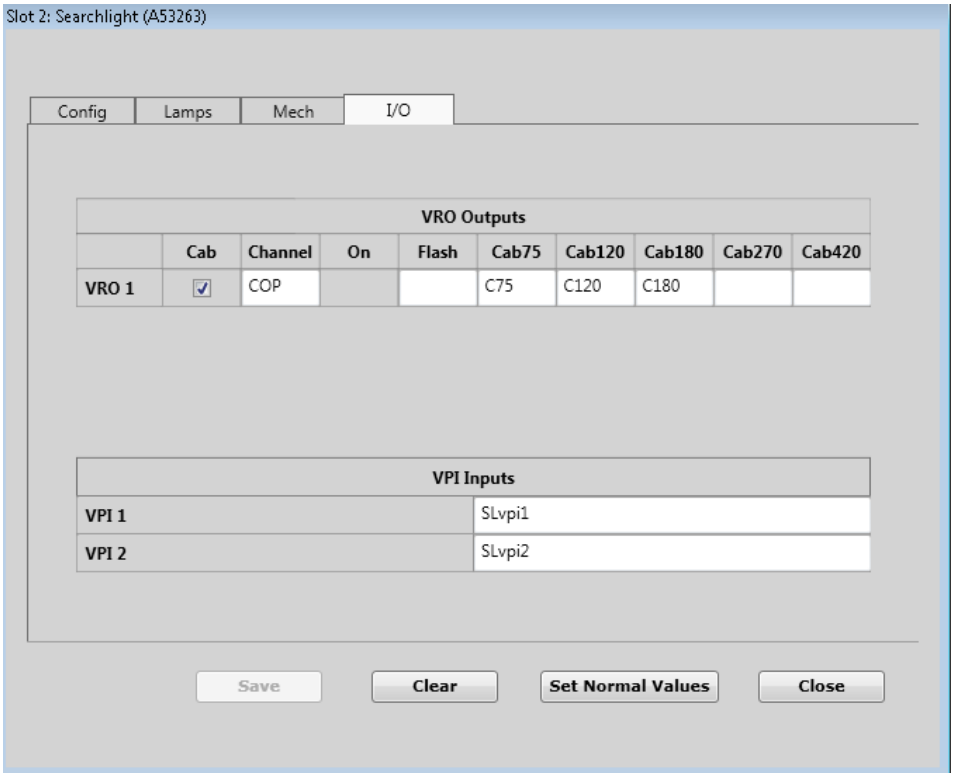


Figure 3-55 Searchlight I/O Configuration

3.11 VPI MODULE EDITOR

The VPI Module Editor has two tabs:

- Config
- Inputs

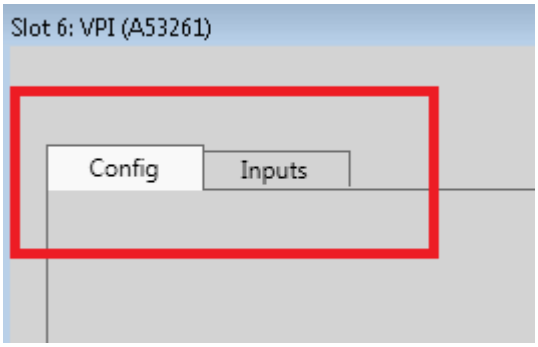


Figure 3-56 VPI Editor Tabs

3.11.1 Config

This tab allows the user to set the VPI debounce for the inputs.

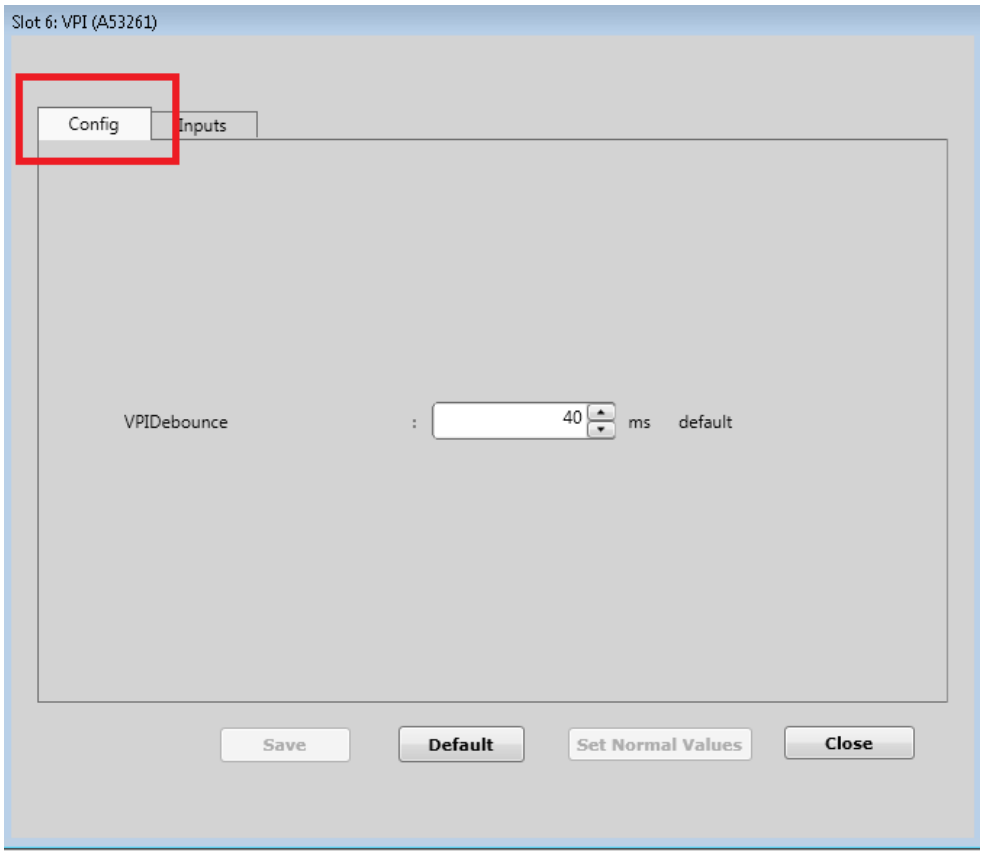


Figure 3-57 VPI Editor

The following table describes the VPI Configuration parameters.

PARAMETER	DEFAULT	RANGE	COMMENT	UCN
VPIDebounce	40 ms	20 to 200 ms in 2 ms steps	Set VPI Debounce time in milliseconds	No

3.11.2 Inputs

This tab allows the user to set logic state names for the eight VPIs on the module. Under failure, the VPI will always report de-energized.

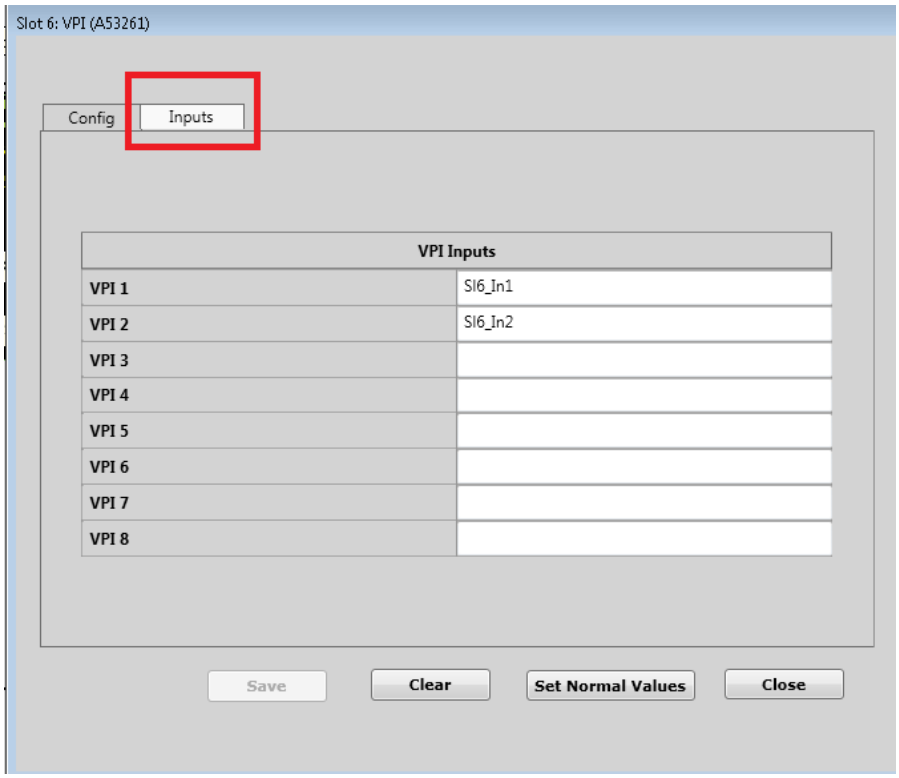


Figure 3-58 VPI Inputs

3.12 VRO MODULE EDITOR

The VRO Module Editor has a single tab: Outputs.

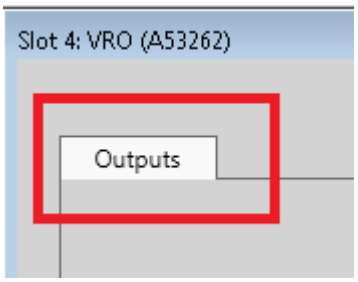


Figure 3-59 VRO Editor Tab

3.12.1 Outputs

This tab allows the user to set logic state names for the six VROs on the module. The VROs can be used as steady on/off or in Cab mode (see Coded Track section 3.6.3).

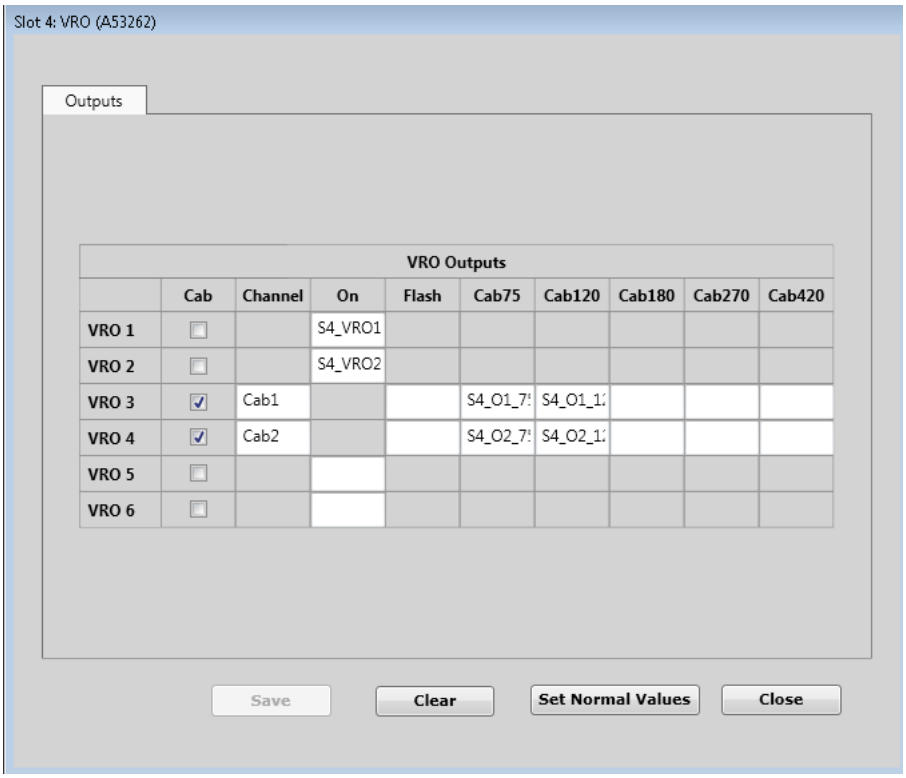


Figure 3-60 VRO Outputs

3.13 RIO MODULE EDITOR

The RIO Module Editor has three tabs:

- Config
- Outputs
- Inputs

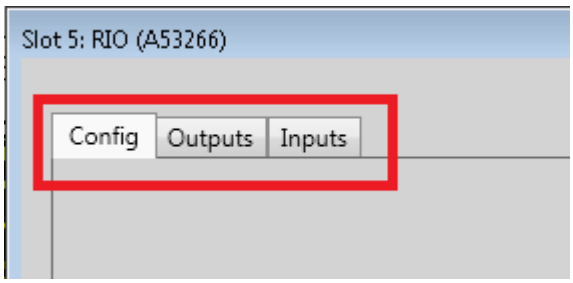


Figure 3-61 RIO Editor Tab

3.13.1 Config

This tab allows the user to set the VPI debounce for the inputs.

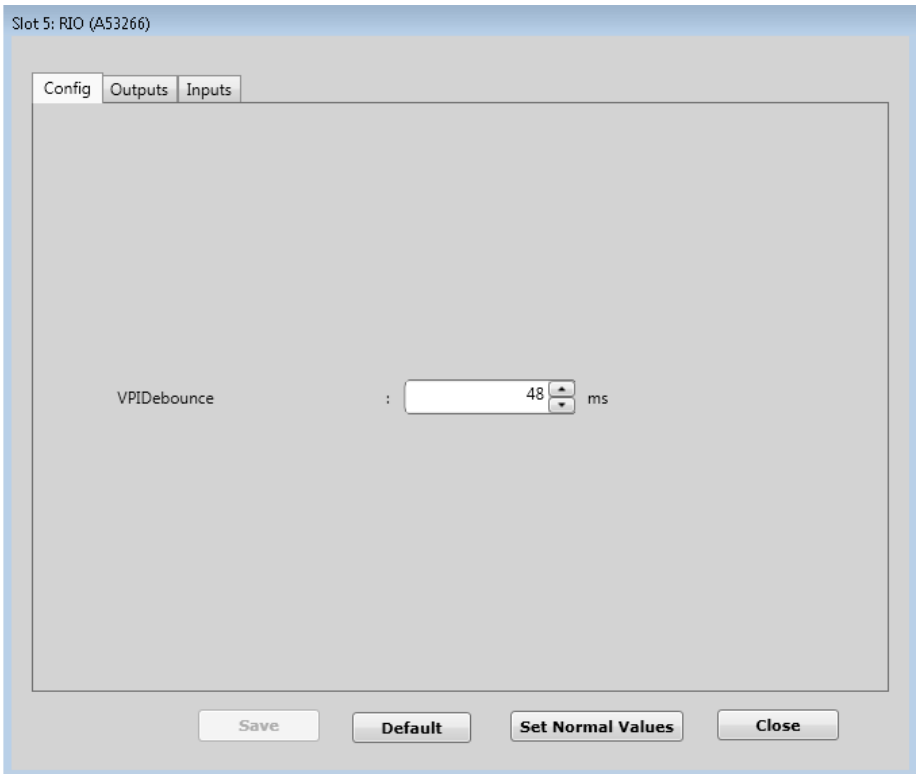


Figure 3-62 RIO Module Editor

The following table describes the RIO Configuration parameters.

Table 3-10 RIO Configuration Parameter

PARAMETER	DEFAULT	RANGE	COMMENT	UCN
VPIDebounce	40 ms	20 to 200 ms in 2 ms steps	Set VPI Debounce time in milliseconds	No

3.13.2 Outputs

This tab allows the user to set logic state names for the four VROs on the module. The VROs can be used as steady on/off, or in the Cab Mode (see Coded Track Section 3.6.3).

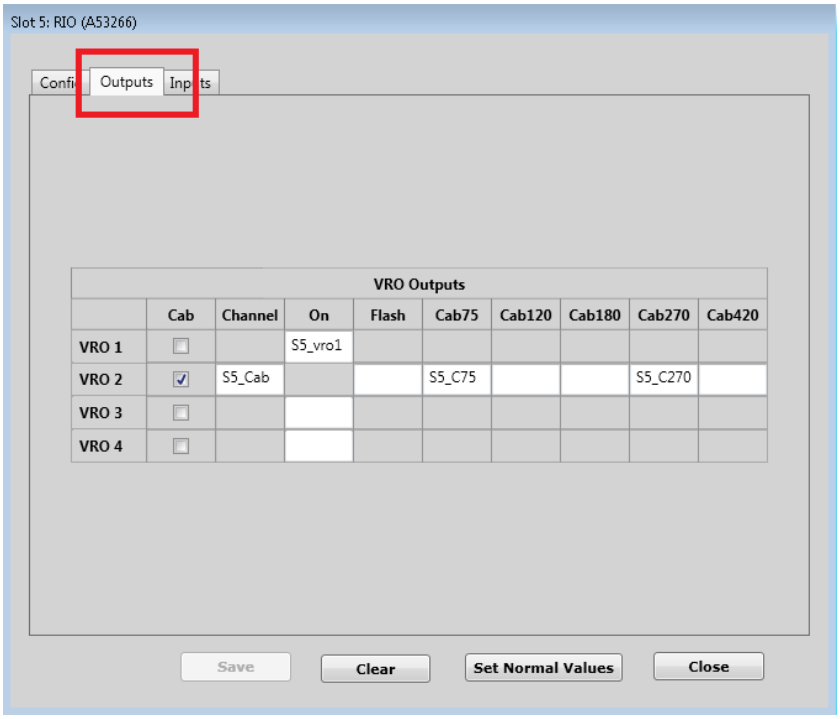


Figure 3-63 RIO VRO Outputs

3.13.3 Inputs

This tab allows the user to set logic state names for the four VPIs on the module. Under failure the VPI will always report de-energized.

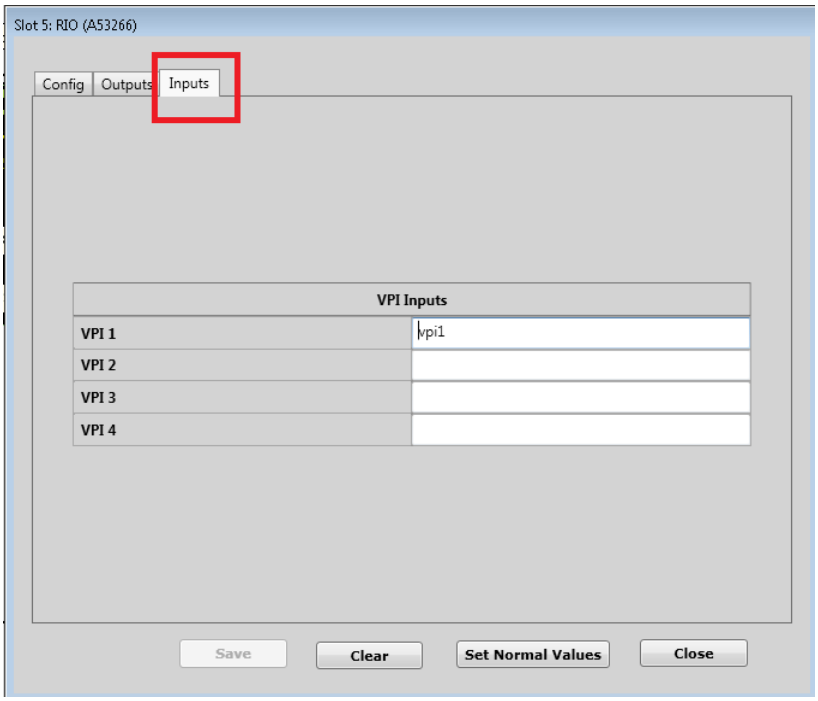


Figure 3-64 RIO VPI Inputs

3.14 PSO MODULE EDITOR

The PSO module has two main applications:

a) In Electrified territory it can be used as a replacement for Electrified Electrocode. A PSO module at one end of a track circuit can send user commanded codes down to a receiver at the other end of the circuit. Eighteen different PSO codes are available to the application, which is sufficient to provide a functional equivalent of Electrocodes, six vital codes (2,3,4,7,8,9), with and without code 5, code 1, and a tumbledown code 6. All PSO codes have equal vitality: all are considered vital. Unlike in Electrocode and Microtrax, the protocol is not half duplex. The PSO's on either end of the circuit can transmit different codes simultaneously. The PSOs on the same circuit must obviously use different frequencies. See WayConneX manual for more details. In this application set PSOMode to Code_Mode

b) Track circuit for detecting train occupancy on a switch (OS Track). In this application, one PSO module is used to detect OS occupancy. The transmitter is tied to one leg of the switch and the two receivers tied to the other two legs. The same frequency and code (Address) is used for the transmitter and two receivers. In this mode the code is selected as a configuration parameter rather than by being set by the application logic. In this application, set PSOMode to OS_Mode. If only one receiver is required, PSO receiver 2 can be disabled.

The PSO Module Editor has four tabs:

- Config
- PSO
- Outputs
- Inputs

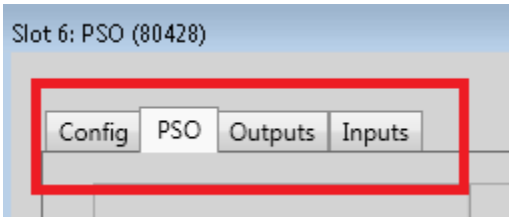


Figure 3-65 PSO Editor Tab

For further information on the PSO module, reference the WayConneX manual SIG-00-16-09.

3.14.1 Config

This tab allows the user to first set the PSOMode. Which other parameters are shown on the tab depends on the value of PSOMode. When PSOMode is set to Code_Mode, the user can set the configuration for the PSO transmitter, PSO receiver 1 and the VPI debounce, as shown in Figure 3-66.

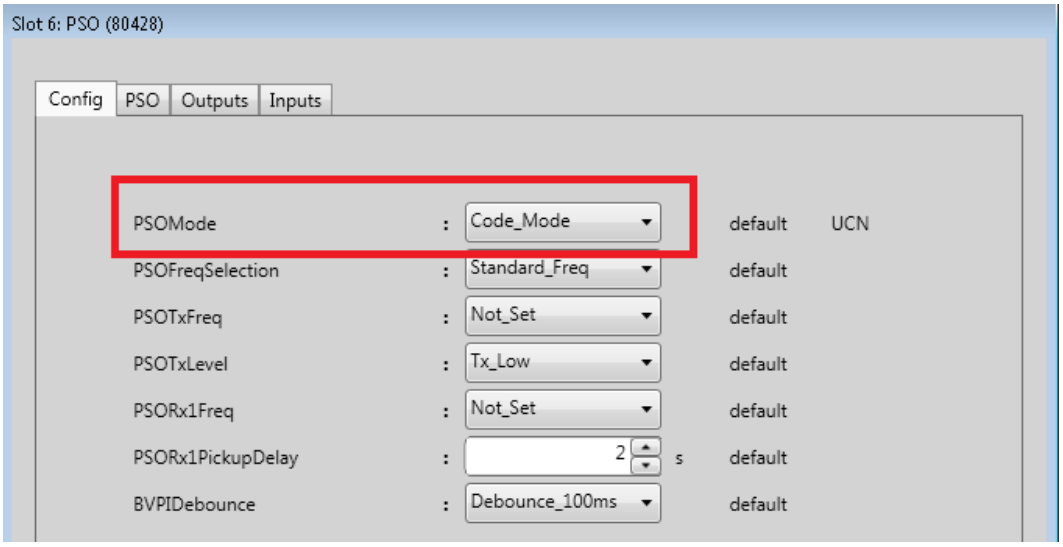


Figure 3-66 PSO Module Editor with PSOMode = Code_Mode

When PSOMode is set to OS_Mode, the user can set the configuration for the PSO transmitter, PSO receiver 1, PSO receiver 2, and the VPI debounce, as shown in Figure 3-67.

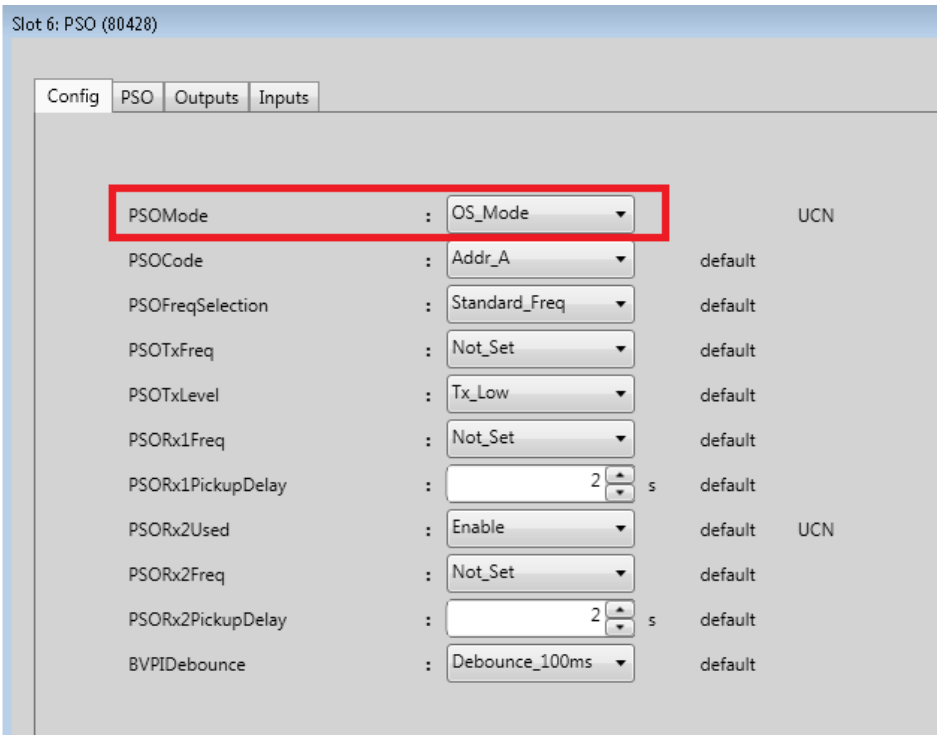


Figure 3-67 PSO Module Editor with PSOMode = OS_Mode

The following table describes the PSO Configuration parameters.

Table 3-11 PSO Module Parameters

PARAMETER	DEFAULT	RANGE	COMMENTS	UCN
PSOMode	Code_Mode	Code_Mode OS_Mode	Used to set the mode the PSO module is used in	Yes
PSOCode	Addr_A	Addr_A, Addr_C	Used to set the PSO code (address) sent by the transmitter, only applicable when PSOMode is OS_Mode	No
PSOFreq Selection	Standard_Freq	Standard_Freq Alternate_Freq	Used to limit the options for frequency observable when on real WayConneX System	No
PSOTxFreq	Not_Set	Not_Set, F_156 to F_20200	Used to set the PSO transmitter frequency	No
PSOTxLevel	Tx_Low	Tx_Low Tx_High	Used to set the transit level for the transmitter	No
PSORx1Freq	Not_Set	Not_Set F_156 to F_20200	Used to set the PSO receiver 1 frequency	No
PSORx1Pickup Delay	2	0-10	Used to set the PSO receiver 1 pickup delay	No
PSORx2Enable	Enable	Enable Disable	Used to enable or disable PSO receiver 2	Yes
PSORx2Freq	Not_Set	Not_Set, F_156 to F_20200	Used to set the PSO receiver 2 frequency. Only applicable when PSOMode is OS_Mode and receiver 2 is enabled	No
PSORx2Pickup Delay	2	0-10	Used to set the PSO receiver 2 pickup delay, Only applicable when PSOMode is OS_Mode and receiver 2 is enabled	No
BVPIDebounce	Debounce_100 ms	50, 100, 150, 200, and 250 ms	Used to set the debounce on the VPIs on the module	No

3.14.2 PSO

The layout of this tab depends on whether the PSOMode is Code_Mode or OS_Mode. In Code_Mode the tab allows the user to:

- set the names of the variables that are used to transmit a specific PSO code (code A to R)
- set the names of the variables that are set when the PSO receives a particular code (code A to R)
- the user can also add a name for the transmitter and receiver health state. Note the health state default to false when the module is not in session.

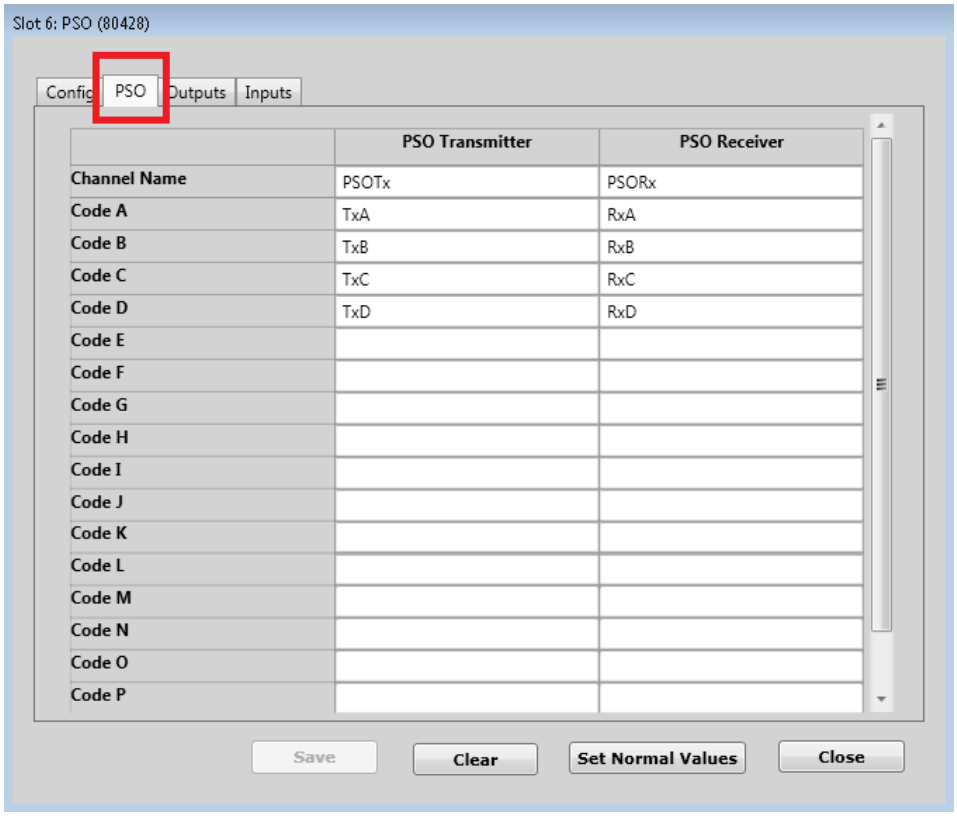


Figure 3-68 PSO Tab in Code_Mode

Use the scroll bar to scroll down to the Code Q and Health bits as shown below.

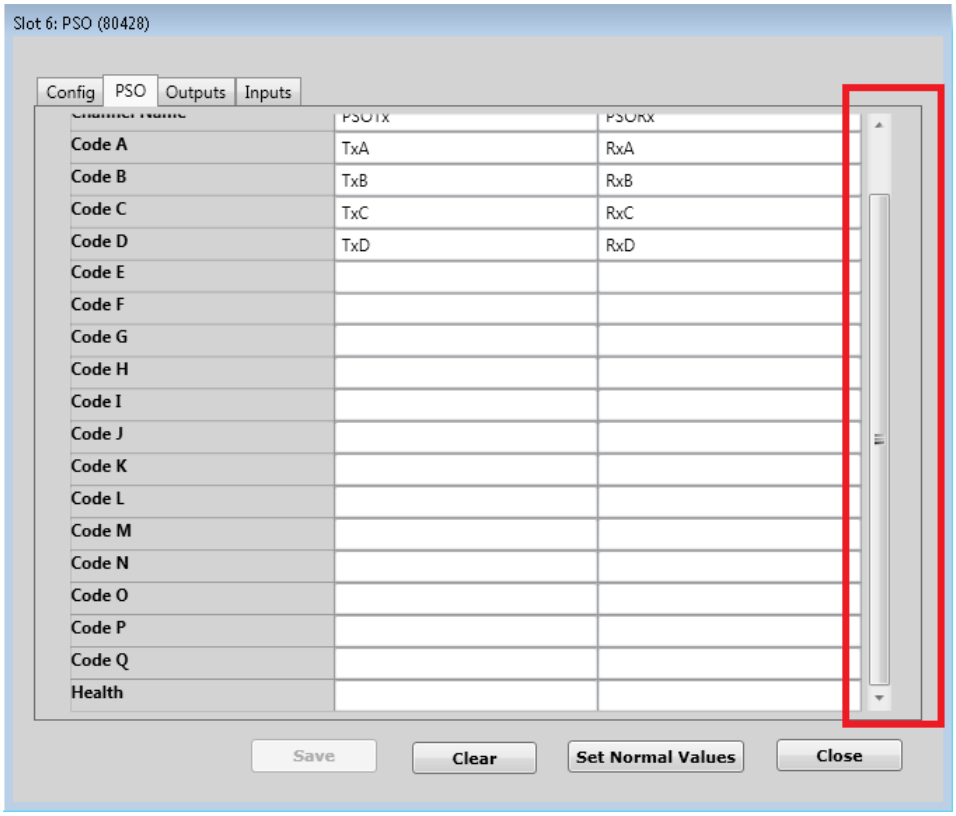


Figure 3-69 PSO Tab in Code_Mode (continued)

In OS_Mode the tab allows the user to:

- set the name for the transmit enable state,
- set the names of the variables that are set that represent the PSO receiver 1 or receiver 2 **Occupancy** or **Occupancy with Pickup**.
- set the names of the variables that are set that represent whether the PSO receiver 1 or receiver 2 require calibration
- set the name for the transmitter and receiver health state.

Notes:

- The transmit enable state must be set to true for the transmitter to transmit a code.
- When the track circuit is occupied, the receiver uncalibrated, or the receiver unhealthy, **Occupancy** and **Occupancy with Pickup** are both false;
- When the track circuit becomes unoccupied (providing receiver healthy and calibrated) **Occupancy** becomes true and the configured receiver pickup delay for the receiver starts.
- **Occupancy with Pickup** becomes true when the receiver pickup delay timer elapses
- **Cal Reqd** is true when the receiver requires calibration and false when it is calibrated
- When the PSO module is not in session with the CPU:
    - The Tx and RX health states will be set to false
    - **Cal Reqd** will be set to true
    - **Occupancy** and **Occupancy with Pickup** will be false

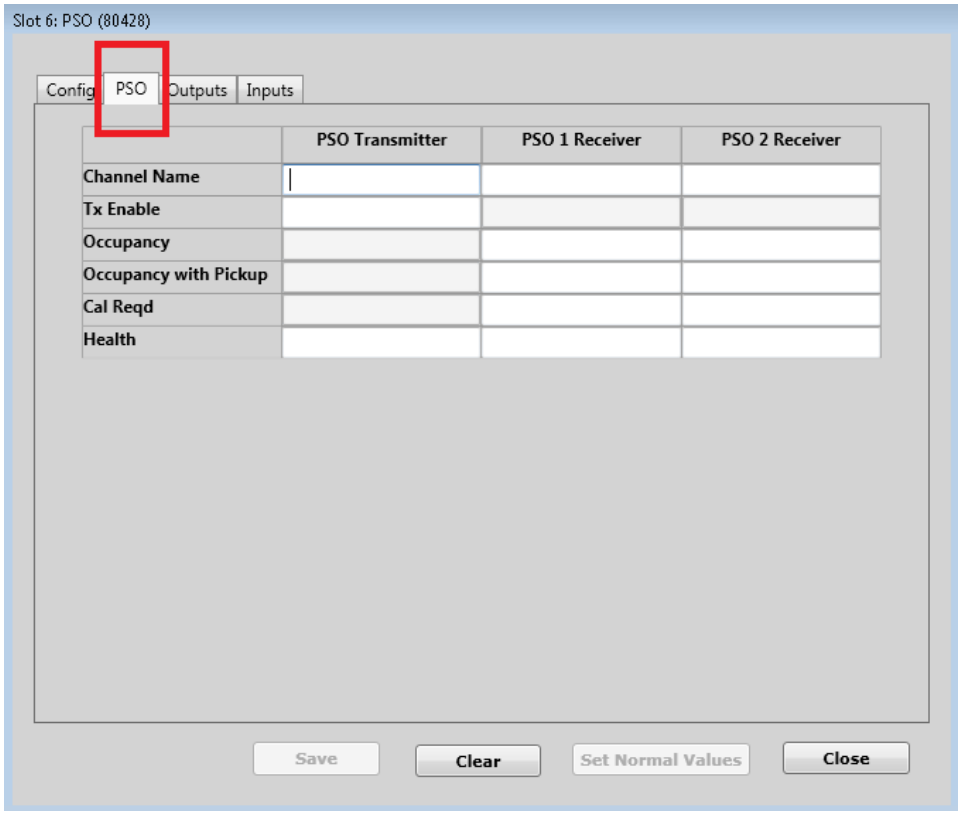


Figure 3-70 PSO Tab in OS_Mode

3.14.3 Outputs

This tab allows the user to set logic state names for the three VROs on the module.

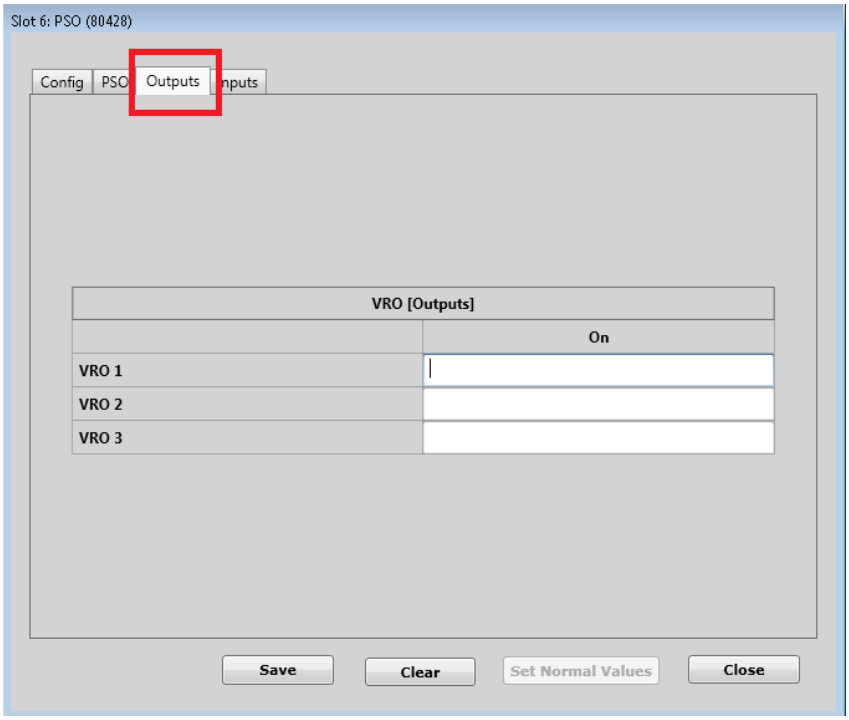


Figure 3-71 PSO VRO Outputs

3.14.4 Inputs

This tab allows the user to set logic state names for the two VPIs on the module. Under failure, the VPI will always report de-energized.

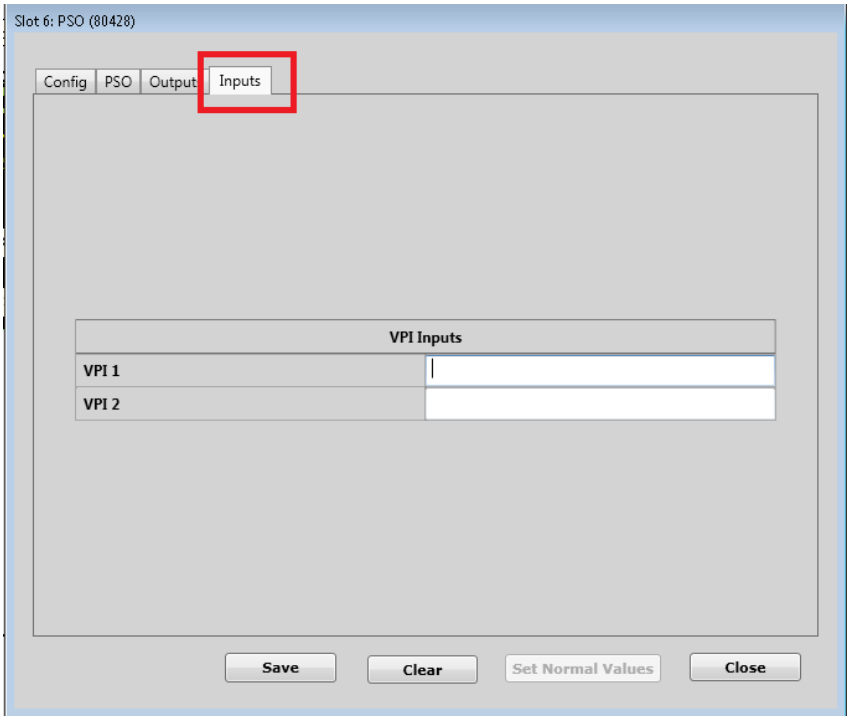


Figure 3-72 PSO VPI Inputs

3.15 BVPI8 MODULE EDITOR

The BVPI8 Module Editor has three tabs:

- Config
- Outputs
- Inputs

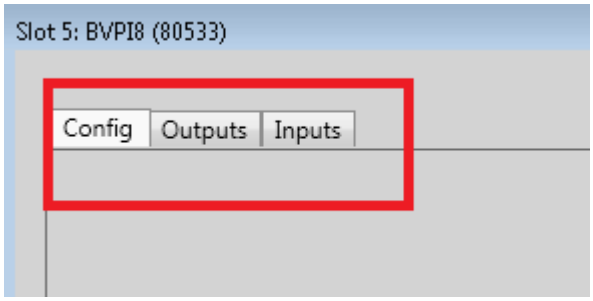


Figure 3-73 BVPI8 Editor Tab

3.15.1 Config

This tab allows the user to set the VPI debounce for the inputs.

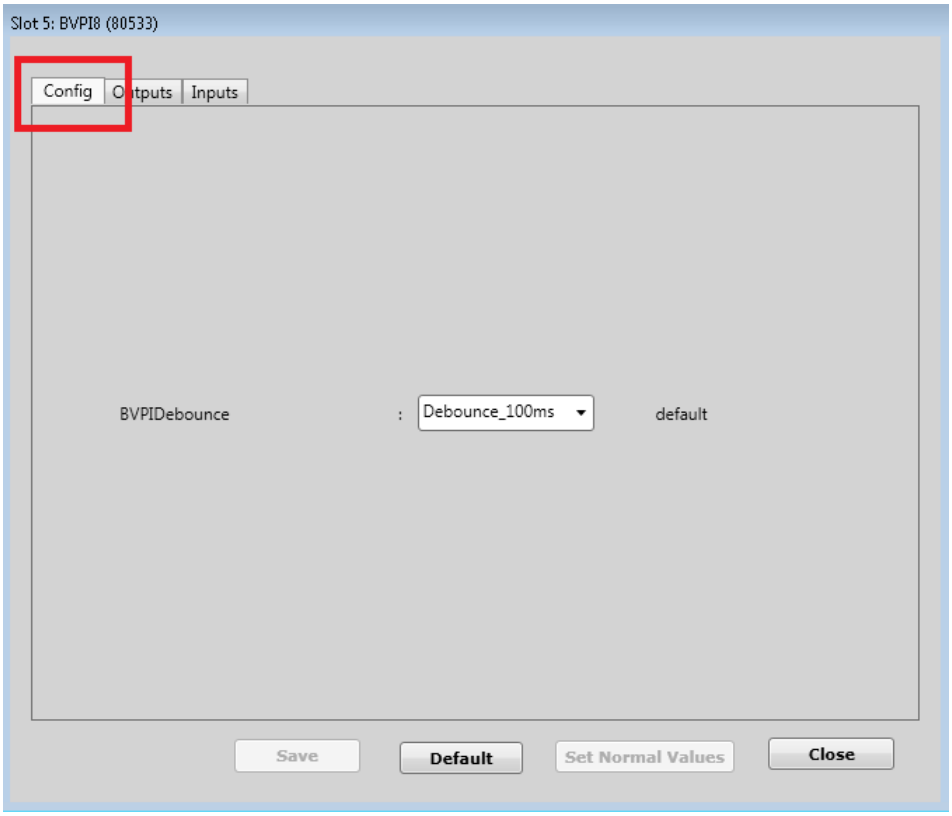


Figure 3-74 BVPI8 Configuration Parameter

The following table describes the BVPI8 configuration parameters.

Table 3-12 BVPI8 Configuration Parameter

PARAMETER	DEFAULT	RANGE	COMMENTS	UCN
BVPIDebounce	100 ms	50, 100, 150, 200, 250 ms	This is used to set the debounce on the inputs	No

3.15.2 Outputs

This tab allows the user to set logic state names for the two VROs on the module.

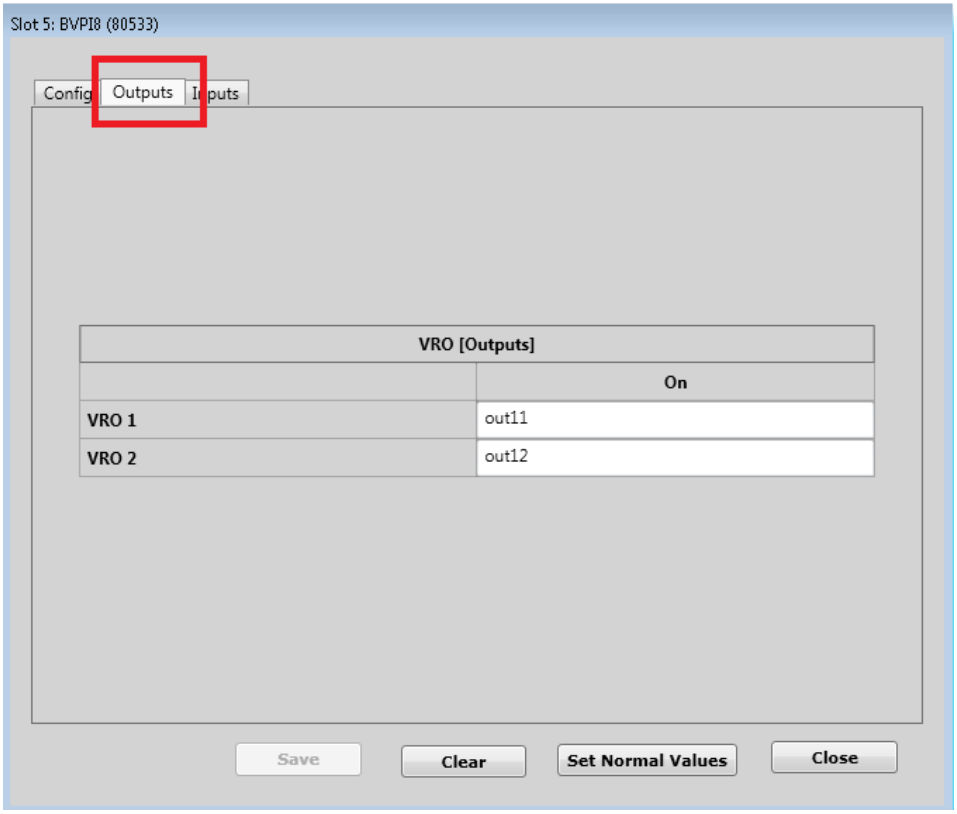


Figure 3-75 BVPI8 VRO Outputs

3.15.3 Inputs

This tab allows the user to set logic states names for the eight Bipolar VPIs on the module.

The On (pos) is the state set when the input has a positive voltage applied, and the On (neg) is the state set when the input has a negative voltage applied.

Note: under failure the VPI will always report de-energized and Fail bit will be set.

The health state defaults to true when the module is not in session.

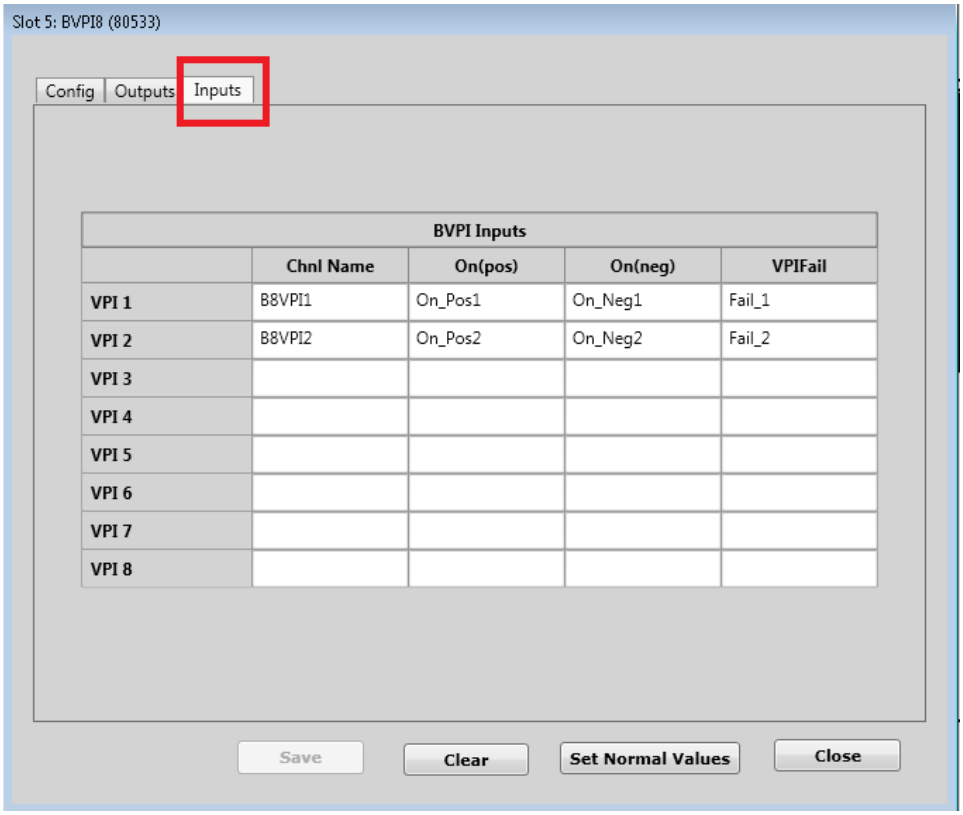


Figure 3-76 BVPI8 VPI Inputs

3.16 BVPI10 MODULE EDITOR

The BVPI8 Module Editor has two tabs:

- Config
- Inputs

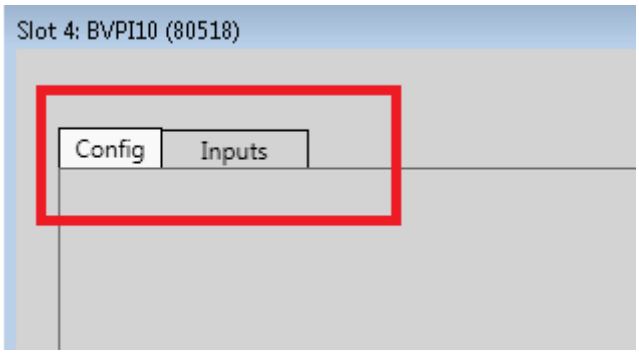


Figure 3-77 BVPI10 Editor Tab

3.16.1 Config

This tab allows the user to set the VPI debounce for the inputs.

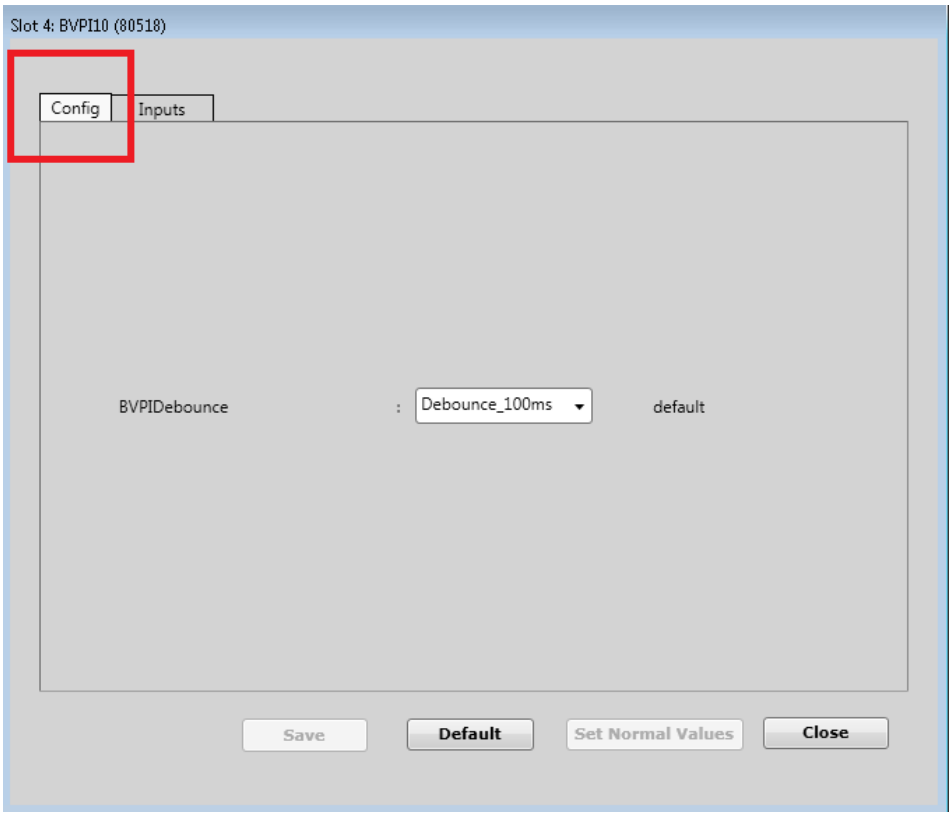


Figure 3-78 BVPI10 Configuration Parameters

The following table describes the BVPI10 Configuration parameters.

Table 3-13 BVPI10 Configuration Parameter

PARAMETER	DEFAULT	RANGE	COMMENT	UCN
BVPIDebounce	100 ms	100, 150, 200, 250 ms	This is used to set the debounce on the inputs	No

3.16.2 Inputs

This tab allows the user to set logic states names for the 10 Bipolar VPIs on the module.

The On (pos) is the state set when the input has a positive voltage applied, and the On (neg) is the state set when the input has a negative voltage applied.

NOTE

NOTE

Under failure, the VPI will always report de-energized and a Fail bit will be set. The health state defaults to true when the module is not in session.

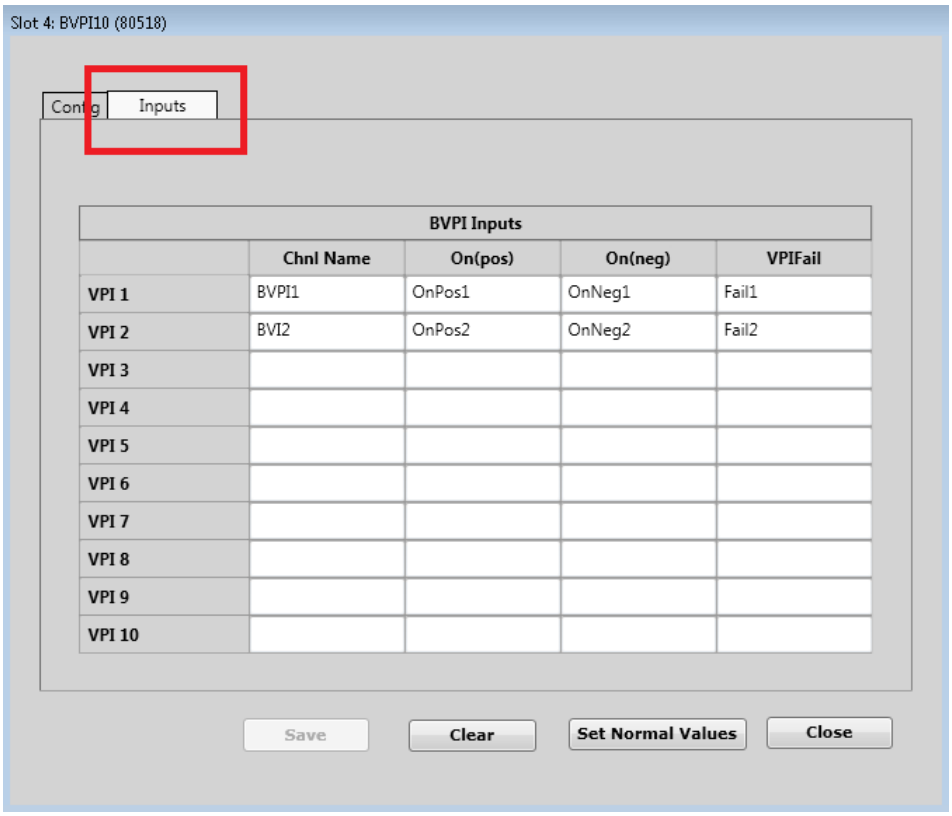


Figure 3-79 BVPI10 VPI Inputs

3.17 LOGIC DATA EDITOR

To enter the Logic Data Editor, either select the **Logic Data** icon, or from **Edit>Logic Data** or use the shortcut **Ctrl L**. The Editor will open at the Boolean Variables window.

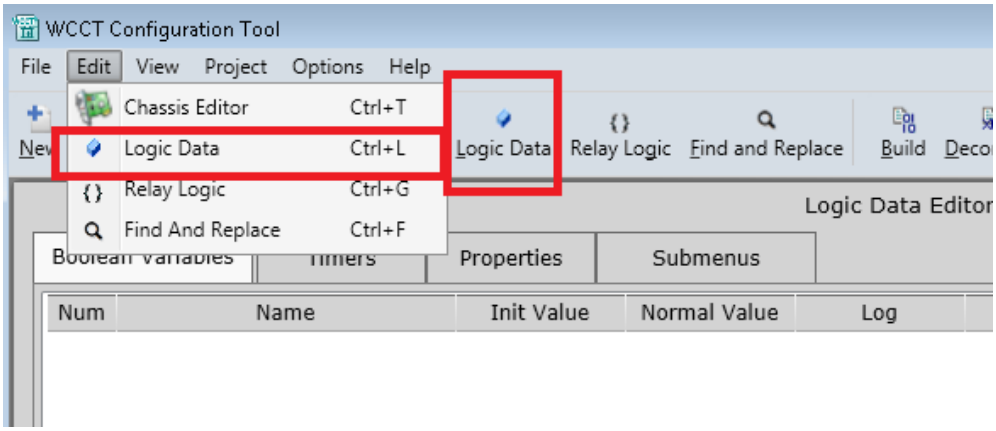


Figure 3-80 Selecting Logic Data Editor

3.17.1 Boolean Variables Editor

This editor is used to enter Boolean variables that are internal to the logic, i.e. they are not outputs or inputs. A logic equation should be written to set the state of each Boolean variable and in general, logic equations are written read the state of the variable. The only exception to this would be if the state was used for logging purposes only, in which case it would be set but not read.

Use the **Add** button to add new states to the end of the list.

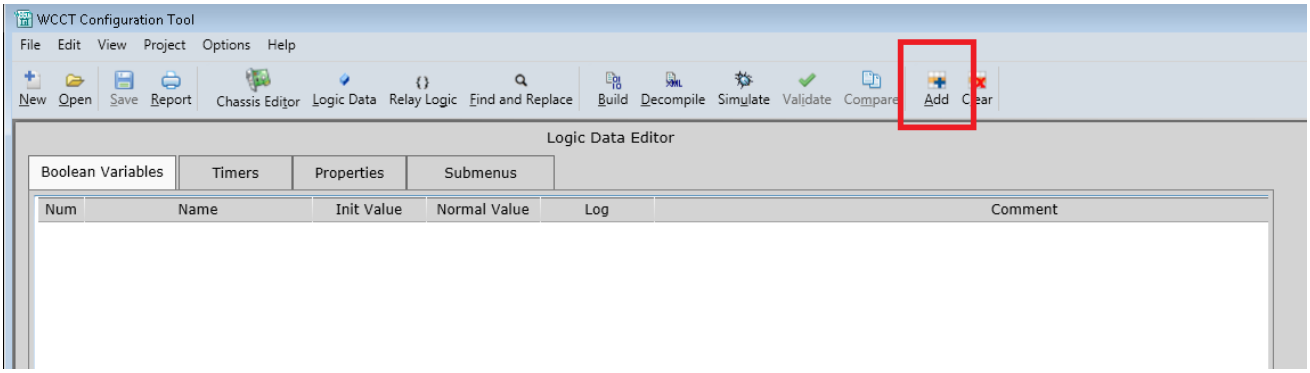


Figure 3-81 Boolean Variables Editor

Right-click on the Num column of an existing state and use **Insert Before** or **Insert After** to insert a new state in the middle of the list.

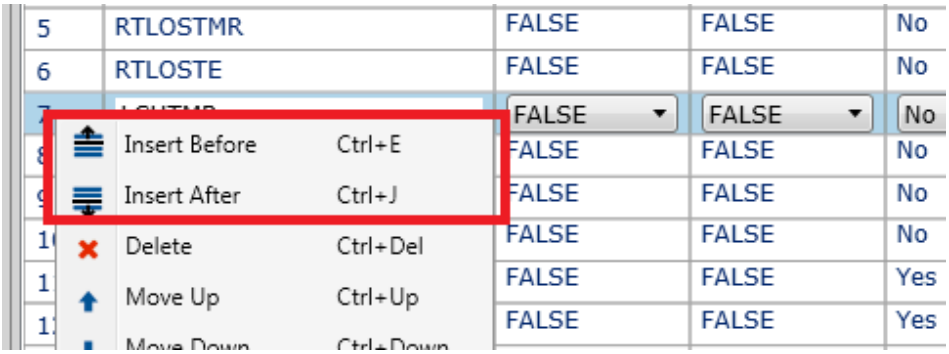


Figure 3-82 Insert Menu

For each variable enter:

- the variable name,
- the initial value (default is false),
- the normal value which is the state of the contact/coil shown in the relay logic,
- whether the variable should be logged on change of state in the CPU log,
- a comment can be added if necessary next to each variable, multi-line comments are allowed.

NOTE

NOTE

a) The name can to be up to 16 characters long, and contain only upper and lower case letters, number and '_"
b) The name must contain a letter or '_', it cannot be all numbers
c) The name cannot be a duplicate of a logic state name entered anywhere else, i.e. I/O module editors, CPU editors, Logic Data Editor,
d) Though upper and lower case names can be entered, the duplicate checking is not case sensitive, so the name Input1 is treated as a duplicate of INPUT1 and input1.

To rearrange the list of variables, right-click in the **Num** column and use the **Move up**, **Move Down**, **Cut**, **Paste**, or **Delete** menus shown below. The short cuts for these actions are shown below.

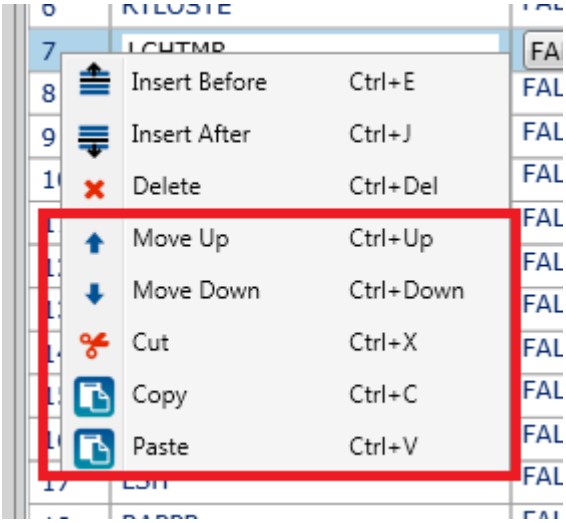


Figure 3-83 Rearrange Variables Menu

An easy way to add similarly named variables is to use Copy/Paste using short cuts **Ctrl C** and **Ctrl V** and edit the copied values to the correct values.

The WCCT allows up to 3000 variables. The names can be up to 16 characters long. If a VLP3 is used, up to 300 variables can be marked as being logged. If a VLP2 is used, up to 50 variables can be marked as being logged.

NOTE

NOTE

Currently, the WCCT allows you to set more than these as logged in the Logic Data Editor; however, the limit error is detected when the project is parsed.

Each name has to be filled in, otherwise, an error message is shown when attempting to save. Right-click and use **Delete** to delete an empty row that is not required.

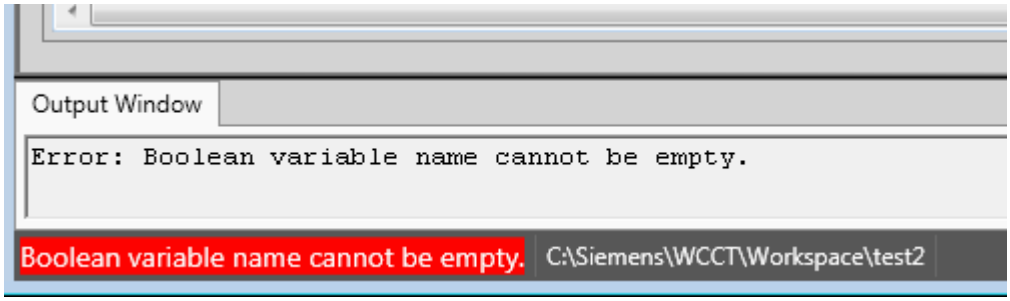


Figure 3-84 Deleting Rows

The names have to be unique, or the WCCT will give an error.

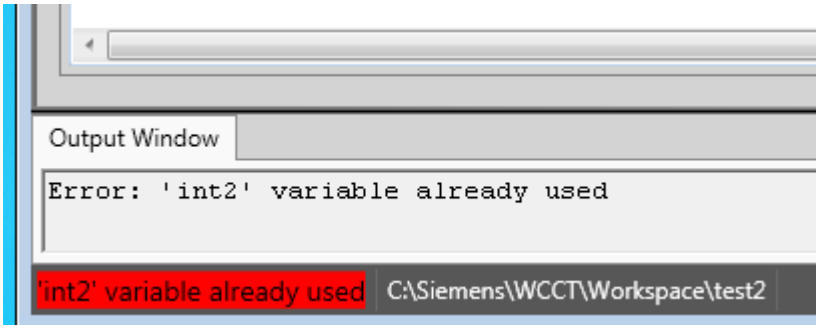


Figure 3-85 Duplicate Variable Error Message

To delete a variable, select the variable to be deleted and then right-click in the Num column and select **Delete.**

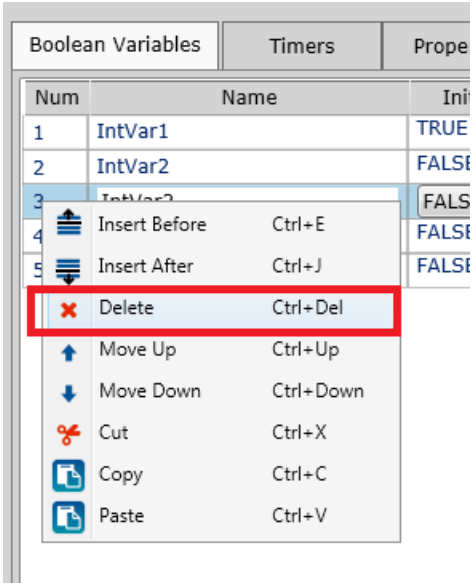


Figure 3-86 Delete Variable

To change the order of the internal variables, right-click in the Num column of the variable and select **Move Up**, **Move Down**, **Cut**, or **Paste** as required.

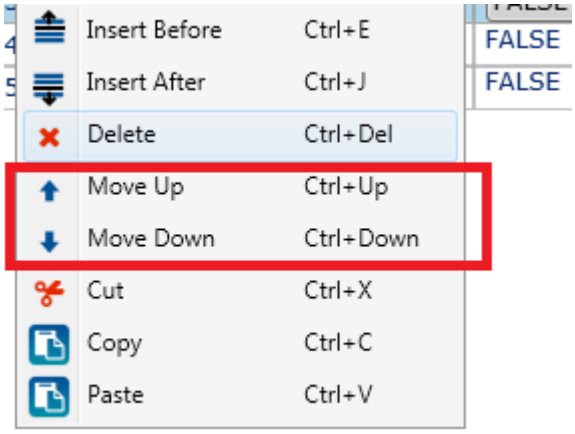


Figure 3-87 Change Variable Order Menu

To Clear all the variable names, select the Clear icon.

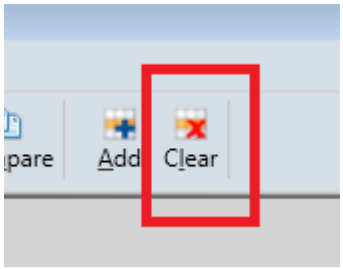


Figure 3-88 Clearing Logic Data Menu

To find a specific variable or to do a string replace, select the **Find/Replace** function. See Find and Replace section for details.

3.17.2 Timers Editor

Select the timer tab, then press **Add** to add new timers.

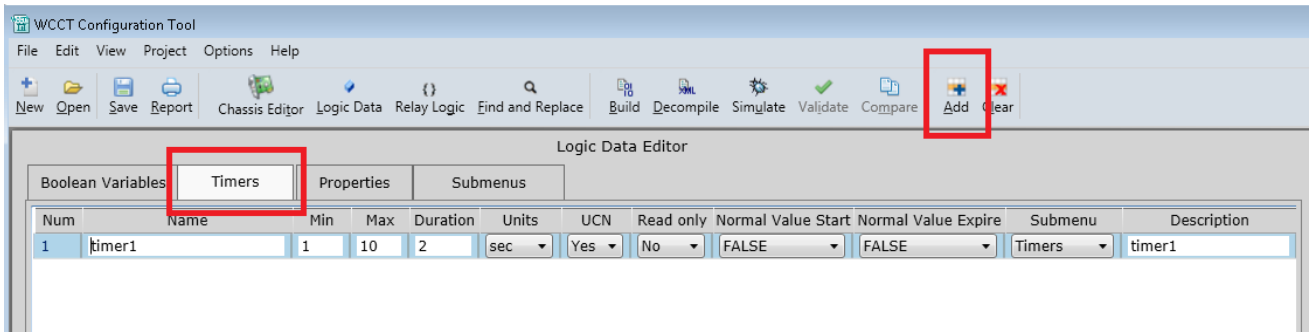


Figure 3-89 Timers Editor

For each timer, enter:

- Name (this is the Name that will be used in the logic equations, upper and lower case letters, numbers, and '_' allowed, max length 20 characters, must not start with a number).
- Min/Max – minimum and maximum timer values
- Duration in seconds or milliseconds (depending on units)
- Units –select either seconds or milliseconds. Note that if milliseconds is selected as the units the timer duration and range have to be entered in 100ms increments. The maximum values of timers are 3600s, 3600ms.
- Select whether the timer value is included in the UCN calculation or not.
- If it is required to change the timer value in the field, set **Read Only** to **No**,
- Set **Normal Value Start** to set the state of the timer coil displayed in the relay logic.
- Use **Normal Value Expire** to set the state of the timer contact displayed in the relay logic.
- Use the submenu field to select which menu the timer is displayed on when the user views the configuration menus with the Web UI or DT when connected to the CPU. The menu will default to **Timers**.
- Enter a description for the timer. The description is the name the timer will have in the user interface menus. A description is required. If the user leaves the description blank, when they press **Save** the WCCT will automatically set the description to the timer name. The Name is up to 40 characters and can have , upper and lower case letters, numbers, spaces, most standard punctuation (, . ; etc.) and '_' allowed, max length 40 characters.
- A comment can be added for each timer if required. The comment is optional.

⚠ WARNING

WARNING

TIMERS WHOSE INCORRECT SETTING COULD CAUSE A SAFETY ISSUE MUST BE INCLUDED IN THE UCN TO PREVENT UNAUTHORIZED CHANGES.

The DT or OCE (for GEO) or OCE (For WayConneX) will be used to define the timer and property settings for a particular installation of an MCF, and obtain the required UCN.

The VLP3 used in WayConneX supports up to 120 timers and the VLP2 using in GEO up to 32 timers.

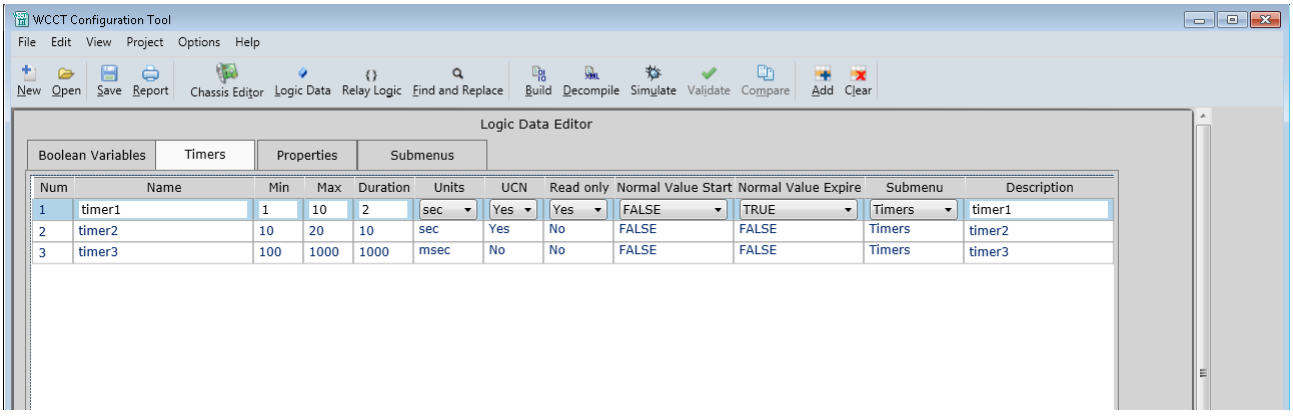


Figure 3-90 Timer Configuration

To change the order of the timers listed, right-click on the Num column and use the **Move Up**, **Move Down**, **Cut** and **Paste** options. To add a new timer at a specific place in the timer list, use the **Insert Before** or **Insert After** options. To delete a timer, use the **Delete** option from this menu.

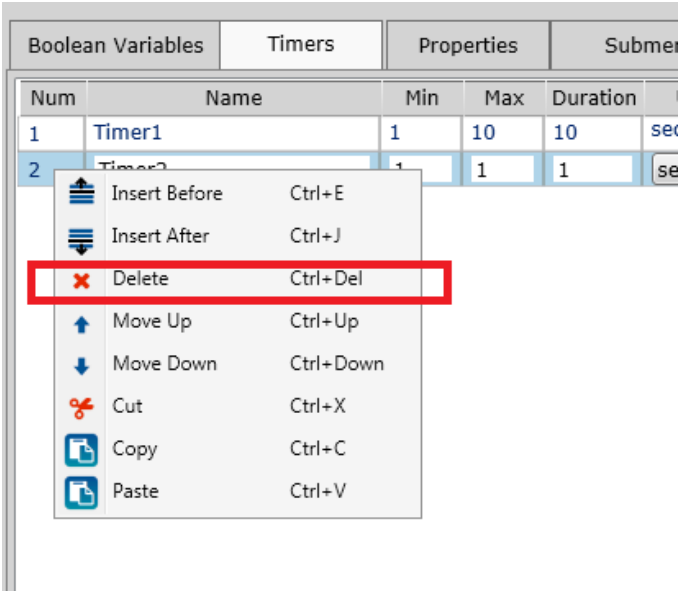


Figure 3-91 Delete a Timer

An easy way to add a similarly named timer is to use the Copy/Paste options (**Ctrl C** and **Ctrl V**) and then edit the copied timer to the correct values. All configurable timers will appear on one menu screen along with properties, unless otherwise specified. See Submenu Section 3.18 for details on how to create submenus.

NOTE

NOTE

The CPU II+ supports a maximum of 16 configurable items per page. If there are more than 16 configurable timers, you will need to create submenus with 16 or less items per page.

The CPU III supports a maximum of 25 configurable items per page. If there are more than 25 configurable timers, you will need to create submenus with 25 or less items per page.

3.17.3 Properties Editor

If you want to create an application that can be used in multiple field sites, it may require different field configurable settings to tailor the logic to that specific site. To create field configurable properties select the **Properties** tab on the Logic Data Editor. Click the **Add** button to enter new properties.

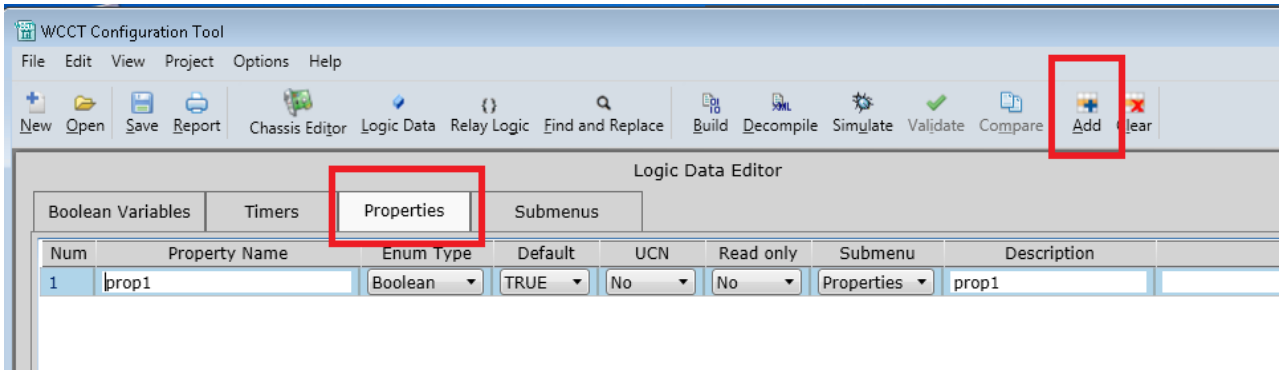


Figure 3-92 Properties Editor

For each property:

- Enter the property name – this is what will be used in the logic to reference the state of this property in the logic. The name is up to 40 characters, upper or lower case letters, numbers or underscores (_).
- The enum type will default to Boolean which is currently the only supported type.
- Select the required default value of the property from the drop down menu, the default value is true.
- Select whether the property value is included in the UCN calculation or not.
- Set **Normal Value** to indicate how the Property relay contact will be displayed in the relay logic.
- Set the **Read Only** column to **Yes** if you do not want the user to change the property value in the field.
- Use the submenu field to select which menu the Property is displayed on when the user views the configuration menus with the Web UI or DT when connected to the CPU. The menu will default to **Properties**.
- Enter a description for the property. The description is the name the property will have in the user interface menus. A description is required. If the user leaves the description blank when they press **Save**, the WCCT will automatically set the description to the property name. The name is up to 40 characters and can have: upper and lower case letters, numbers, spaces, most standard punctuation (, . ; etc.), and underscores (_). Max length 40 characters.
- A comment can be added for each property if required. The comment is optional.

⚠ WARNING

WARNING

PROPERTIES WHOSE INCORRECT SETTING COULD CAUSE A SAFETY ISSUE MUST BE INCLUDED IN THE UCN TO PREVENT UNAUTHORIZED CHANGES.

The DT or OCE (for GEO) or OCE (For WayConneX) may be used to change the default timer and property values set in the MCF to configure them for a particular installation of an MCF, and obtain the required UCN.

The WCCT supports up to 100 properties.

To change the order of the way the properties are listed, right-click on the Num column and use the **Move Up**, **Move Down**, **Cut** and **Paste** options. To add new properties at a specific place in the properties list, use the **Insert Before** or **Insert After** options. To delete a property, use the delete option from this menu. Short cuts are available for these operations as shown below.

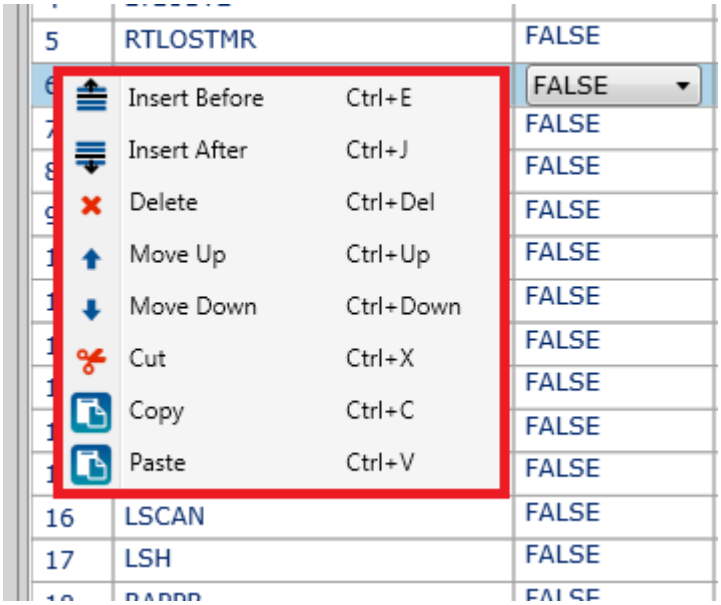


Figure 3-93 Change Order of Properties

An easy way to add a similarly named property is to use the Copy/Paste options (**Ctrl C** and **Ctrl V**) and then edit the copied properties to the correct values.

All properties will appear on the properties menu screen unless otherwise specified, see Submenu Section 3.18 for details on how to create submenus.

NOTE

NOTE

The CPU II+ supports a maximum of 16 configurable items per page. If there are more than 16 configurable timers and properties in an application chassis, you will need to create submenus with 16 or less items per page.

The CPU III supports a maximum of 25 configurable items per page. If there are more than 25 configurable timers and properties in an application, you will need to create submenus with 25 or less items per page.

3.18 CREATING SUBMENUS FOR CONFIGURABLE VALUES

By default the WCCT will put timers on a submenu called Timers, and Properties on a submenu called Properties.

To create submenus for the properties, select the Submenu tab in the Logic Data Editor and use the **Add** icon to add new submenus.

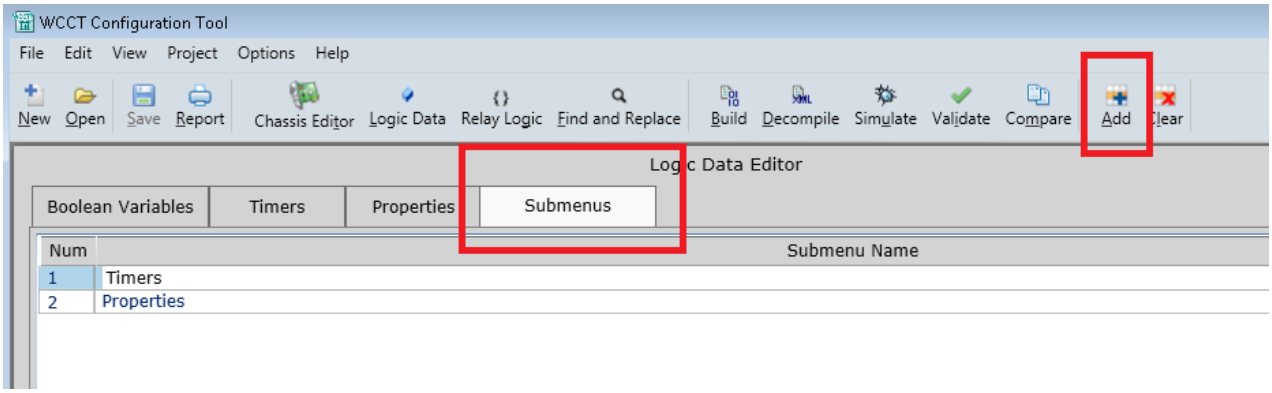


Figure 3-94 Submenu Screen

The Timers and Properties submenus are created by default and cannot be deleted or moved. New submenus can be added. The following example shows that three new submenus have been added to the submenu tab.

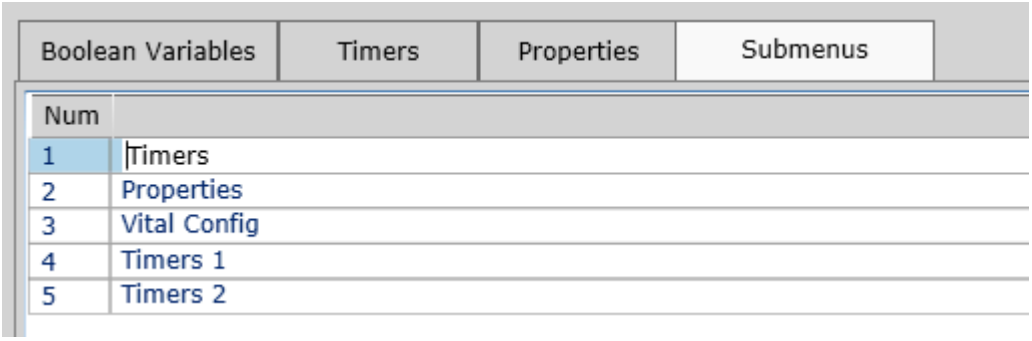


Figure 3-95 New Submenus

Once the submenus are created, go back to the Timer and Property tabs and assign the timers and properties to the new submenus as follows:

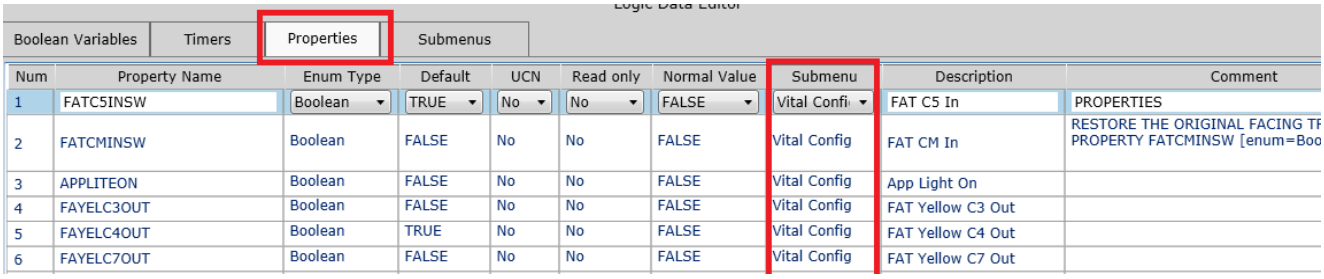
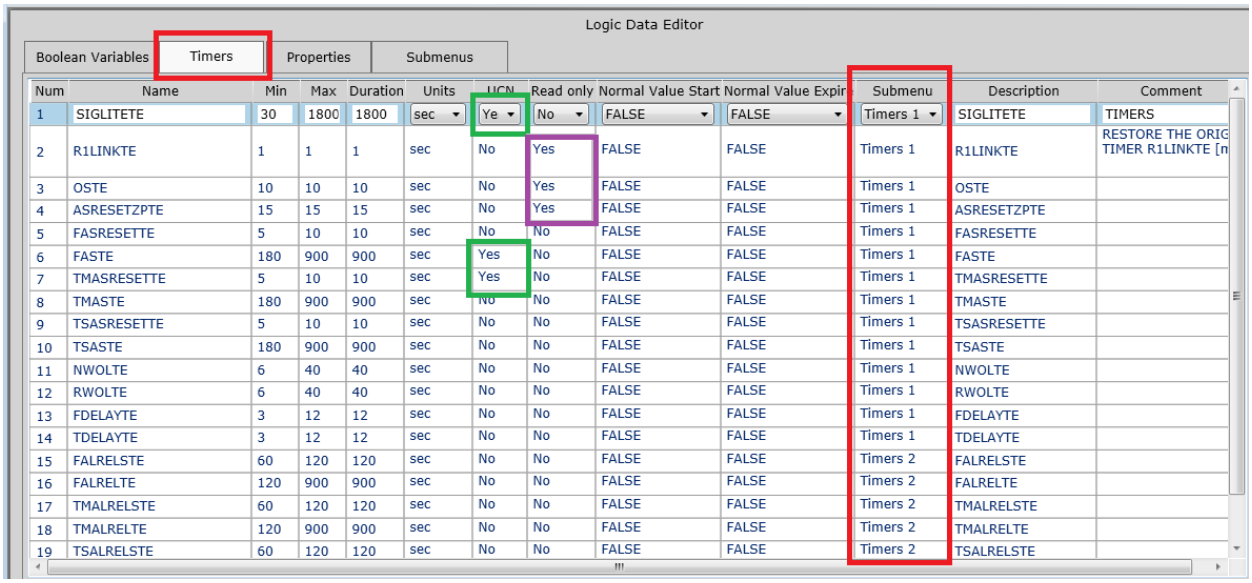


Figure 3-96 Assign Submenu on Property Page

In the following graphic, some Logic Data timers have been assigned to the submenus Timer 1 and Timer 2. Note that some timers have been set as **Read Only** (purple box) and as **UCN protected** (green box).



Num	Name	Min	Max	Duration	Units	UCN	Read only	Normal Value Start	Normal Value Expir	Submenu	Description	Comment
1	SIGLITETE	30	1800	1800	sec	Ye	No	FALSE	FALSE	Timers 1	SIGLITETE	TIMERS
2	R1LINKTE	1	1	1	sec	No	Yes	FALSE	FALSE	Timers 1	R1LINKTE	RESTORE THE ORIG TIMER R1LINKTE [n
3	OSTE	10	10	10	sec	No	Yes	FALSE	FALSE	Timers 1	OSTE	
4	ASRESETZPTE	15	15	15	sec	No	Yes	FALSE	FALSE	Timers 1	ASRESETZPTE	
5	FASRESETTE	5	10	10	sec	No	No	FALSE	FALSE	Timers 1	FASRESETTE	
6	FASTE	180	900	900	sec	Yes	No	FALSE	FALSE	Timers 1	FASTE	
7	TMASRESETTE	5	10	10	sec	Yes	No	FALSE	FALSE	Timers 1	TMASRESETTE	
8	TMASTE	180	900	900	sec	No	No	FALSE	FALSE	Timers 1	TMASTE	
9	TSASRESETTE	5	10	10	sec	No	No	FALSE	FALSE	Timers 1	TSASRESETTE	
10	TSASTE	180	900	900	sec	No	No	FALSE	FALSE	Timers 1	TSASTE	
11	NWOLTE	6	40	40	sec	No	No	FALSE	FALSE	Timers 1	NWOLTE	
12	RWOLTE	6	40	40	sec	No	No	FALSE	FALSE	Timers 1	RWOLTE	
13	FDELAYTE	3	12	12	sec	No	No	FALSE	FALSE	Timers 1	FDELAYTE	
14	TDELAYTE	3	12	12	sec	No	No	FALSE	FALSE	Timers 1	TDELAYTE	
15	FALRELSTE	60	120	120	sec	No	No	FALSE	FALSE	Timers 2	FALRELSTE	
16	FALRELTE	120	900	900	sec	No	No	FALSE	FALSE	Timers 2	FALRELTE	
17	TMALRELSTE	60	120	120	sec	No	No	FALSE	FALSE	Timers 2	TMALRELSTE	
18	TMALRELTE	120	900	900	sec	No	No	FALSE	FALSE	Timers 2	TMALRELTE	
19	TSALRELSTE	60	120	120	sec	No	No	FALSE	FALSE	Timers 2	TSALRELSTE	

Figure 3-97 Timers Assigned to New Submenus

This will result in the Web UI on the CPU III or OCE showing the new submenus under the Logic Configuration as shown in the red box below. Note that the default Timer and Properties submenus are not shown under Logic Configuration as in this case they have no properties or timers associated with them.

The figure also shows that the timers that have been marked **Read Only** in the WCCT (shown in the purple box) are greyed out in the Web UI to prevent them from being edited, and the timers that were marked as UCN in WCCT have the key symbol next to them (shown in green box).

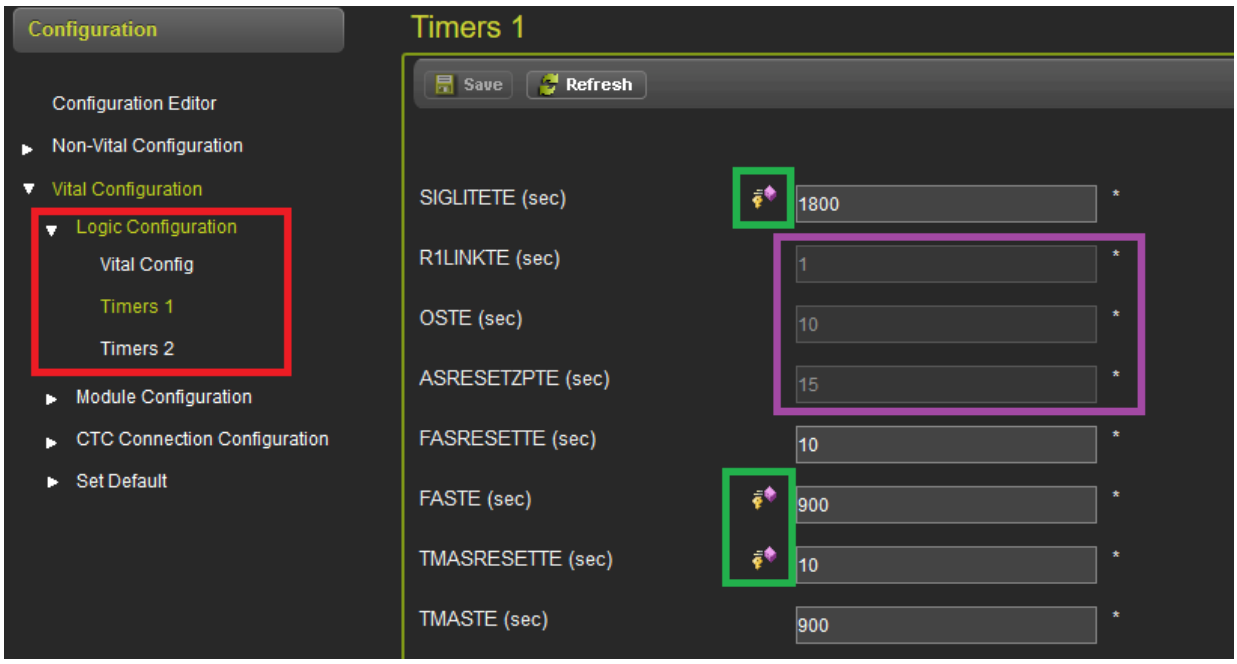


Figure 3-98 Web UI Display Example – Submenus

Because of limits on the number of parameters allowed on one screen, the submenus will need to be used if you have:

- A CPU II+ application with more than 16 items on a page
- A CPU III application with more than 25 items on a page

If the limit is exceeded, the WCCT will give an error such as the one shown in the following graphic.

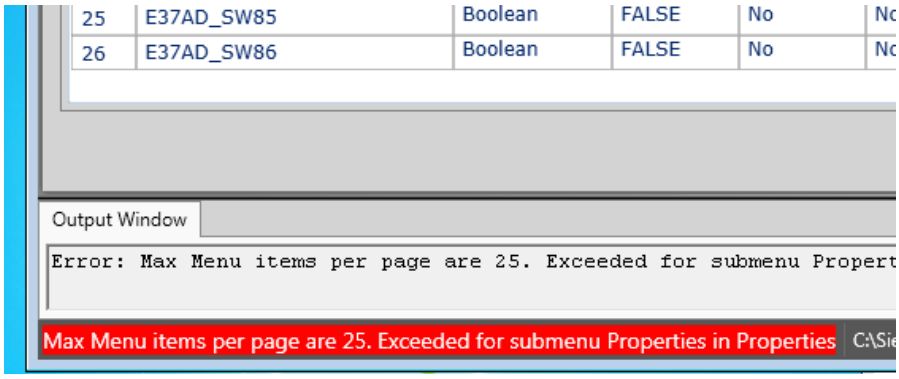


Figure 3-99 Items on a Page Error Message

3.19 RELAY LOGIC EDITOR

To open the Relay Logic Editor, select the Relay Logic icon from the tool bar, or **Edit>Relay Logic** from the menus, or use the **Ctrl G** shortcut.

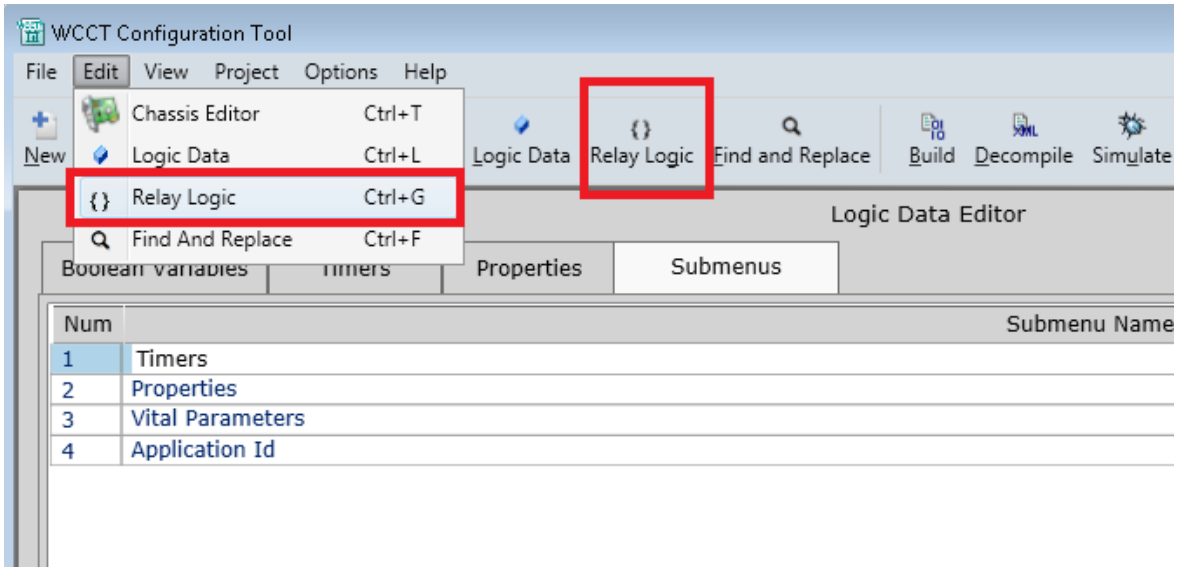


Figure 3-100 Selecting Relay Logic Editor

When there are no equations in a project, the Relay Logic Editor will appear as illustrated in the following figure.

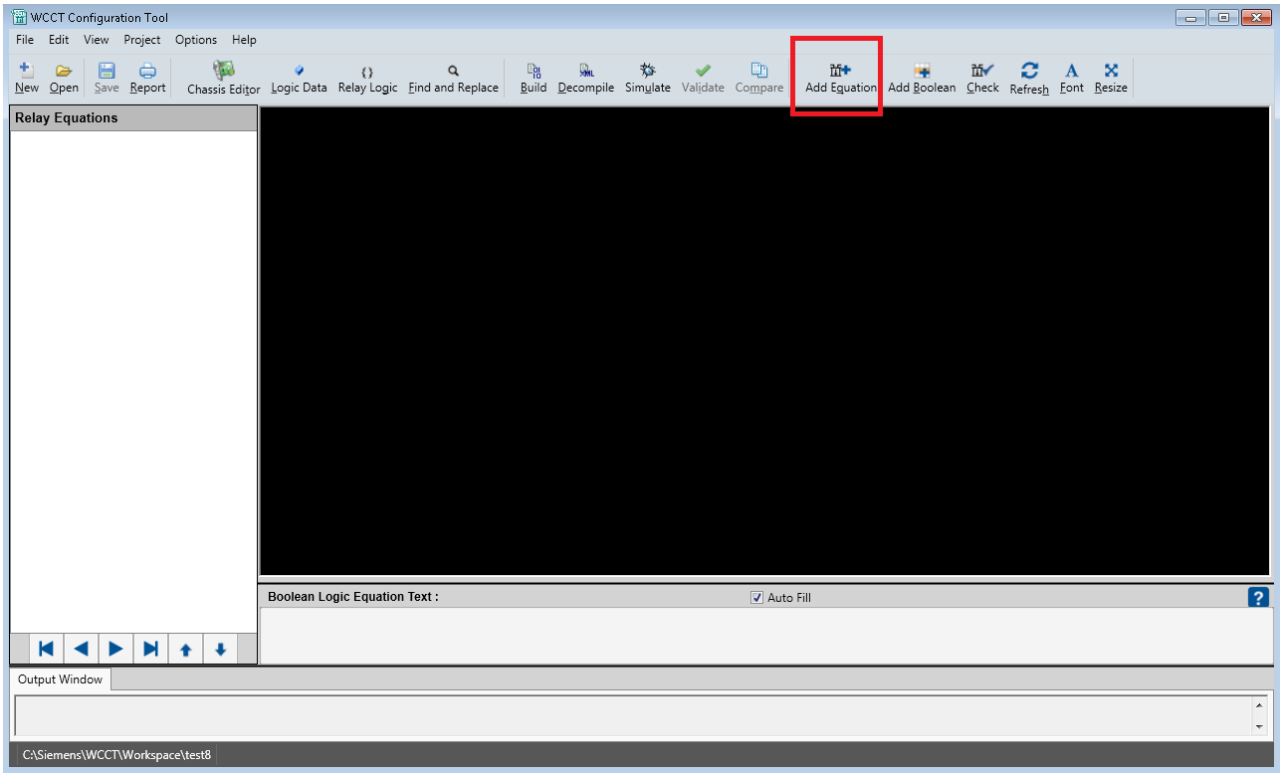


Figure 3-101 Initial Relay Logic Window

3.19.1 Adding and Editing Equations

To add an equation, select the **Add Equation** icon shown, then type your equation in Boolean format in the New Logic Equation text box and press **Refresh** or **Save** and the equation will appear in the left box and the relay logic equivalent in the top box.

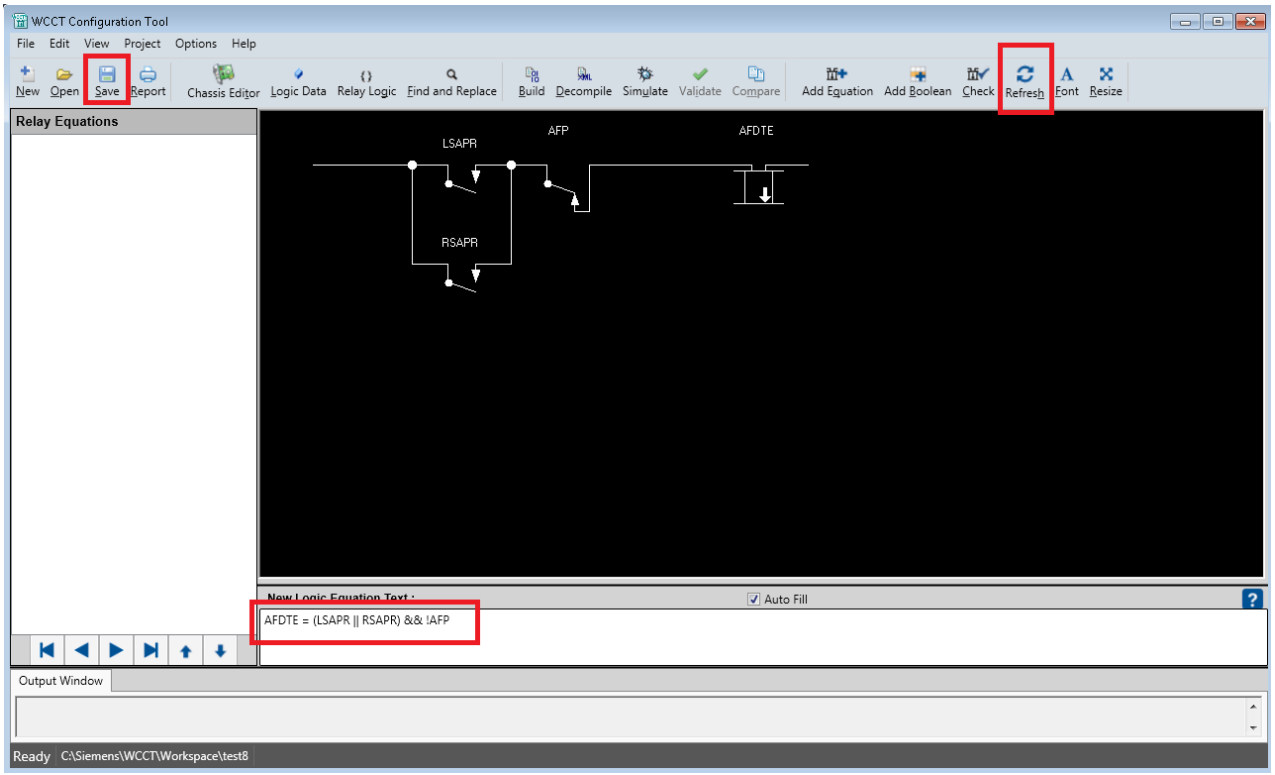


Figure 3-102 Equation and Relay Logic Equivalent Display

Use the following symbols in the equations:

&& – AND

|| – OR

! – NOT

= – Assign

( ) – Parenthesis

To start a timer, use the following:

StartTimerCond ({expression}, {timername})

Where {expression} is a variable or a Boolean expression, and {timername} is a timer that was created in the Logic Data Editor timer tab.

To check if a timer has expired, use **TimerExpired** ({timername}).

Note: StartTimerCond and TimerExpired are case sensitive.

Selecting the **?** will bring down a help box to remind you of the syntax.

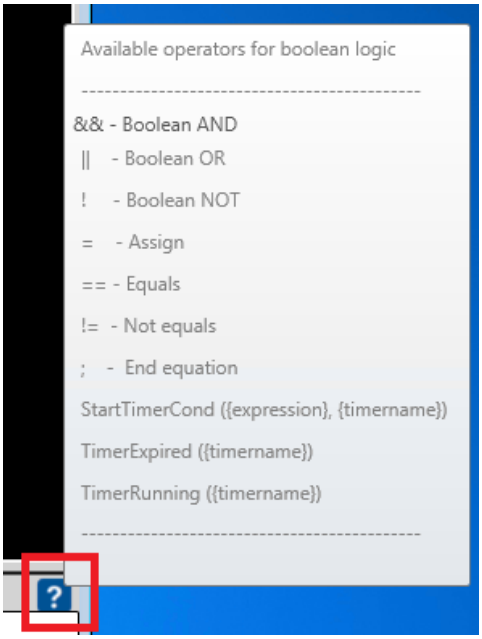


Figure 3-103 Syntax Help Box

Starting a timer based on a variable looks as follows in the relay view. The timer is marked with **TE** and the timer min, max, and duration are shown.

Figure 3-104 Starting a Timer Based on a Variable

Starting a timer based upon an expression looks as follows in the relay view:

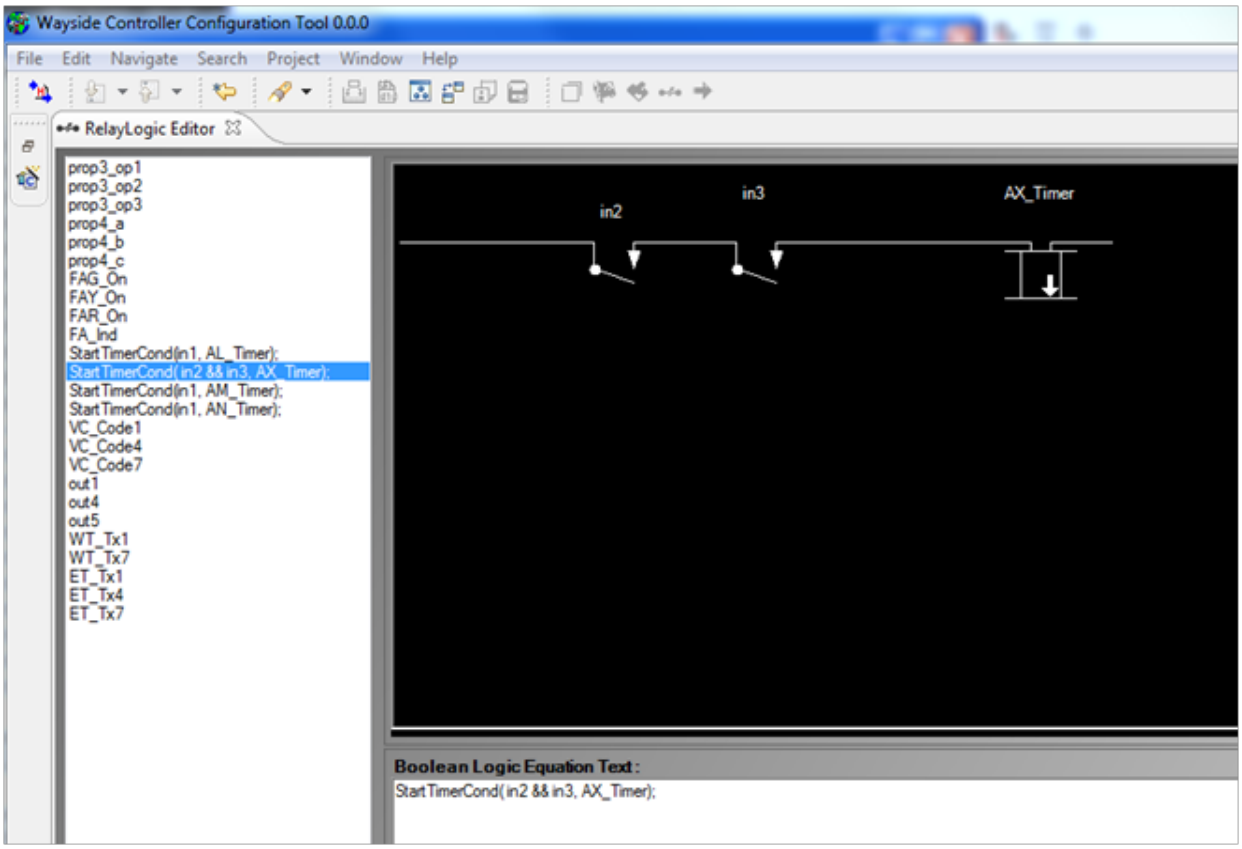


Figure 3-105 Starting a Timer Based on an Expression

Checking the state of a timer looks like the following in relay view:

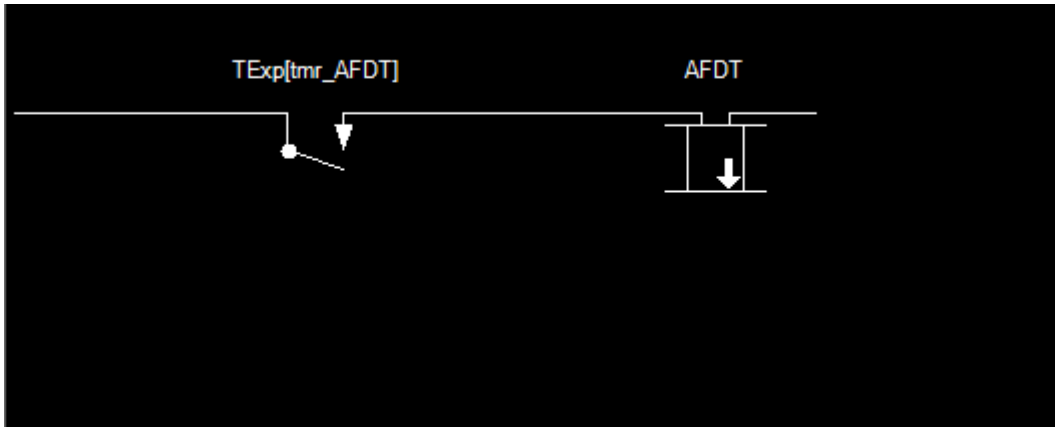


Figure 3-106 Checking a Timer State - Relay View

The state of the contacts and coils shown in the logic will depend on how the Normal values of these have been set in the Module I/O and Logic Data editors.

The Normal value can be changed by right-clicking on the contact or coil and selecting the **Toggle Normal Value**.

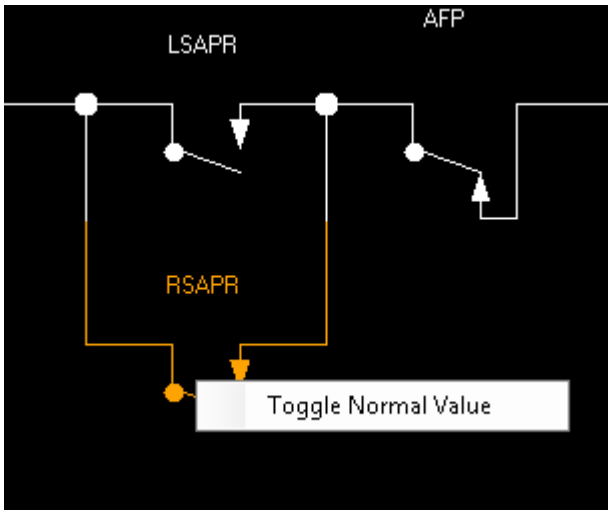


Figure 3-107 Changing Normal Value in Relay Editor

To add a new equation to the end of the list, select the **Add Equation** icon or use the shortcut **Ctrl+Alt Q**.

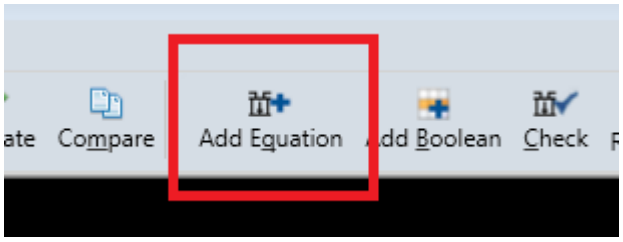


Figure 3-108 Add an Equation Icon

Type in the next equation in Boolean format; carriage returns and white space can be used in the text to make it more readable. Comments can be added in the text box preceding the equation, by using # to start the comment.

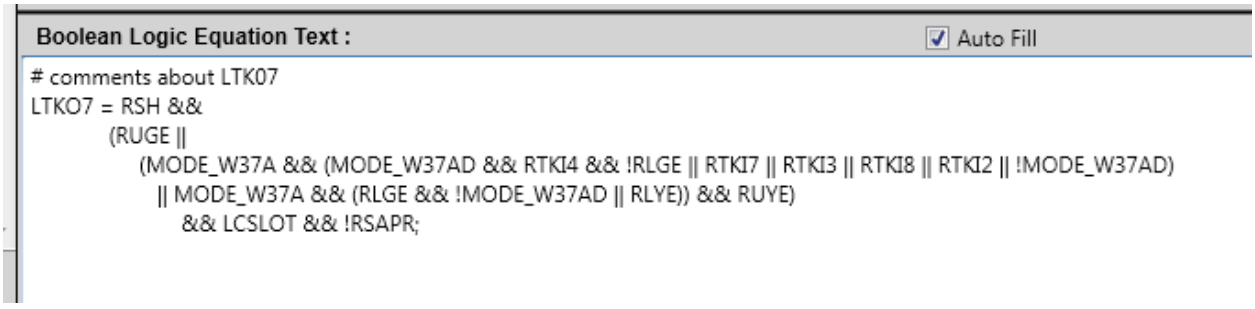


Figure 3-109 Line Breaks in Equation Text

Note that the window used to display the logic text can be expanded by putting the cursor on the black line just above the Boolean Logic Equation Text, until the cursor changes, then dragging the line upwards.

Continue to enter new equations. Note: equations can be cut and pasted from other sources using standard Ctrl C (copy) Ctrl V (paste). Multiple equations and their comments can be cut and pasted into the Boolean Logic Equation Text box.

As you are typing in a new equation, the editor provides an intelligent type-ahead function. As a new logic state name is typed in, a text box will show all existing logic states matching the letters typed. If the logic state has already been created, you can select it from the list and double-click to have it inserted automatically in the equation.

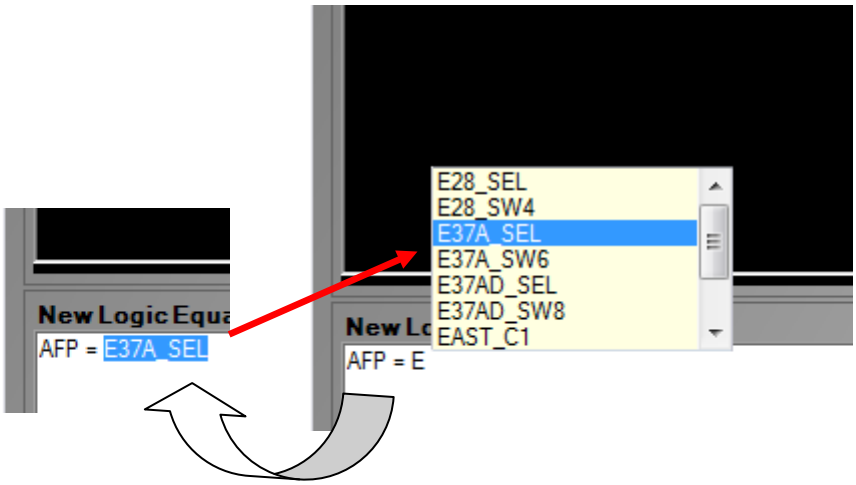


Figure 3-110 Type Ahead Function

If the type-ahead box pops up and you want to hide it, press the ESC button. Sometimes it is useful to turn the type-ahead feature off. To do so, uncheck the **Auto Fill** checkbox.

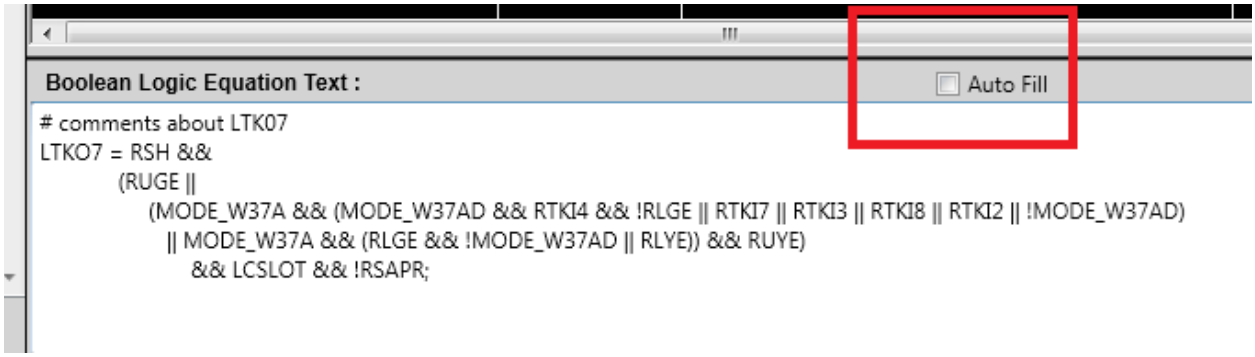


Figure 3-111 Turning Auto Fill Off

3.19.2 Navigating and Re-Ordering Equations

The Relay Editor will show a list of all the equations on the left side of the window.

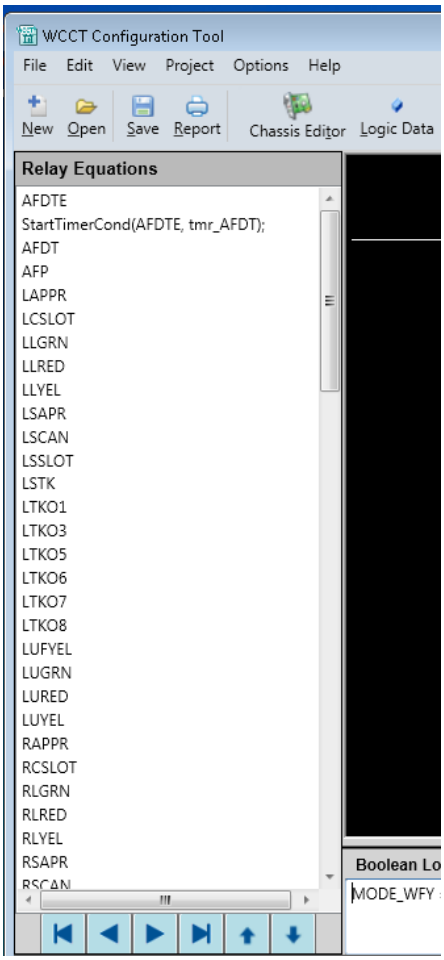


Figure 3-112 List of Relay Equations

To view an equation, double-click it from the list on the left, or right-click and select **View**.

Figure 3-113 Selecting View

To rearrange the order of the logic, right-click on an equation in the left window and from there either cut/paste, move the equation up or down, or insert a new equation before or after this one. The left two buttons at the bottom also provide a quick way to move an equation up or down.

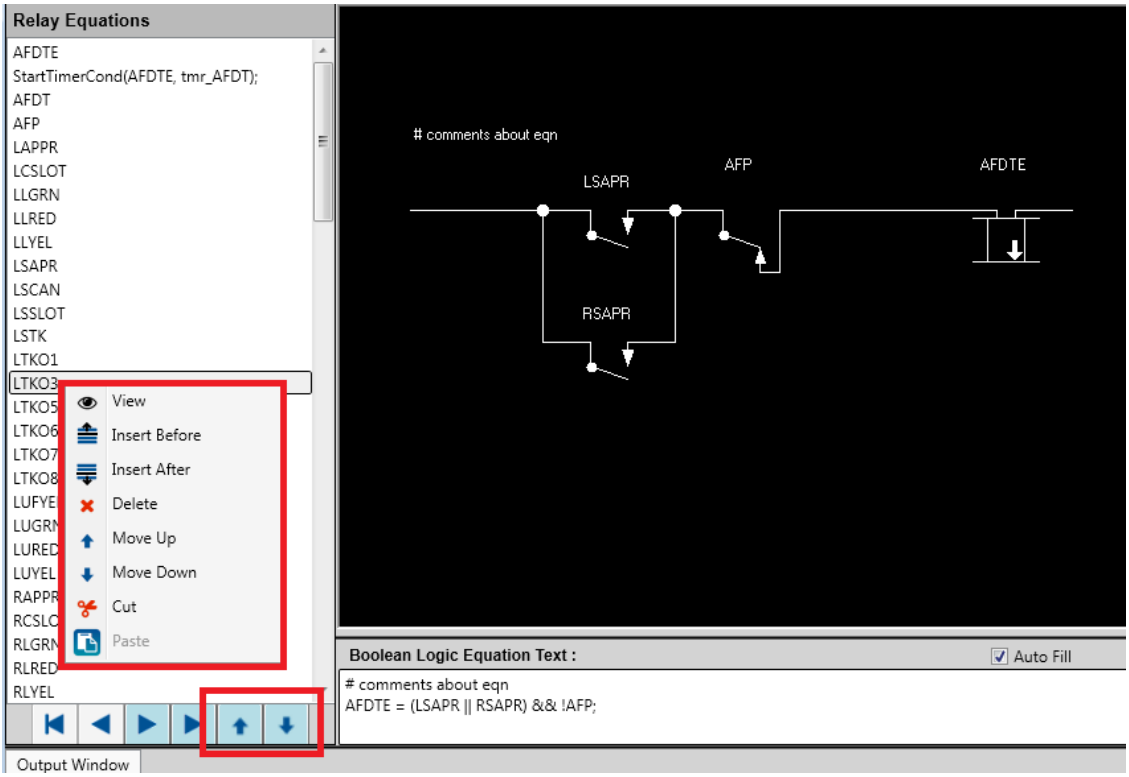


Figure 3-114 Rearrange Logic Order

When inserting a new equation, the header of the text box will indicate where it's being inserted.

Figure 3-115 New Equation Header

If the relay equation becomes too large, scroll bars will appear so all of the equation can be viewed.

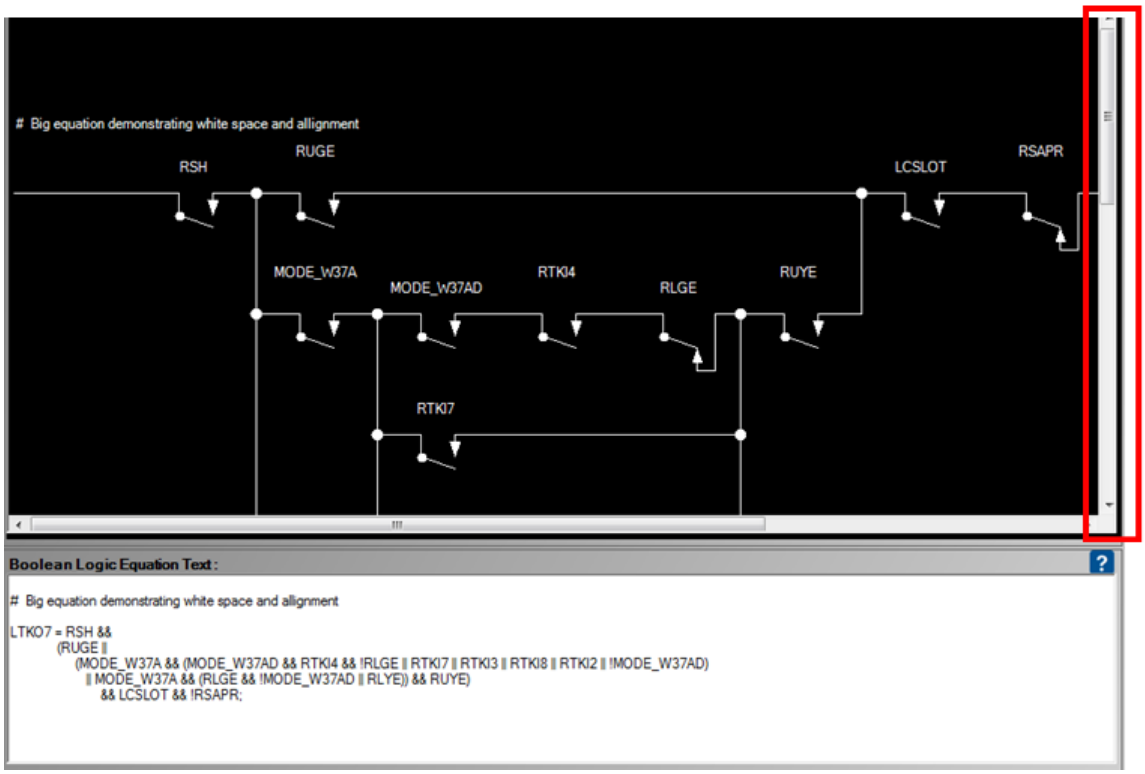


Figure 3-116 Display Scroll Bars

Once the project has many equations it is useful to use the **Find and Replace** function to locate where logic states are defined or used. Enter the Logic State name to be found and press **Find**. The relevant equation can then be found in the location column. To go to this equation, double-click it, or select it and click **Go to Equation**.

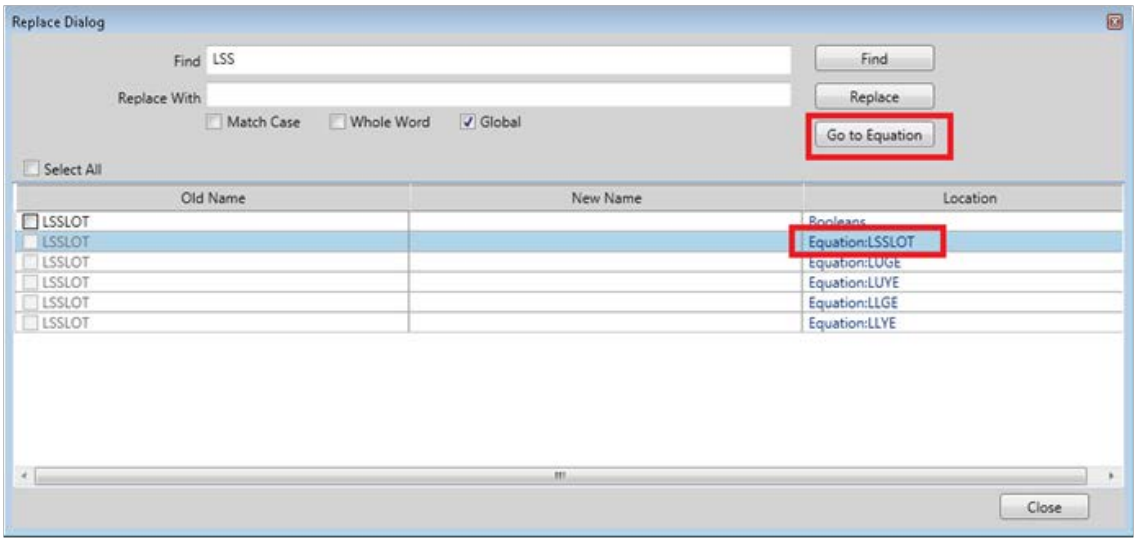


Figure 3-117 Equation Find Function

3.19.3 Check Equations and Adding Missing States

Once an equation has been typed in, it is often useful to display it to make sure its structure is correct before running any checks on it or permanently saving it. To do this, select the Refresh button. If all the logic states exist, the relay diagram will be shown. If some of the logic states do not exist, for instance they may be typed incorrectly or have not been entered yet, the WCCT will show the pop up window below and list the states that do not exist.

Press refresh at this point to show the relay equation.

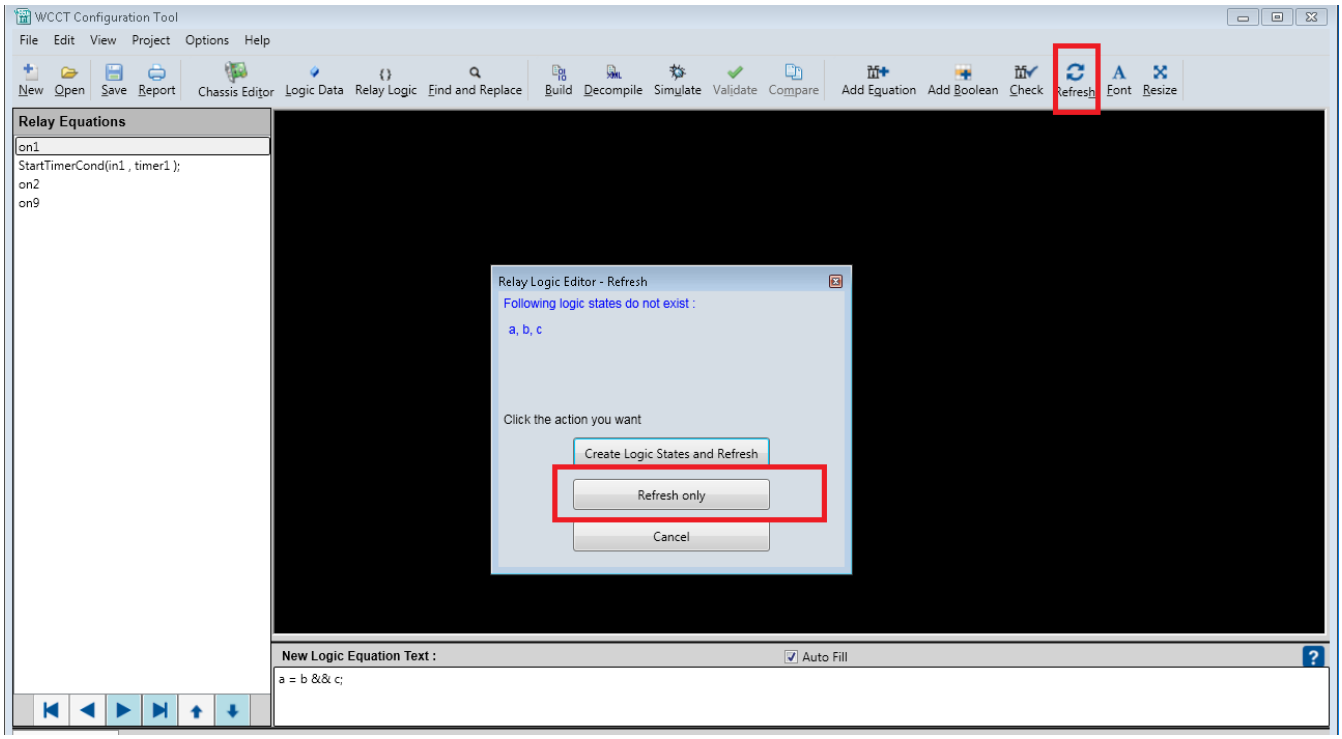


Figure 3-118 Refresh Logic Window

If terms have been typed incorrectly, press **Cancel** and go back and correct them. If there are logic states that need creating, press **Create Logic States and Refresh**.

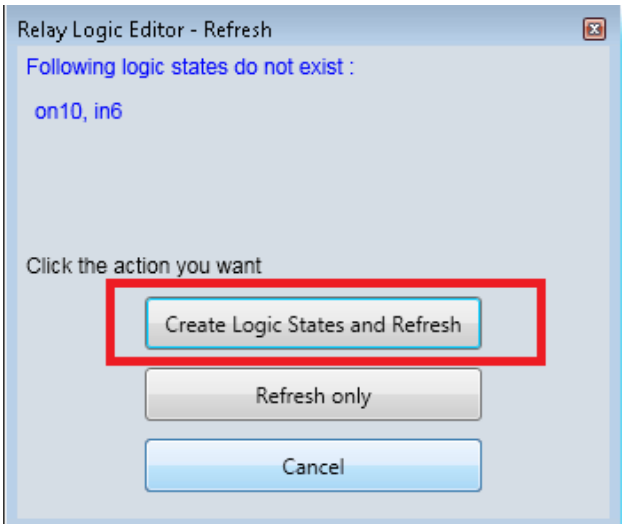


Figure 3-119 Refresh Logic Options

This will create the new logic states as internal variables with a default value of **False** and Log as **No**.

Note: at this stage there is no easy way in the WCCT to change these variables to module inputs or outputs; either enter all inputs and outputs prior to writing the relay logic, or create a new module input or output with a different name and write an equation to assign the new internal state to it. If the **Create Logic States and Refresh** has been selected and the equation looks correct, you may proceed and check the equation using the **Check** icon from the top toolbar:

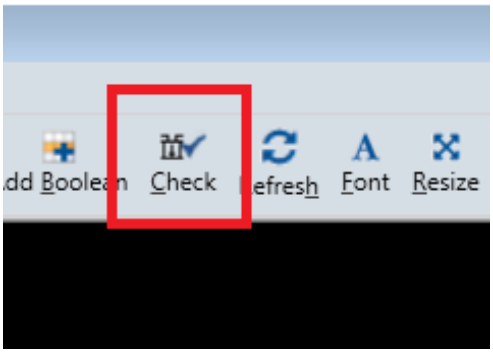


Figure 3-120 Check Equation Icon

This will check the syntax of the equation and whether the states in it exist. If the logic states don't exist, a message will indicate which ones don't exist and ask if you want to create, as shown in the following figure.

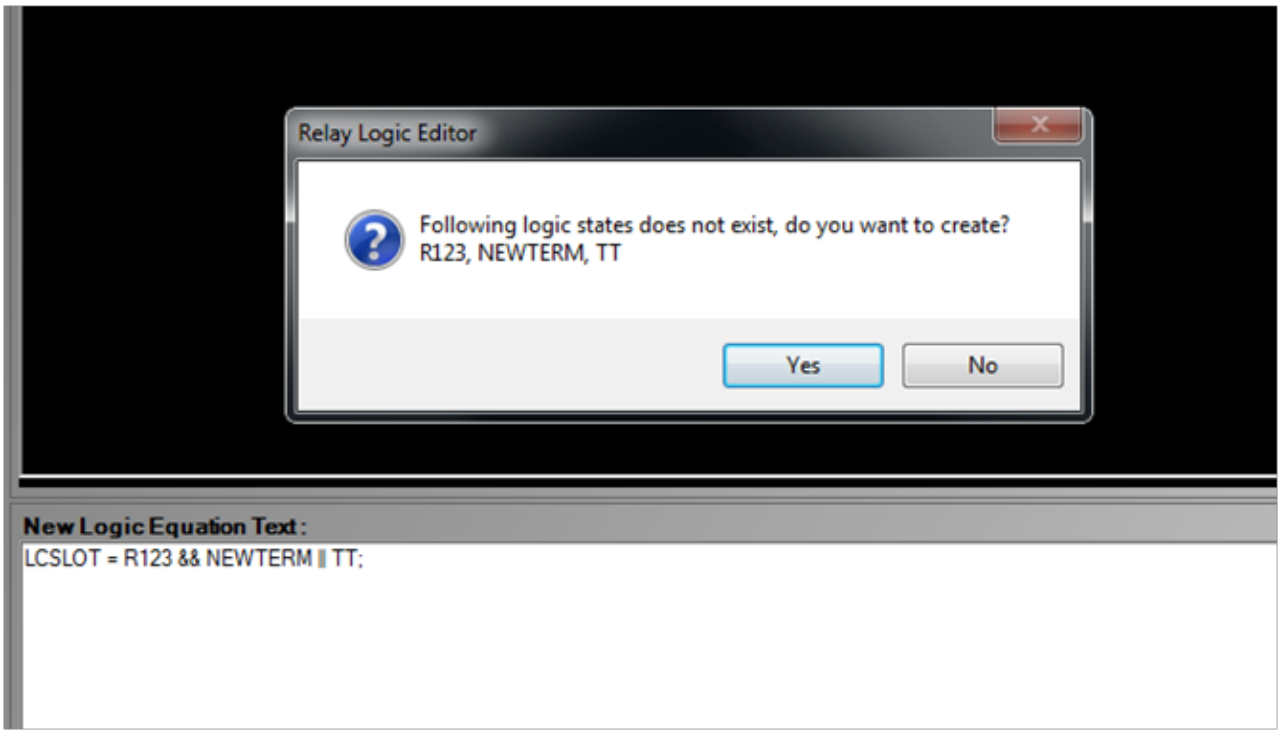


Figure 3-121 Check Equation Result Example

If you answer yes, the tool will create these Boolean variables with the default value **False**, and **Log** = **No**. If the properties of the new states need changing, go to the Logic Data Editor and change them there. If you do not desire to change the properties, you can go back and correct the logic state if you have made a typo, or manually add in the missing variable by selecting the **Add Boolean** icon.

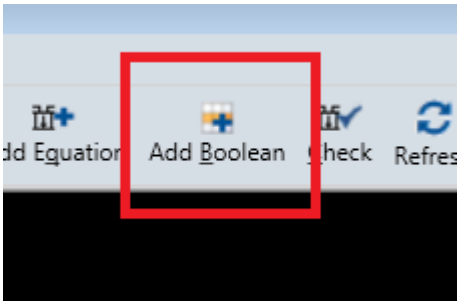


Figure 3-122 Add Boolean Icon

Add Boolean brings up a new window to add a new Boolean variable and set the properties.

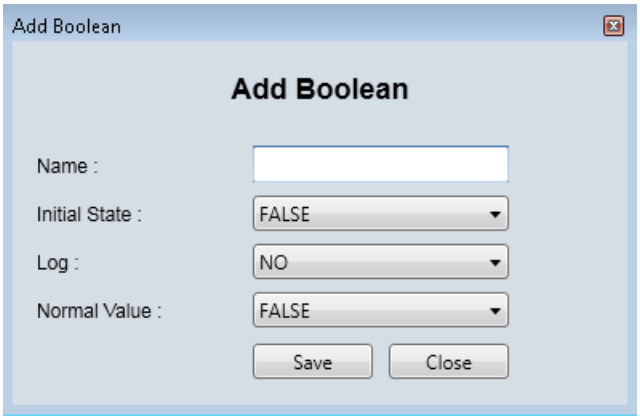


Figure 3-123 Add Boolean Variable Properties

If an equation is typed in and **Save** selected, the tool will check whether the logic states exist. If some do not exist, the WCCT will show the following window indicating which logic states are missing.

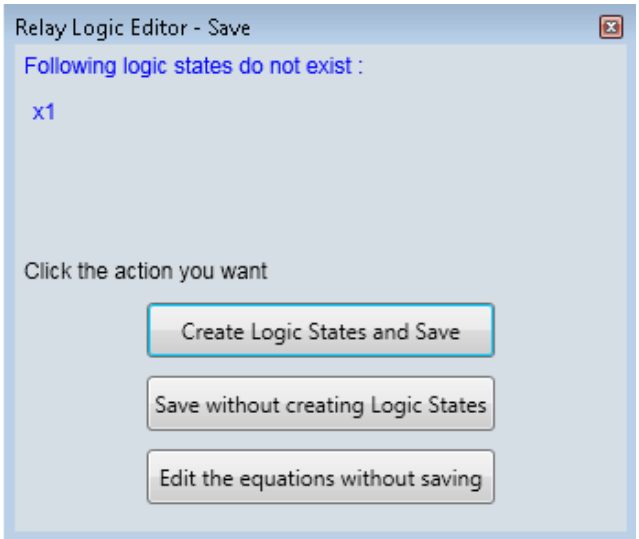


Figure 3-124 Logic State Check Result Window

Select **Save without creating Logic States** to continue without creating the missing states. If you do not want to proceed, select **Edit the Equations without Saving** and go back and correct the equation, or select **Create Logic States and Save** and have the WCCT automatically add the missing terms as Boolean Variables. You may want to continue with the save, if, for example, you have used a logic state that is an input or an output and will need to be added through one of the other editors.

Generally, it is expected that before the logic is written, all the input and outputs, timer and properties and most of the Boolean variables required have been created. The relay logic editor allows the user to add in new Boolean variables, but if new inputs or outputs are required, it is necessary to open the chassis editor and go to the appropriate module to add them. If more timers and properties are required, go to the Logic Data Editor to add them.

Under the Options/Settings menu there is a selection to auto save the logic.

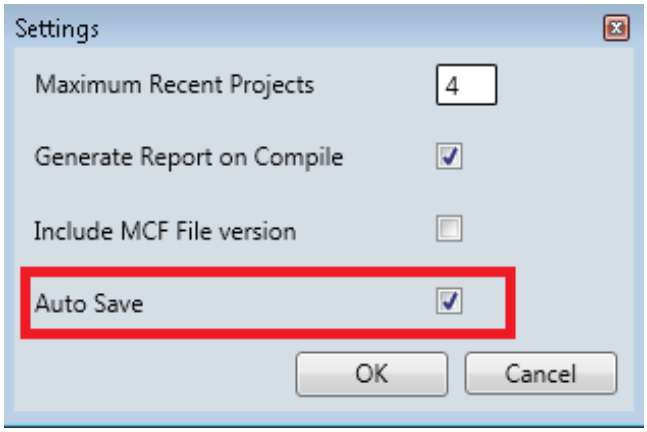


Figure 3-125 Auto Save Logic

When this is selected, there are pending changes to the current equation, and the user chooses to move to another equation or out of the relay editor, the tool will automatically save the pending equation. As usual, if there are errors in the equation, the tool will show the error window (as per Figure 3-121).

When Auto Save is not selected, there are pending equation changes, and the user chooses to move to another equation or out of the relay editor, the tool will ask the user whether they want to save their changes.

3.19.3.1 Resizing Equations

The relay contacts shown on the diagram can be resized by selecting the **Resize** icon.

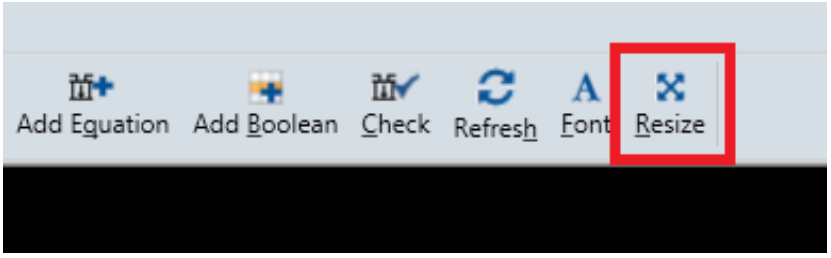


Figure 3-126 Selecting Resize

This brings up a window that lets the user change the size of the relays. This becomes more useful when printing the PDF report as it allows the user to adjust equations to fit the page. See the section on Relay Reports 6.6.

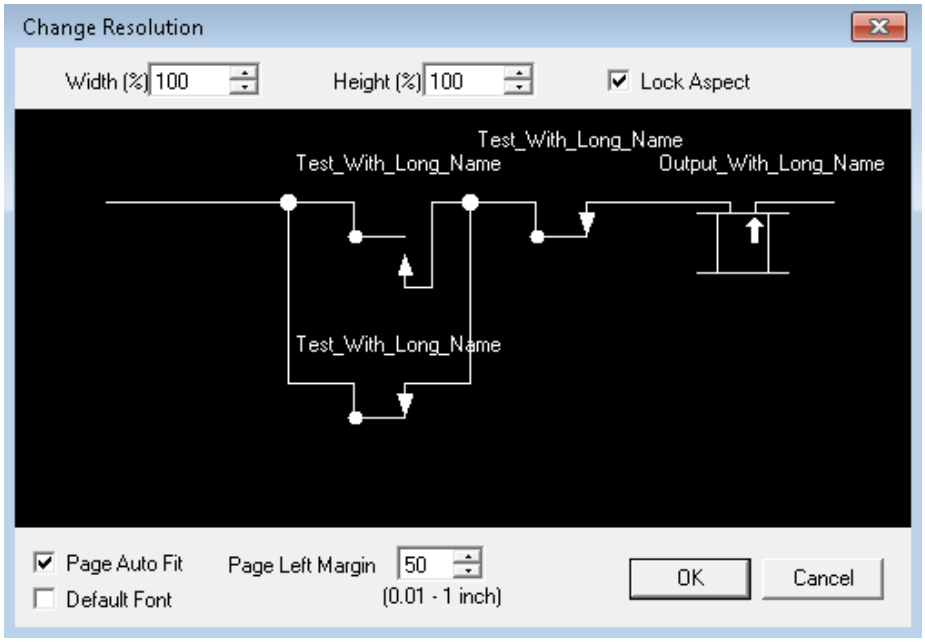


Figure 3-127 Resize Windows

3.20 FIND AND REPLACE EDITOR

To bring up the Find and Replace editor, select **Edit>Find and Replace** from the menu, select the **Find and Replace** icon, or use the **Ctrl F** shortcut.

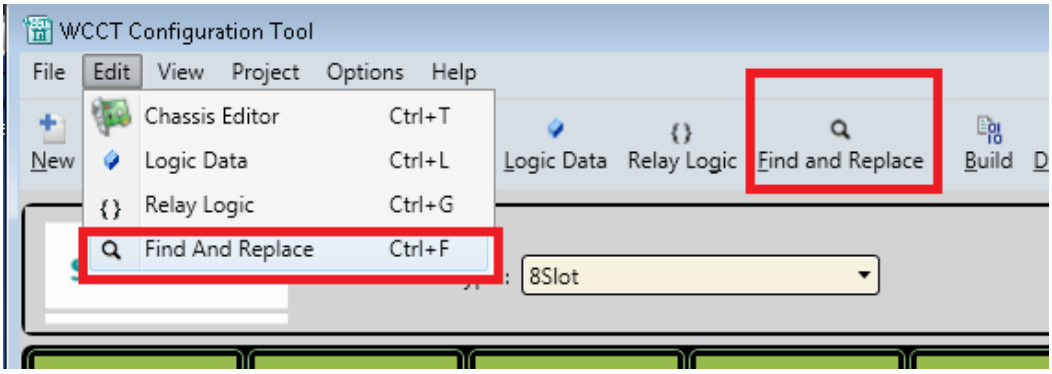


Figure 3-128 Selecting Find and Replace

The following window will appear.

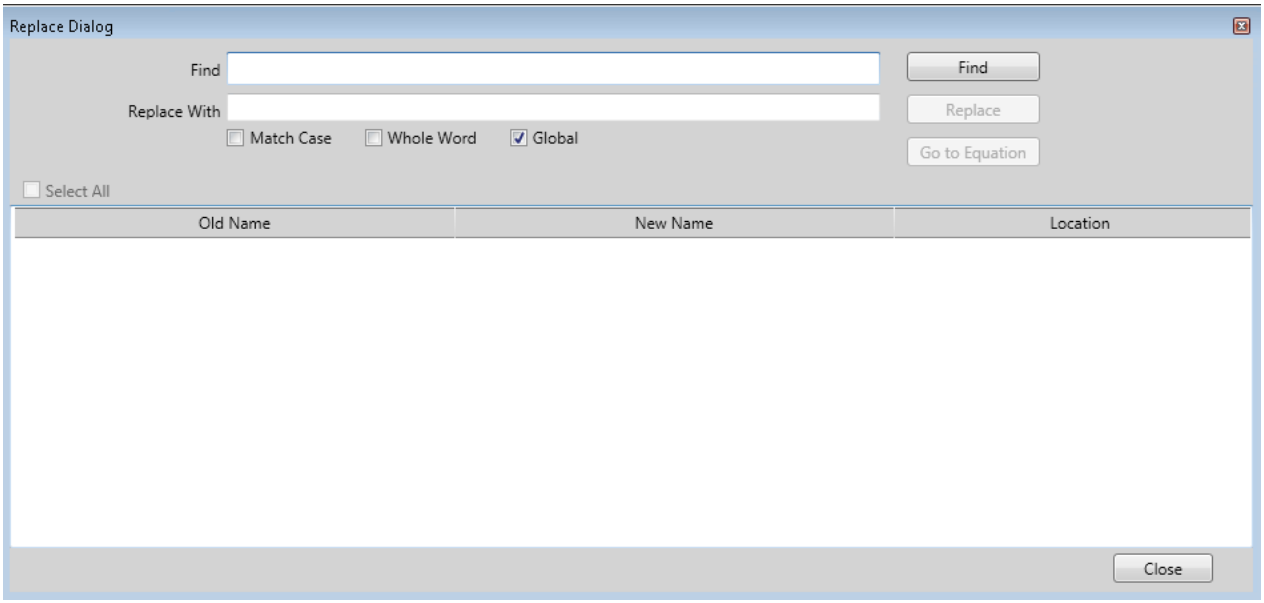


Figure 3-129 Find and Replace Window

The window can be used to find where logic states are used, or can be used to rename them.

First type the string to be found. This can be a complete logic state name or a partial one.

To match the case in the find string, check the **Match Case** check box.

Note: logic state names can be entered as upper case, lower case, or mixed case, but the WCCT is not case sensitive in terms of how these are used.

For example, the logic states: LSSLOT, lsslot and LsSlot all refer to the same logic state.

To match the whole word, check the **Whole Word** check box.

The search results can be restricted to just the Editor that is currently open: for example, if **Global** is unchecked and the Chassis editor is currently open, the results only show matches that are used as module input and outputs; if the Logic Data editor is currently open, the results only show matches that are used the logic data editor; if the Relay Logic editor is currently open, the results only show matches that are used in the relay logic.

If **Global** is checked, the search results show matches from the whole project. Logic state names matching the entered text are shown in the **Old Name** column. The Location column shows where the logic state is found. It will show:

- Booleans
- Timer
- Properties
- Equation: {eqn name}
- Slot-{slot number}

Note: **Find** also finds the names of submenus, which may occasionally be useful. These are marked in the location column as **Submenus**.

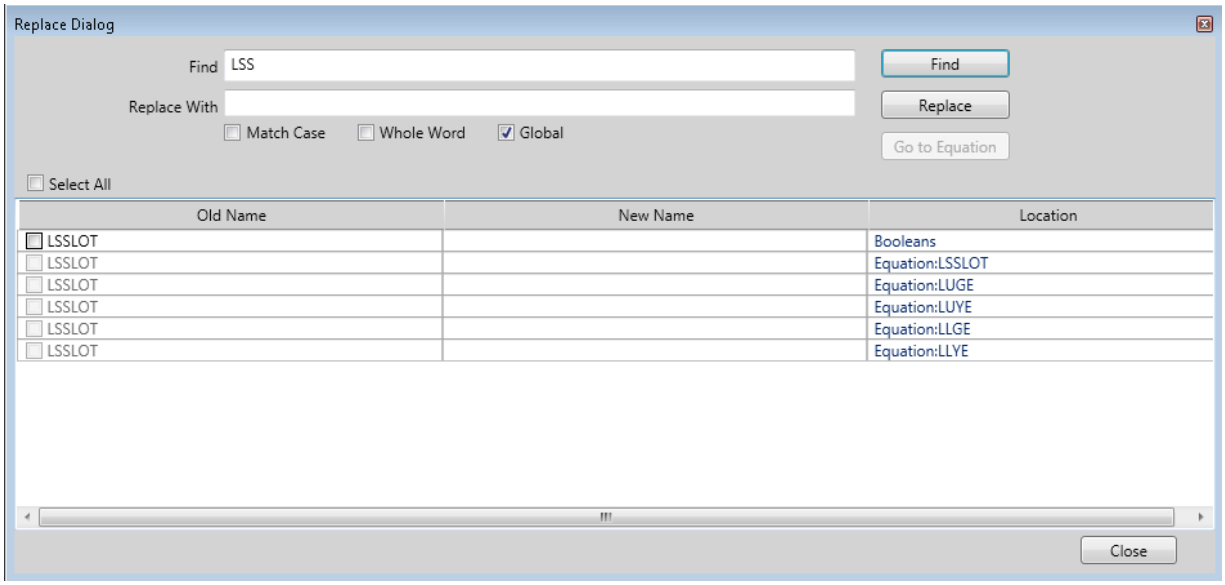


Figure 3-130 Enter Find String

To change the name(s) of a logic state, type the new text in the **Replace With** text box and press **Find**.

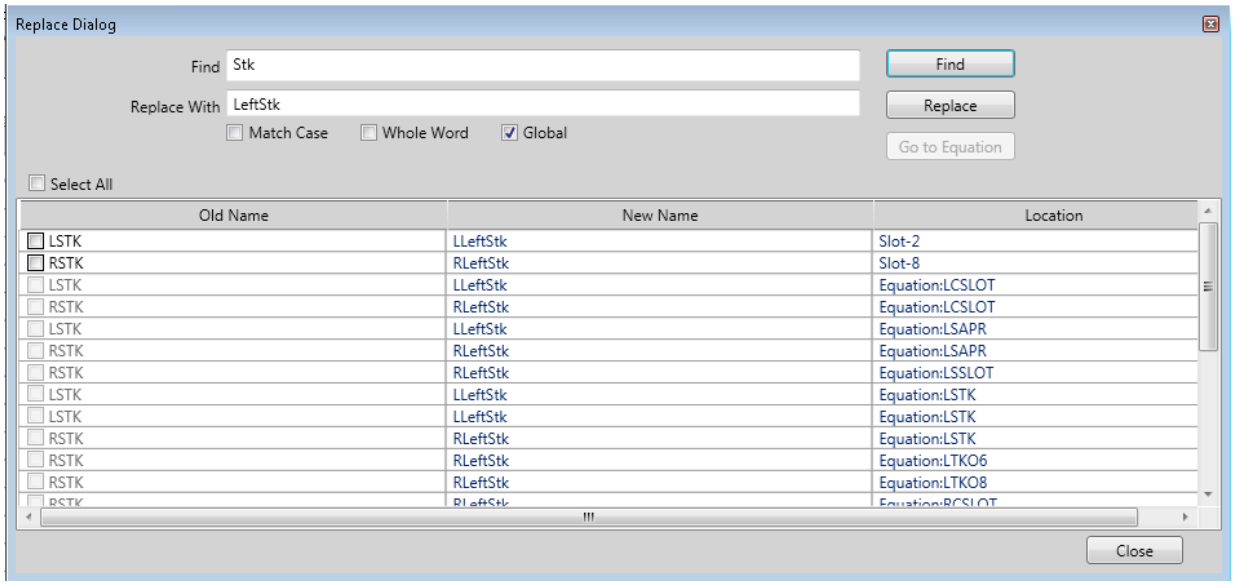


Figure 3-131 Enter Replace String

Then select the rows to be replaced by checking the check box in the **Old Name** column. Note that if identical names are found, only the first instance needs to be selected.

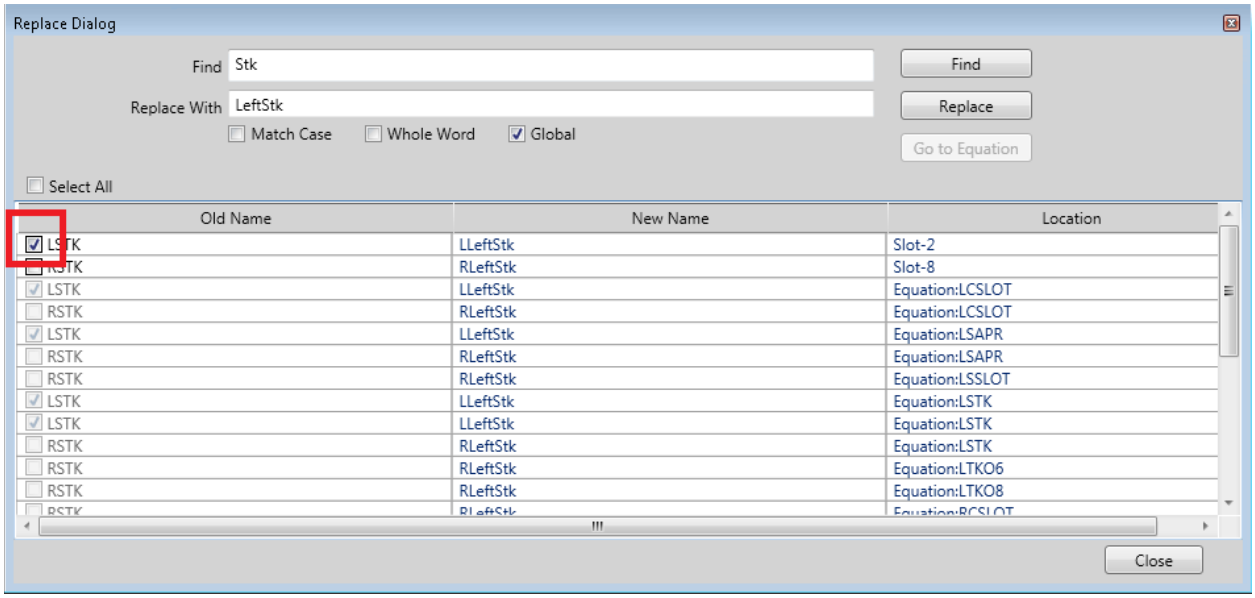


Figure 3-132 Selecting Which Items to Replace

Once the rows are selected, press **Replace**. The WCCT will replace the selected rows with the new name.

SECTION 4 CREATING AN APPLICATION

4.0 GETTING STARTED

This section provides an introduction on how to get started with the WCCT to create a basic application. When the WCCT starts it will open with a blank workspace.

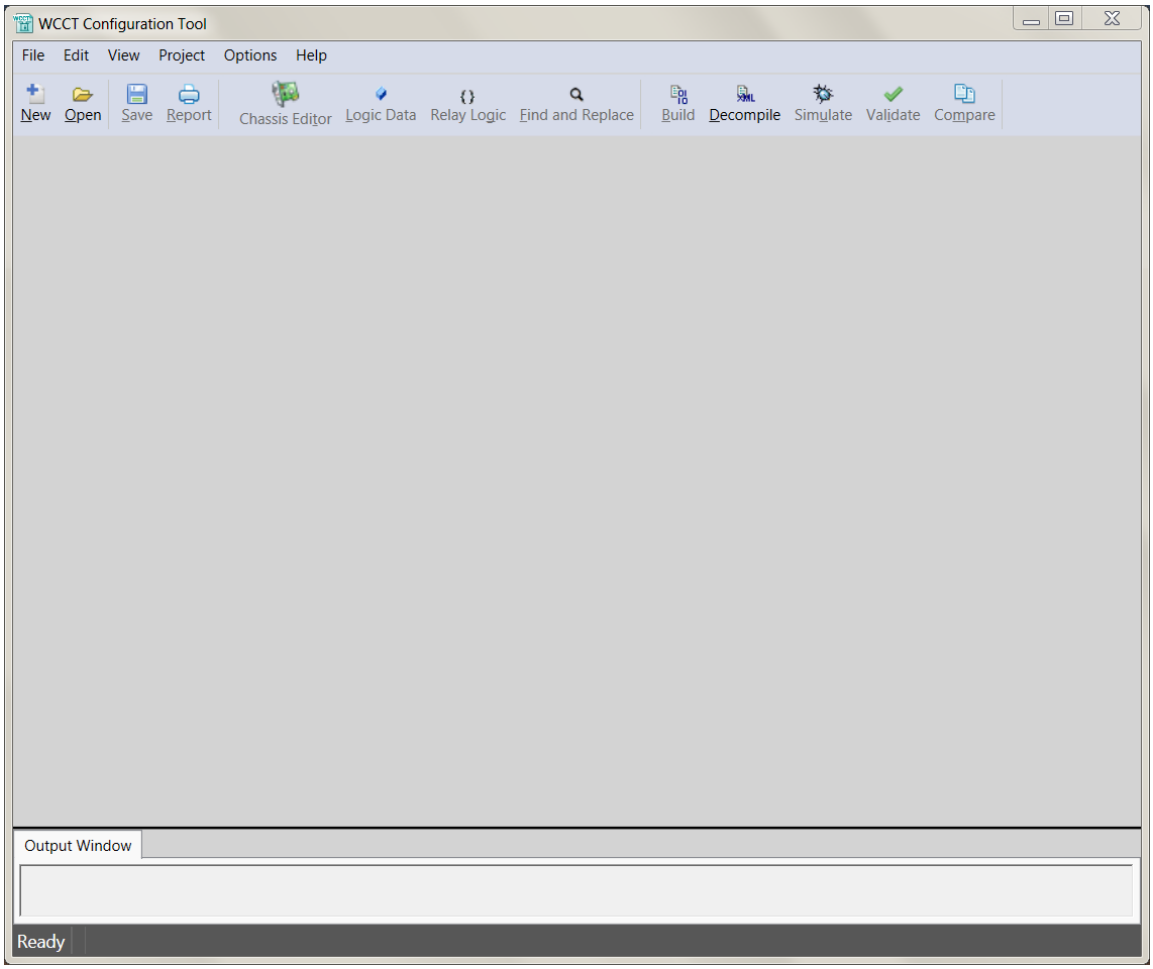


Figure 4-1 WCCT with No Project Open

From the main WCCT window, select **File>New**, the **New** icon, or use the **Ctrl + N** shortcut.

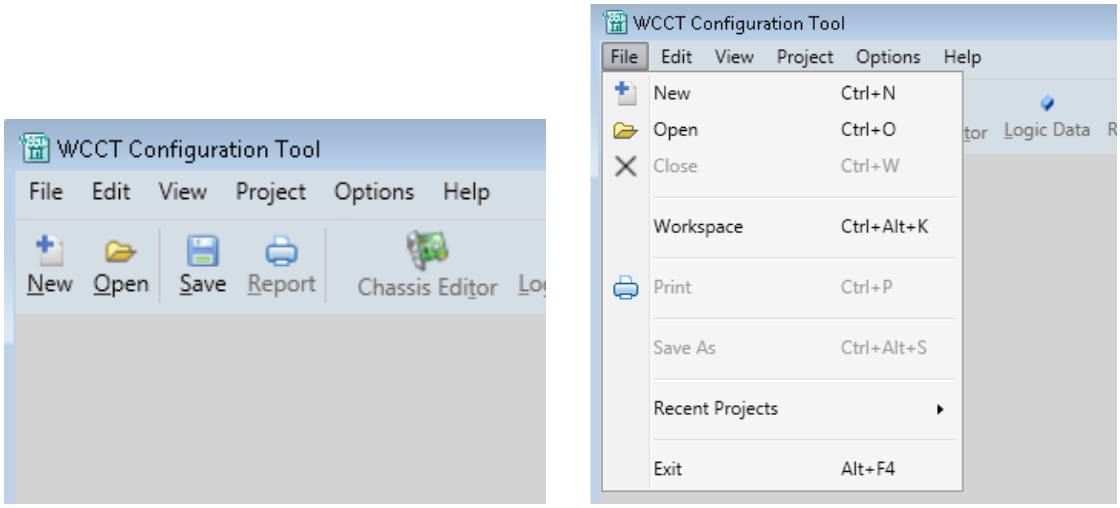


Figure 4-2 Creating a New Project

A new project window will open, type in the project name (note: spaces or special characters are not allowed). To change the location where you want to keep your projects, select the **Browse** button next the **Location** field. The default **Location** is shown below (note: the default location for all projects can be changed – see setting Workspace in section 2.2.1).

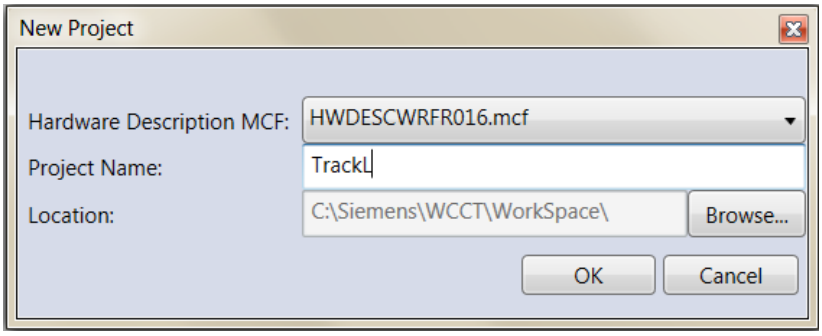


Figure 4-3 New Project Window

The Hardware Description MCF is used to define module properties used by the tool. Generally, it is recommended to use the latest version available.

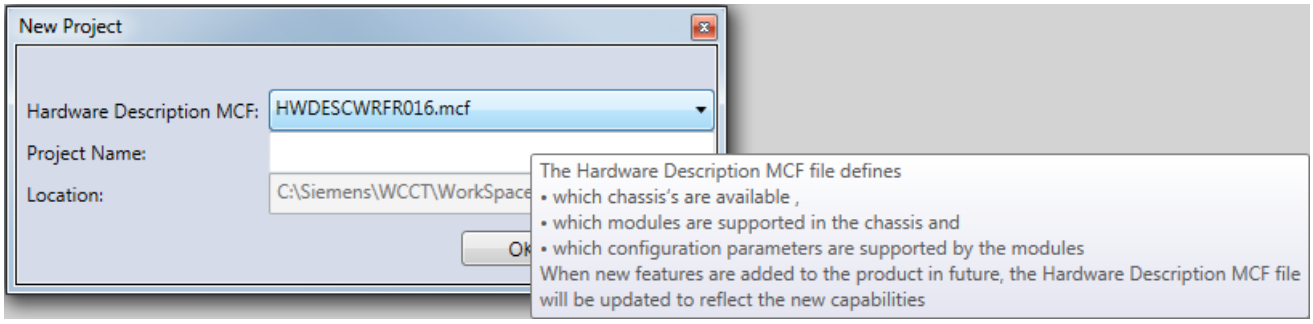


Figure 4-4 Hardware Description MCF

The WCCT will then open up the Chassis editor.

4.1 SET PROJECT PROPERTIES

First, set the project properties by selecting the **Project Properties** option from the Project menu.

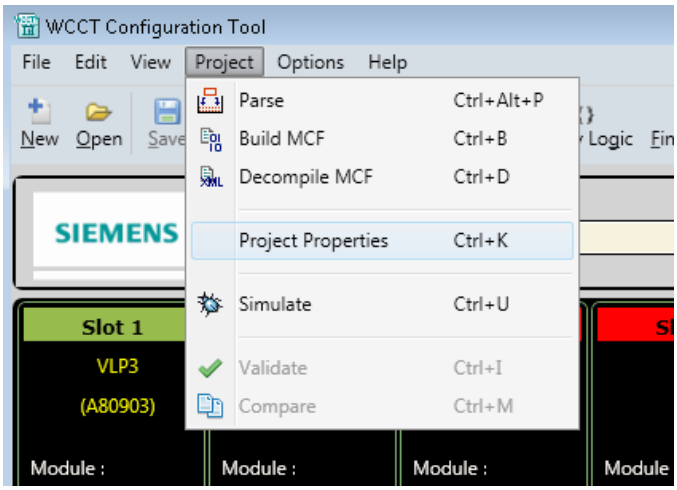


Figure 4-5 Project Properties

This will open the Project Properties window as shown below. Here, the user can enter information into the various fields describing the application. The MCF Version number is usually part of the MCF file name; however, it can be excluded by using the Settings menu option (see **Settings menu > Section**).

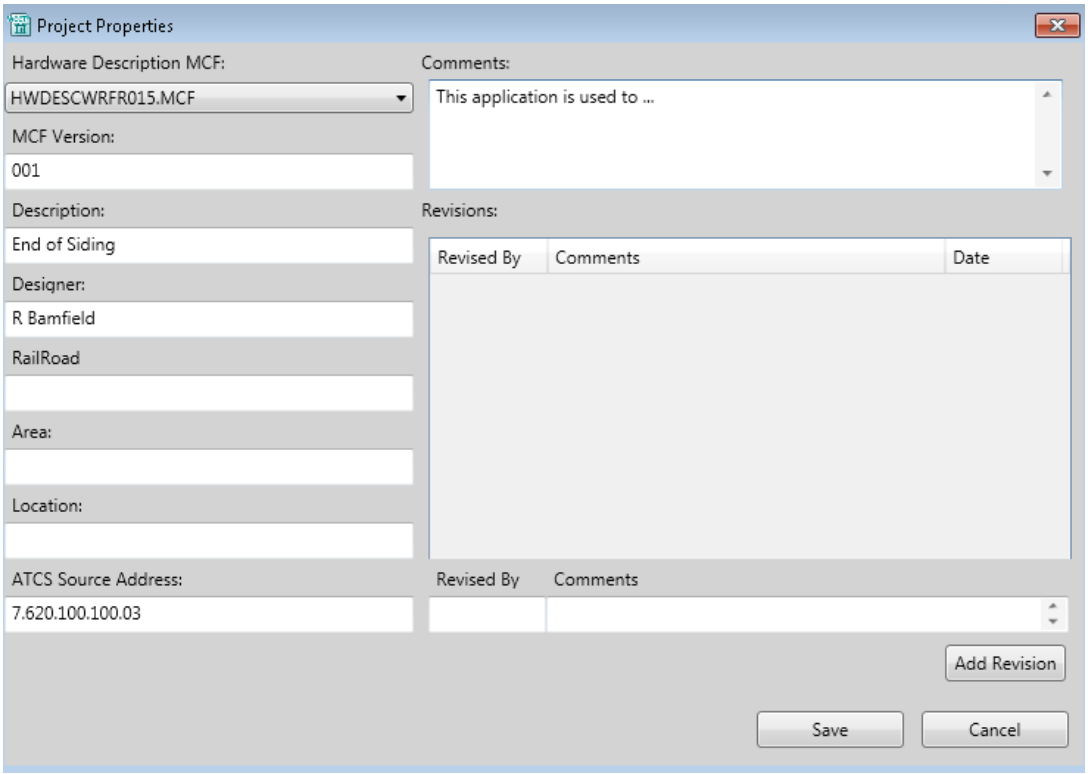


Figure 4-6 Project Properties Window

The user can add revision comments for the project by entering in the **Revised By** and **Comments** fields then pressing the **Add Revision** button.

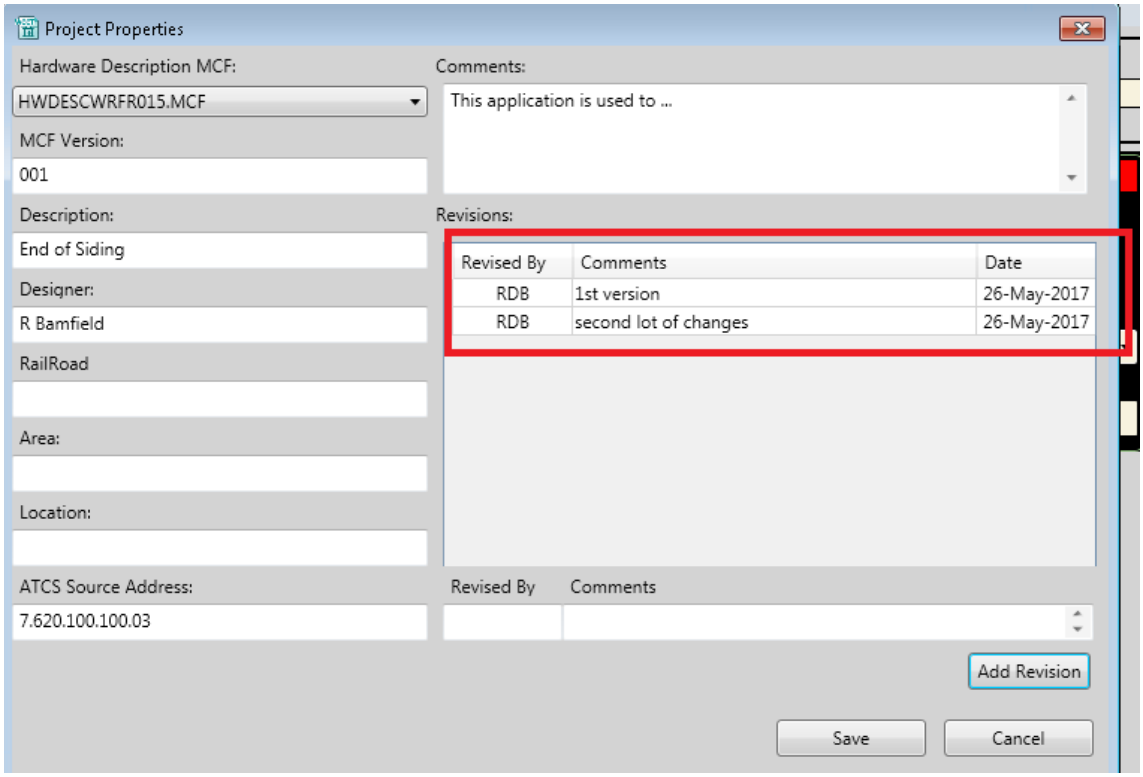


Figure 4-7 Setting Revision Comments

The user can also set the ATCS address of the WayConneX/GEO in this window. The ATCS address has to be in the form 7.RRR.LLL.GGG.SS where:

- RRR in range 1-999
- LLL in range 1-999
- GGG in range 1-998
- SS in range 03-98

Once the project properties have been set, click **Save** to close the window.

4.2 SELECT CHASSIS

The WCCT will show the Chassis Editor; this is where the user can select the modules required for the application. The default chassis type will be the 8Slot WayConneX chassis, which allows any I/O or track module in any of slots 2 through 8, the TwoTrackNE WayConneX is also available, where slot 2 and 8 are used only for Coded track or Line modules. If using older GEO equipment, change the type to TwoTrackExtended, ThreeTrackShort, or the ThreeTrackExtended option; whichever option best suits the equipment installed.

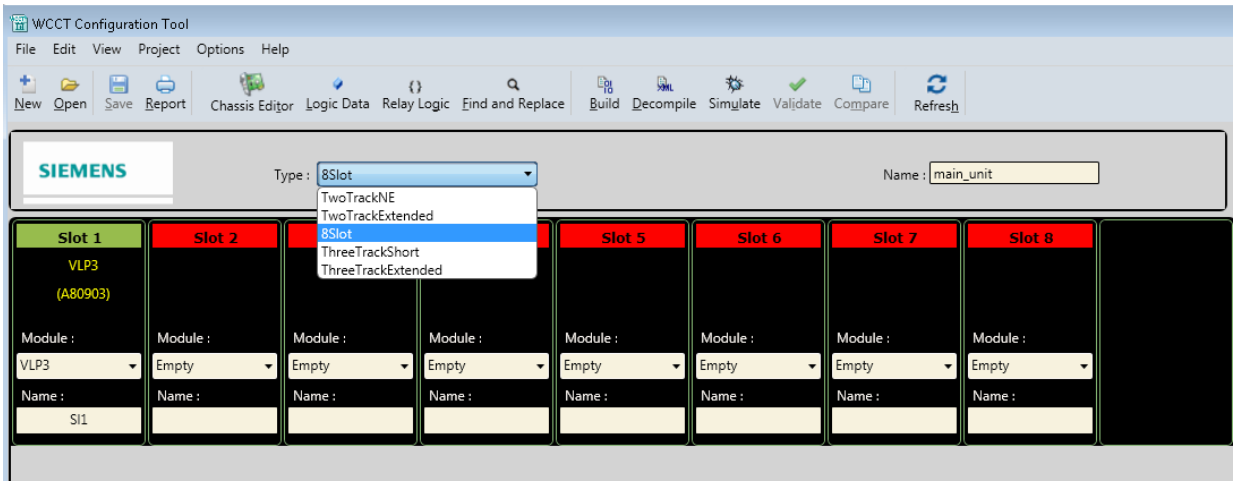


Figure 4-8 Chassis Editor

4.3 SELECTING MODULES

Click on the **Module** drop down box on the required slot to select the desired module.

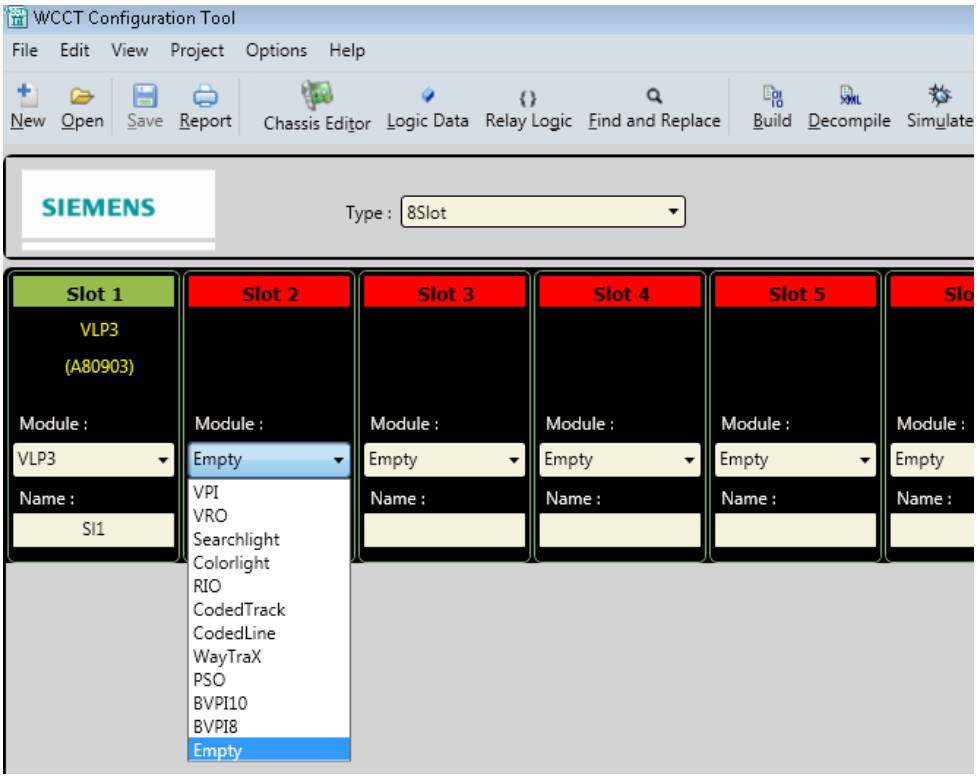


Figure 4-9 WCCT Chassis Editor Selecting Modules

The WCCT will now show the selected modules in green.

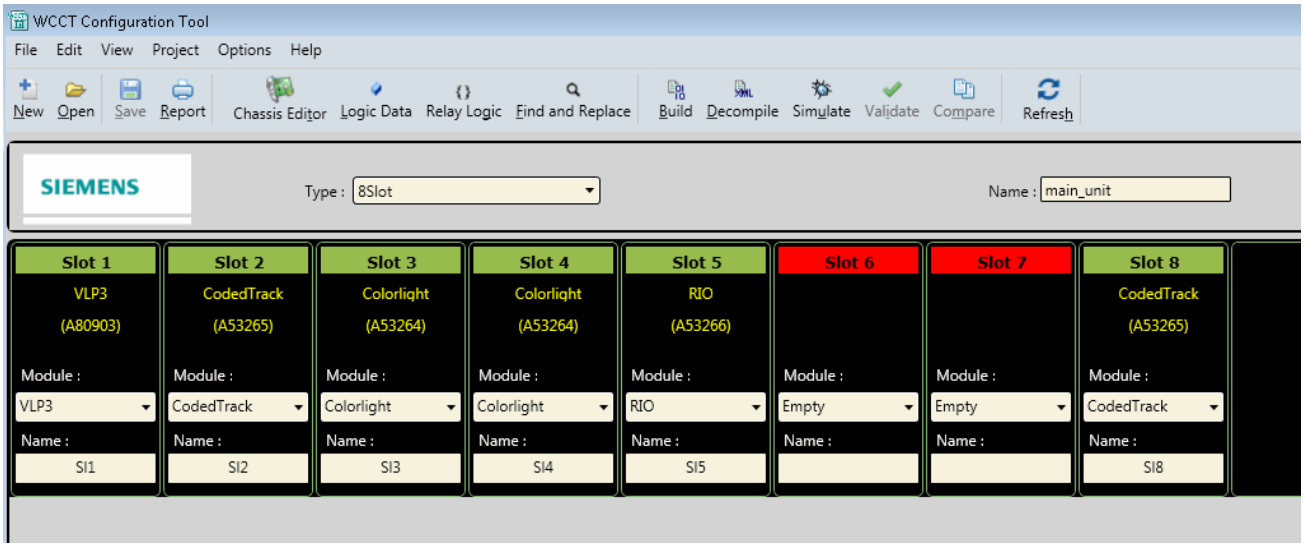


Figure 4-10 Saved Modules Screen

The default module names will appear as SL1, SL2, etc. (meaning slot 1, slot 2). To change these, type in a new name for the module, e.g. CPU, WestTrk, FA, IO, etc. as shown below:

Figure 4-11Setting Module Names

New modules can still be added by selecting empty slots.

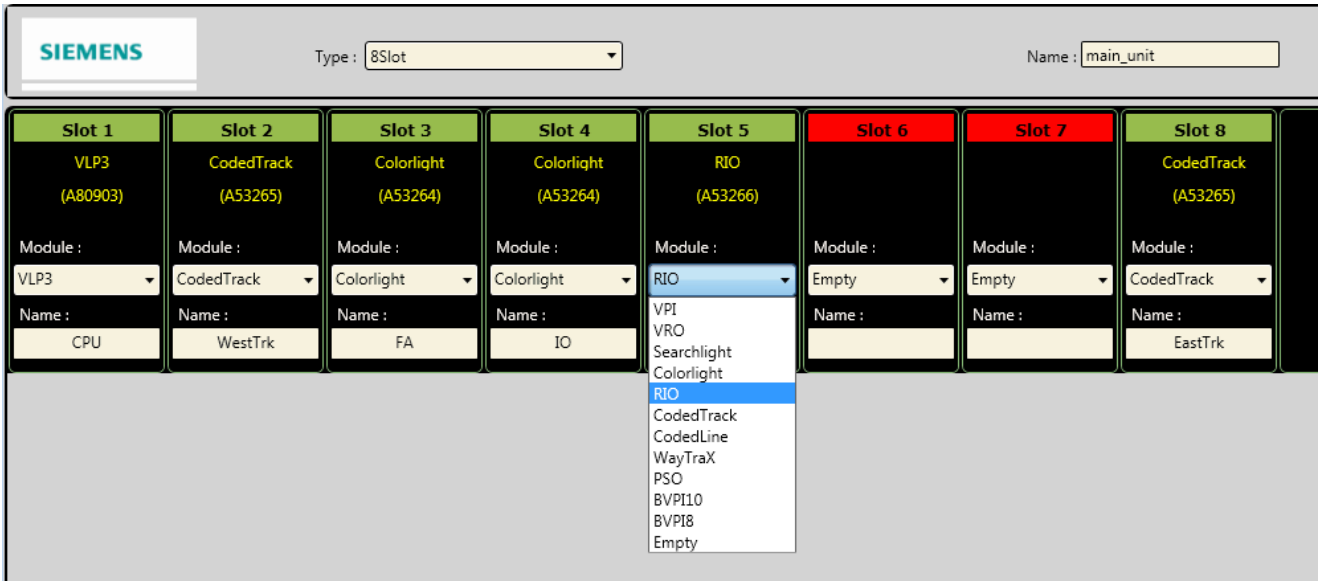


Figure 4-12 Adding New Modules

To edit the module, either double-click it or right-click and select **Edit** from the drop down menu. The drop-down menu also allows the module to be deleted, copied, or the last change undone.

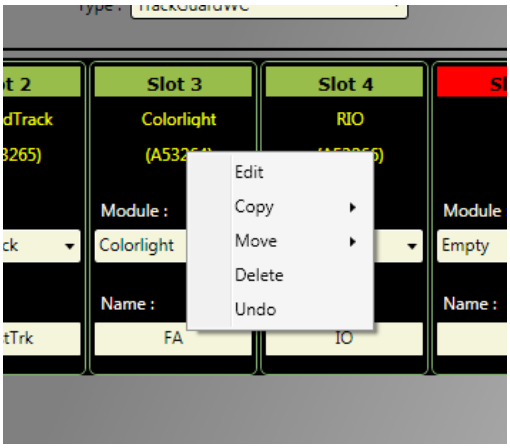


Figure 4-13 Edit or Configure Modules

4.4 SETTING MODULE CONFIGURATION AND I/O NAMES

Double-click on a module or right-click and select **Edit**; this will bring up windows that allow the user to set the configuration parameters for each module and assign the names to the I/O points that are used in the application logic (see section 3.0).

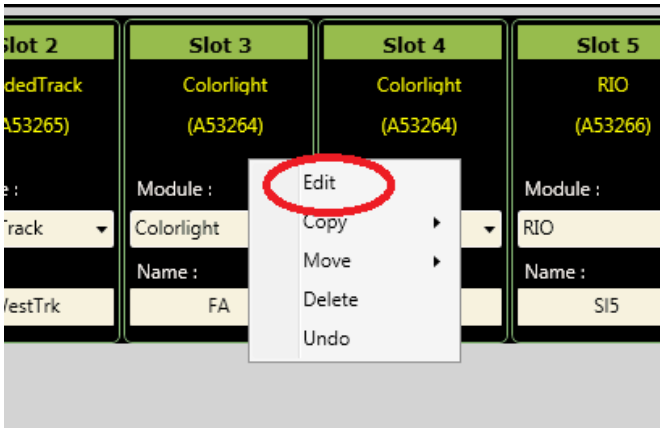


Figure 4-14 Setting Configuration Parameters

Selecting the **Edit** button for the **CodedTrack** slot, for example, brings up the following window which allows you to set the coded track configuration parameters. Note that these are the default configuration parameter settings for this MCF. The actual values to be used in a specific installation can be changed using the Office Configuration Editor (OCE) or set in the field. In the screen below, **default** indicates the parameter is still at its default value. UCN indicates the parameter is included in the UCN calculation. Selecting the **Default** button will set all parameters back to their default values.

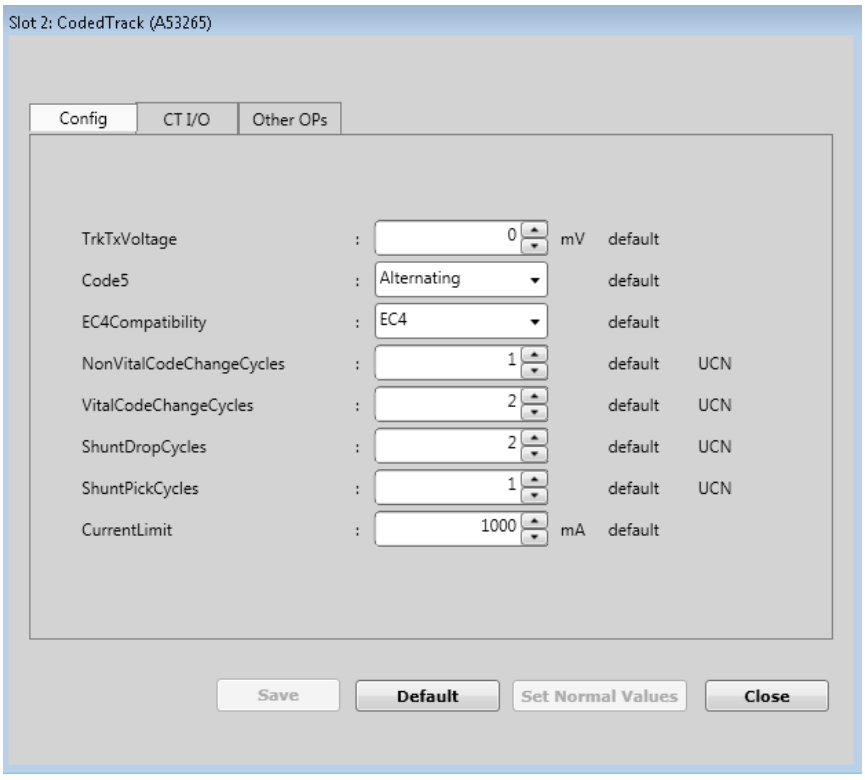


Figure 4-15 Configuration Editor

To assign names to the I/O to be used, select one of the tabs at the top of the window. Selecting CT I/O, for example, shows the available fields (as shown in the following figure). For coded track modules, a channel name needs to be entered, then the names for the Boolean variables (representing codes to be transmitted or received) can be added.

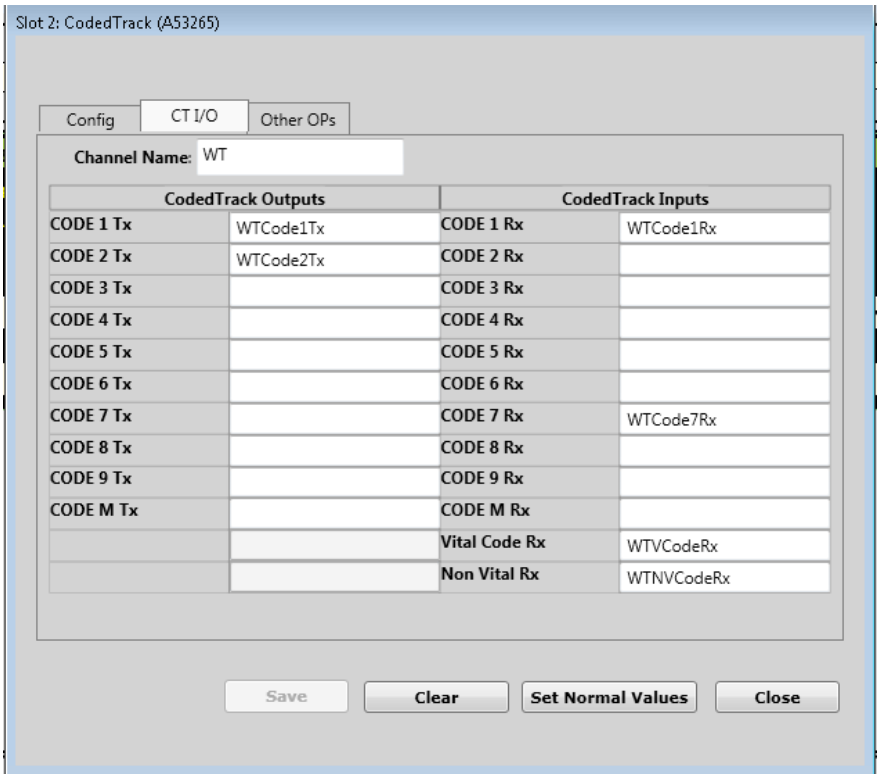


Figure 4-16 Assigning Names to the I/O

To save changes at any time, use the **Save** button and continue. To delete all entries, select **Clear**. To select the normal states of the relay contacts that will be shown in the relay logic for these inputs, select the **Set Normal Values** button. Note: this button can only be selected once inputted values are saved. This will bring up a window that allows the selection of the Normal Values for each state that has been defined for this module. Press the **Close** button once you are done setting the Normal value.

Note: Since there are no scroll bars on the Normal State windows, use the up/down arrow keys to navigate.

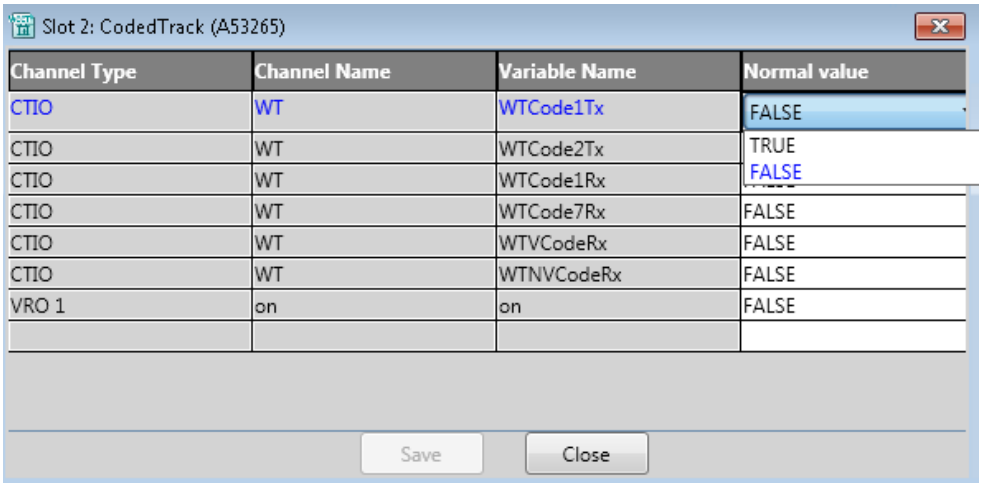


Figure 4-17 Normal State Window

Once the selected module is complete, close the window and select the next module. For the Colorlight module, follow the same process of right-clicking and selecting **Edit.** This will bring up its configuration parameters.

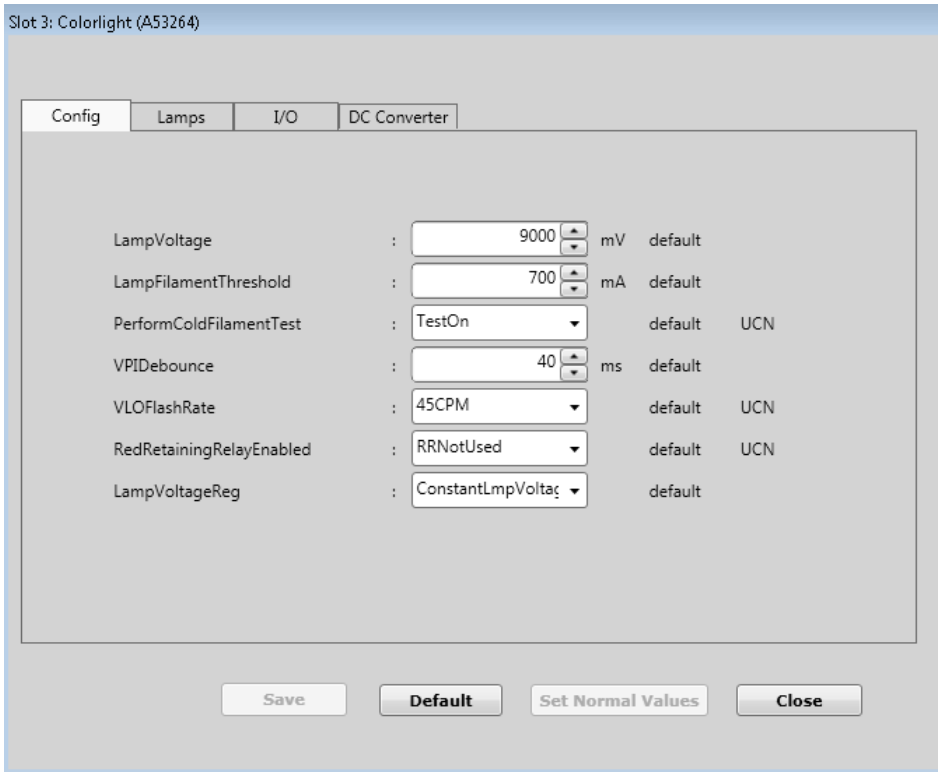


Figure 4-18 Saving Changes

NOTE

NOTE

If RedRetainingRelayEnabled is set to RRUsed, then the DC Converter must be controlled by the application logic.

If no converter channel has been defined in the **DC Converter** tab, the WCCT will give a warning message that the converter channel needs to be defined. Before leaving the Colorlight module editor, go to the **DC Converter** tab and add in the necessary channel and logic state names (see section 3.9.4)

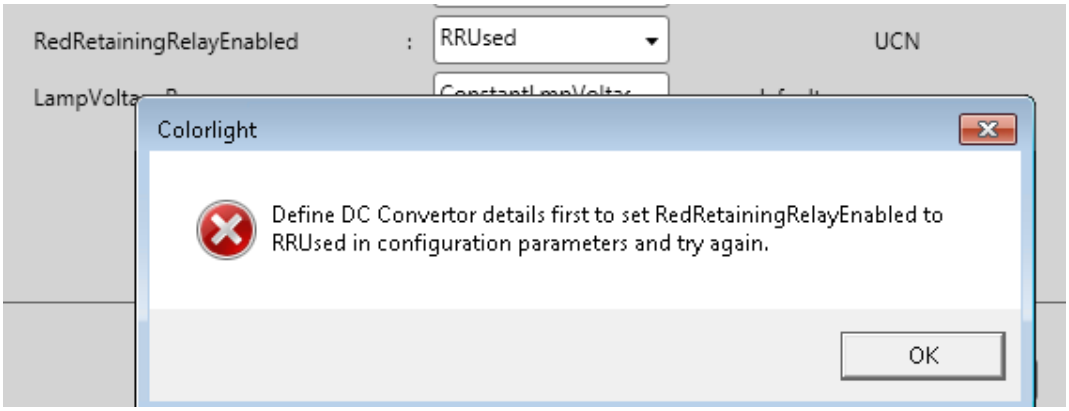


Figure 4-19 DC-DC Converter Warning

After the configuration changes have been made, enter names for the lamp controls and lamp feedback.

NOTE

NOTE

Ctrl C can be used to copy names and Ctrl V to paste.

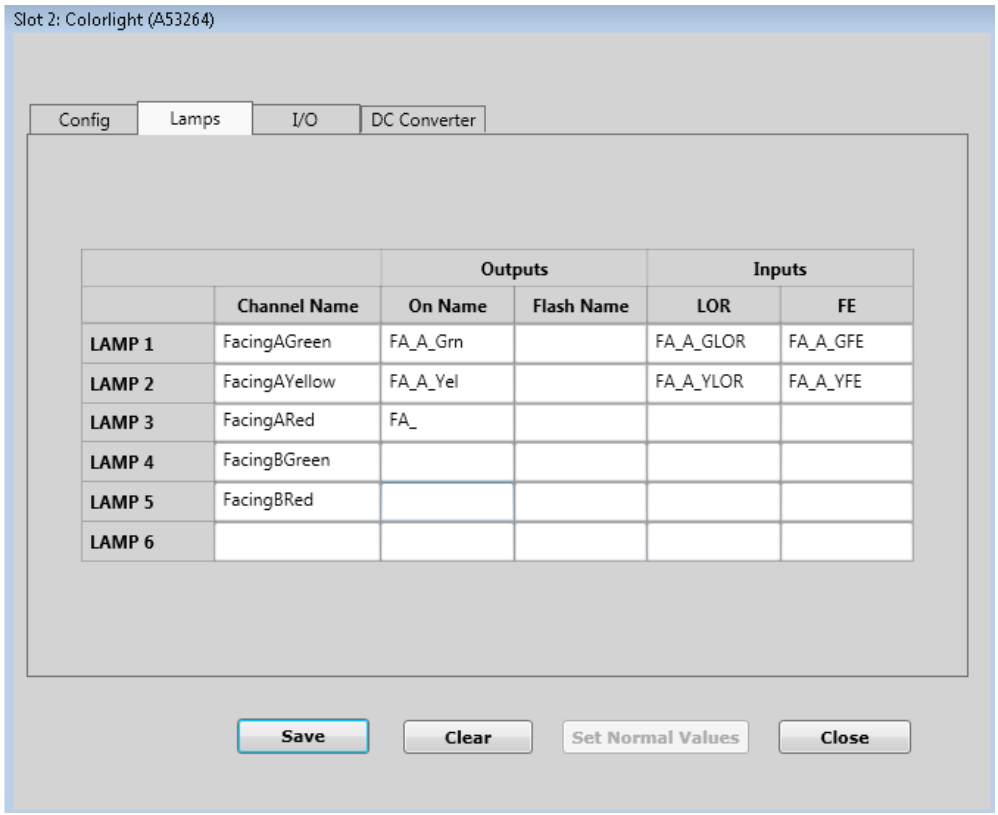


Figure 4-20 Entering Names for Lamps and Controls

If a duplicate name is found, the WCCT will indicate an error.

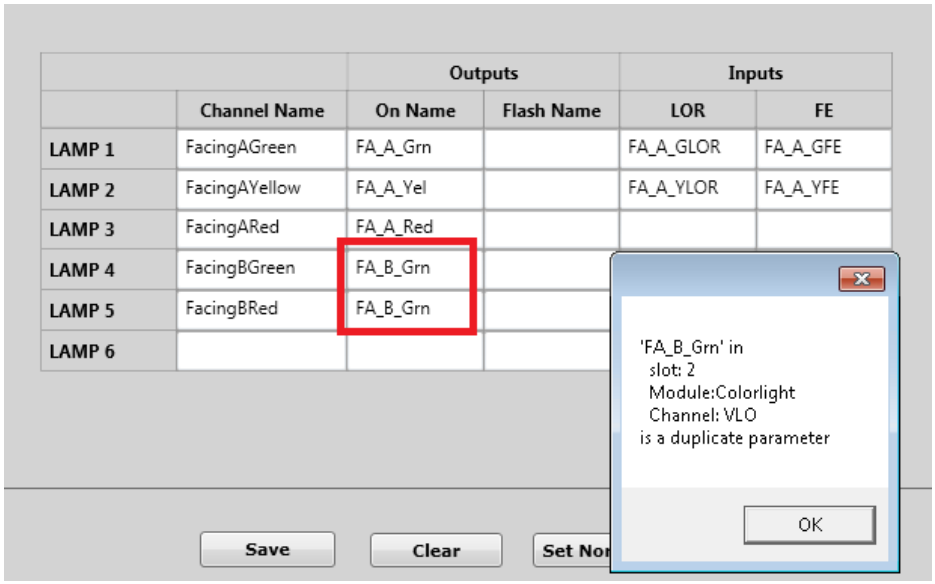


Figure 4-21 Duplicate Name Error Message

After the lamp control and feedback names have been entered, enter names for the vital input and outputs.

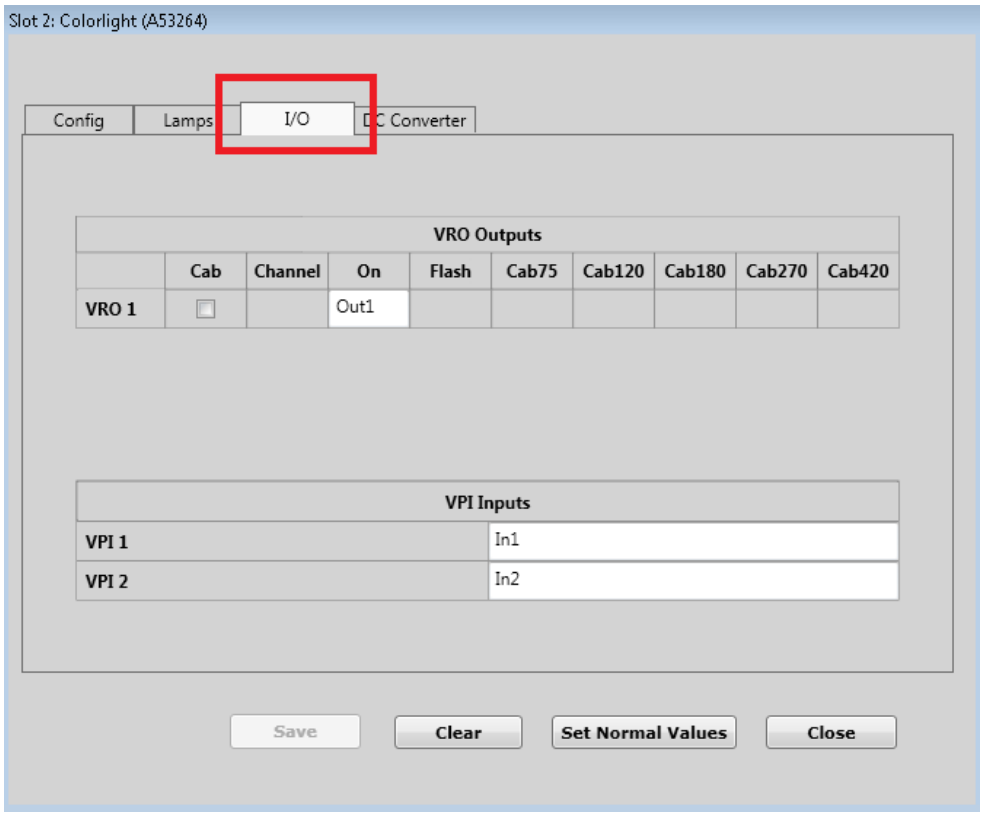


Figure 4-22 Entering Vital Input and Output Names

If a signal is split over multiple modules, you may need to write logic to synchronize the DC-DC converters on the modules (see Section 3.9.4), in this case, select the DC Converter tab and enter a name for the Boolean states that will be used to control the converter and monitor its state.

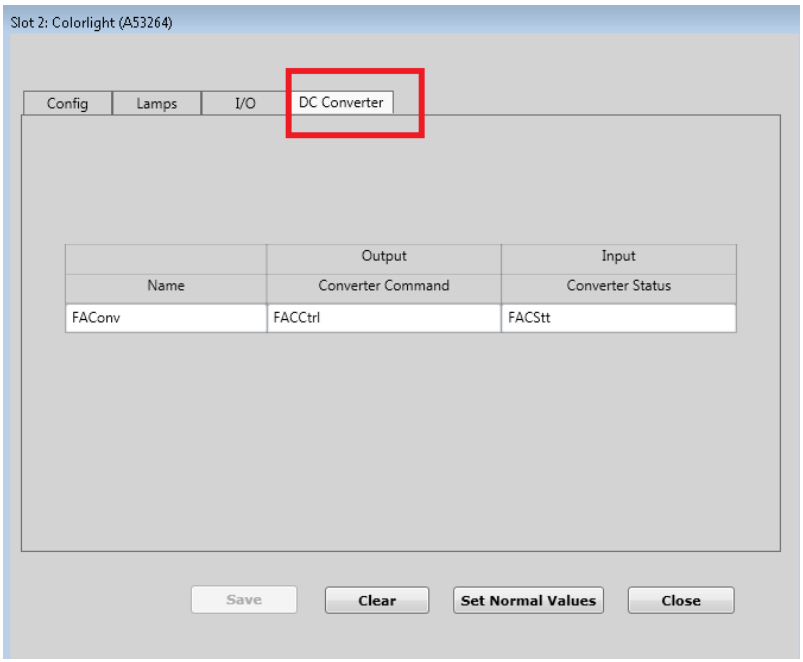


Figure 4-23 Entering Boolean State Names for Converters

Proceed through each module, setting the configuration and I/O points. See individual I/O Module editor sections (Sections 3.4 through 3.16) for more details.

4.5 SETTING VITAL COMMUNICATIONS BETWEEN MULTIPLE WAYCONNEX/GEOS

If the application uses multiple WayConneX chassis that need to communicate to each other, you will need to create vital communication links (VComms channels) between them. Double-click on the Slot 1 VLP2/3 module and then select the Vital Comms Tab. This will bring up the screen shown in the following graphic. Up to 10 vital communication channels can be created.

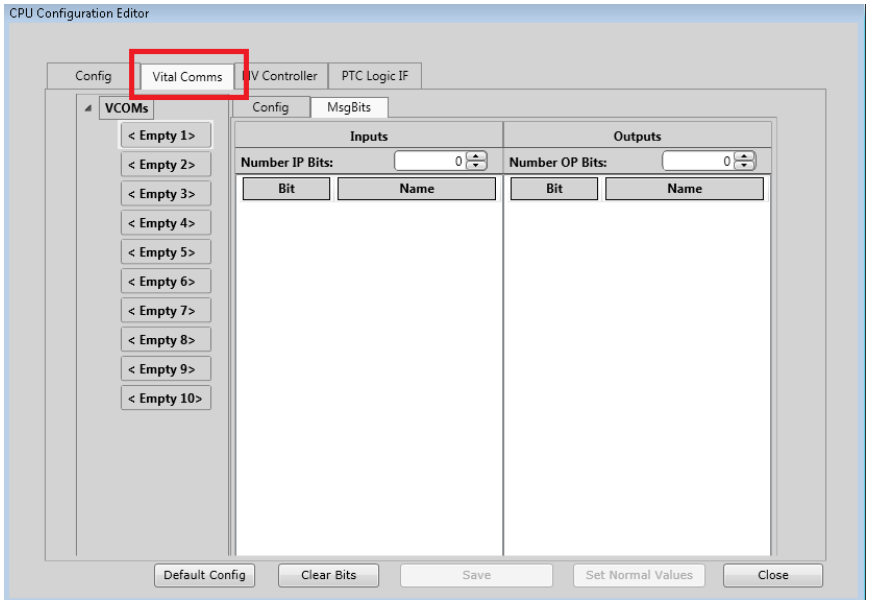


Figure 4-24 Setting Vital Communication Links for Multiple WayConneX/GEOs

Select the required number of input and output bits for the first channel. Note that bits can be left empty if necessary.

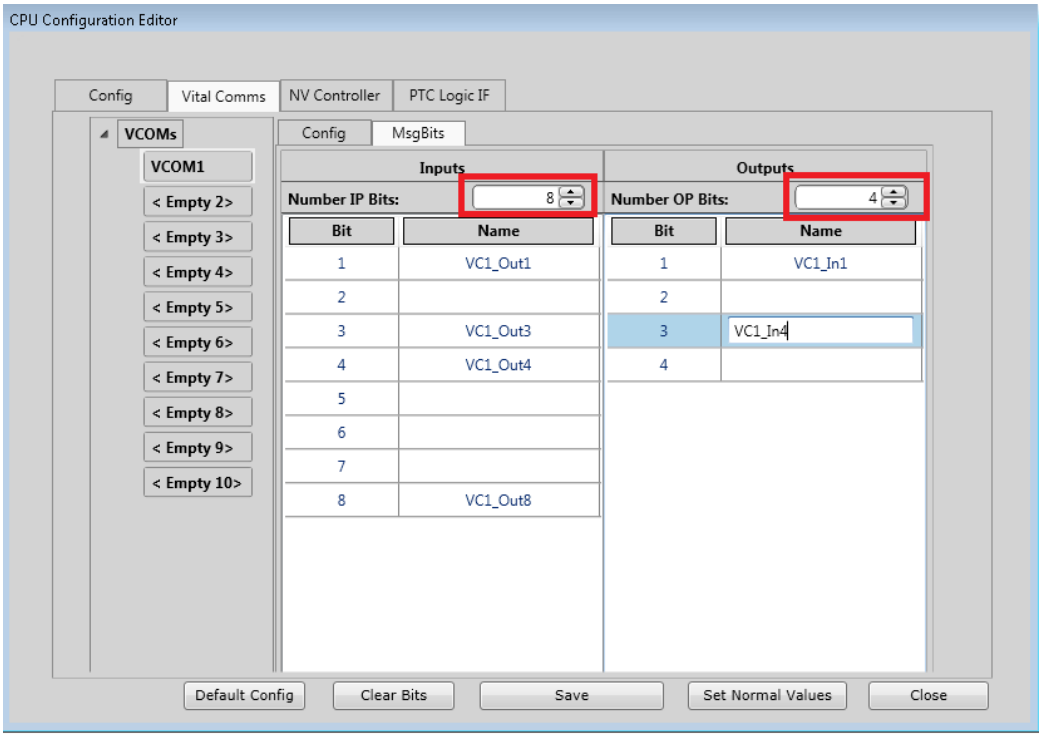


Figure 4-25 Selecting Number of Input and Output Bits

The user can set normal values to be displayed for these inputs and outputs by selecting the **Set Normal Values** button. This will bring up a window like the one shown below where the user can select the Normal values.

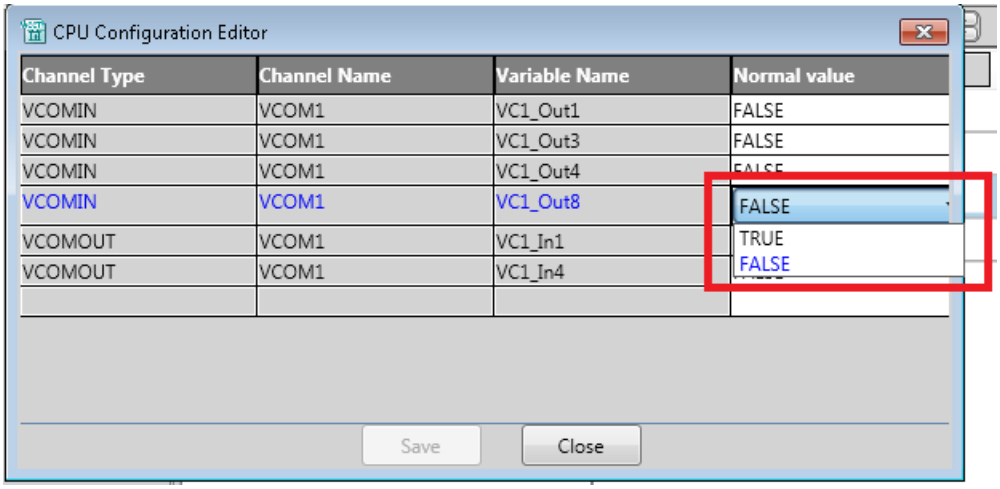


Figure 4-26 Setting Normal States

Select the **Config** tab to show the configuration properties for this VComms channel. Note as with all configuration parameters, the actual values used for a particular installation in the field will be set using the OCE.

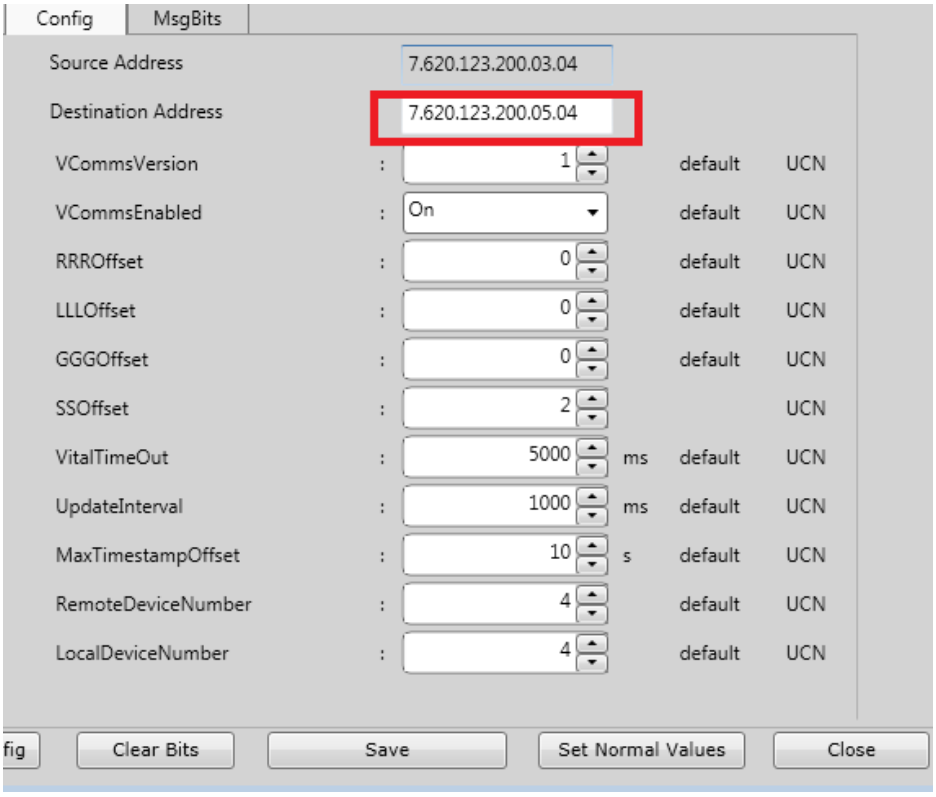


Figure 4-27 Vital Communication Parameter Configuration per Channel

The source ATCS address of the unit shown is set in the Project Properties. The destination ATCS address of the neighbor can be set either by editing the Destination address text box, or by changing the offsets (RRROffset, LLLOffset, GGGOffset, SSOffset). The WCCT will limit the value of the offsets entered so that the resulting address is not illegal.

When you change the value in the Destination Address, the WCCT will automatically recalculate the offsets. Similarly, if you change the offsets, the WCCT will recalculate the Destination address. See section 3.4.2.1 for more details.

NOTE

NOTE

Each VComms channels must have a unique set of ATCS source/destination addresses. If multiple VComms channels are used between the same WayConneX/GEOs, give them unique Local and Remote Device Numbers.

4.6 SETTING INTERFACE TO A NON-VITAL CONTROLLER

If the application is at a control point and is required to receive controls from the dispatch system and send back indications to it, a non-vital controller such as the SEAR II can be used. The SEAR II will receive a control message from the dispatcher and, using its internal non-vital logic, will send a message on to each WayConneX (GEO) it is connected to. Each WayConneX (GEO) will send an indication message back to the SEAR II, and the SEAR II will consolidate this into one indication message back to the dispatcher.

In order to define the format of the Control and Indication message for one particular WayConneX (GEO), select the NVController tab. This will bring up a screen like the one illustrated in the following graphic.

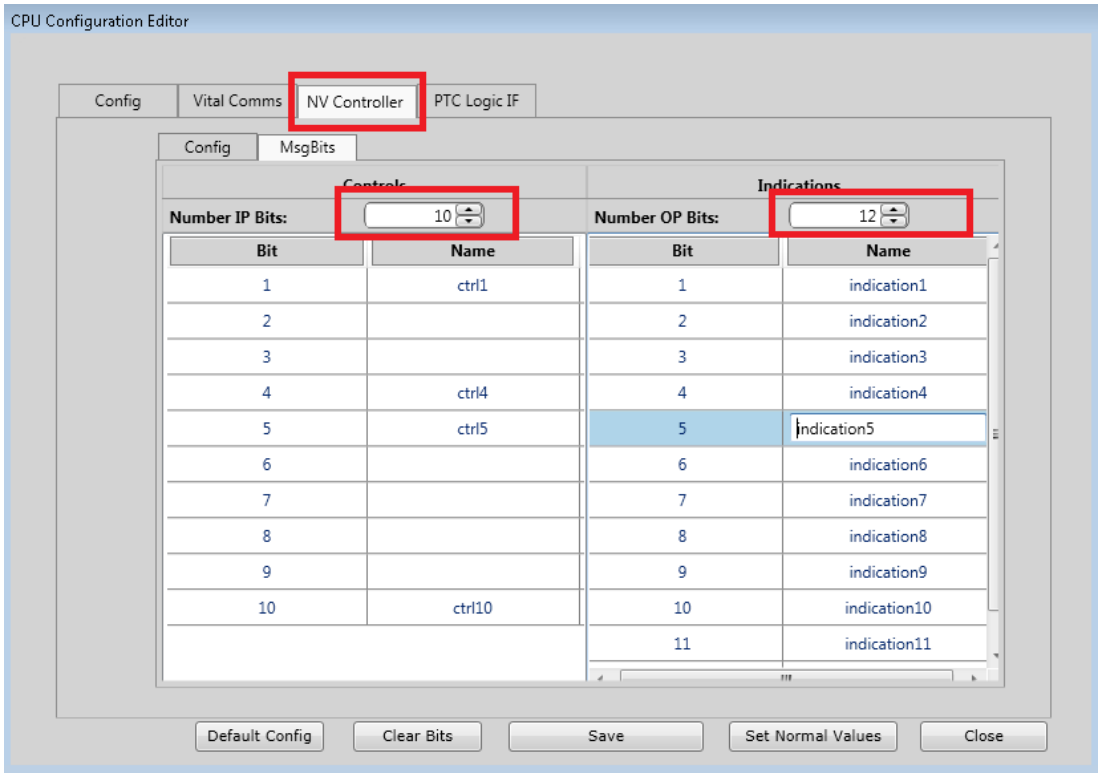


Figure 4-28 NV Controller Tab Display

Select the desired number of control bits and indication bits in the message. Then fill in the names to be used in the logic for the control (input) bits and indication (output) bits. Unused bits can be left blank. The WCCT will check to make sure the names used in this screen are unique. The user can set the normal values to be displayed for these inputs and outputs by selecting the **Set Normal Values** button.

4.7 SETTING INTERFACE TO PTC SYSTEM

If the application is required to send PTC messages to PTC equipped locomotives, the WCCT provides a straightforward way of defining the PTC message. PTC messages are only supported for WayConneX applications using CPU III. Selecting the **PTC Logic IF** tab will bring up the following screen.

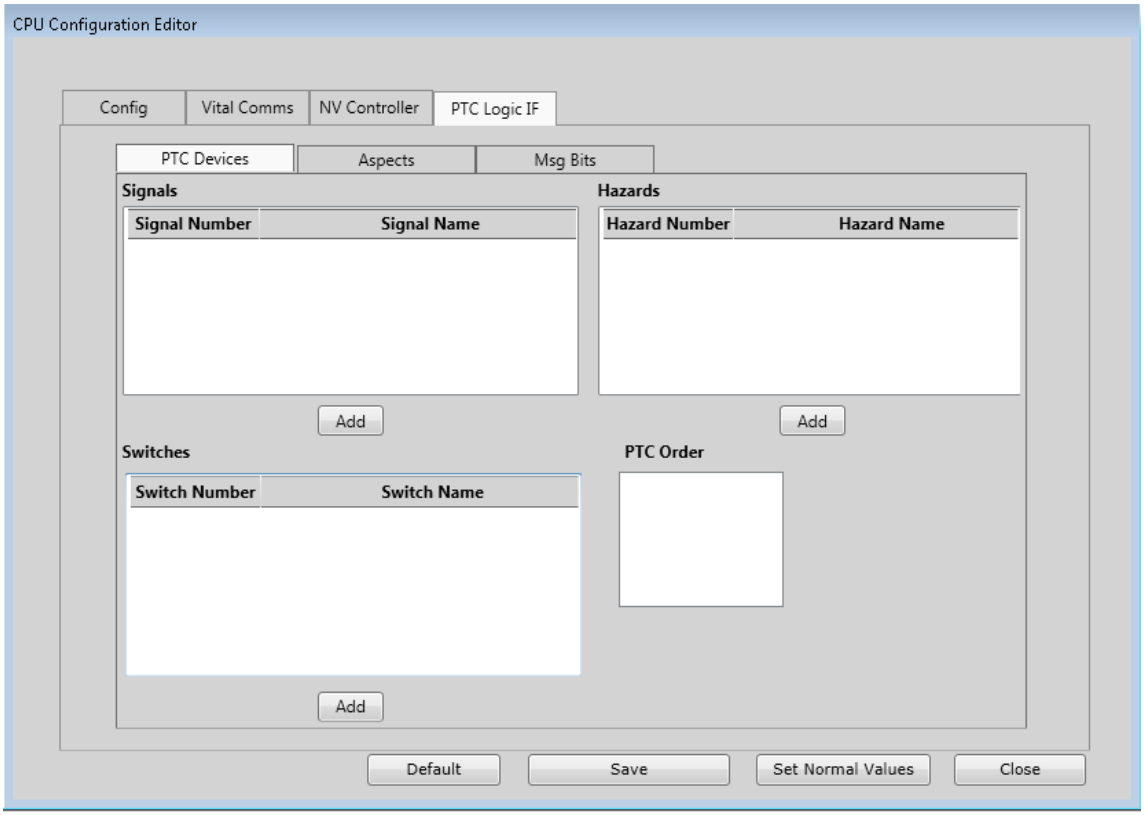


Figure 4-29 PTC Logic IF Tab Display

For each signal used in the application, press **Add** to add a signal and enter its name. The same process is repeated for switches and hazard detectors. The following figure illustrates a location with two signals, one switch, and one hazard detector.

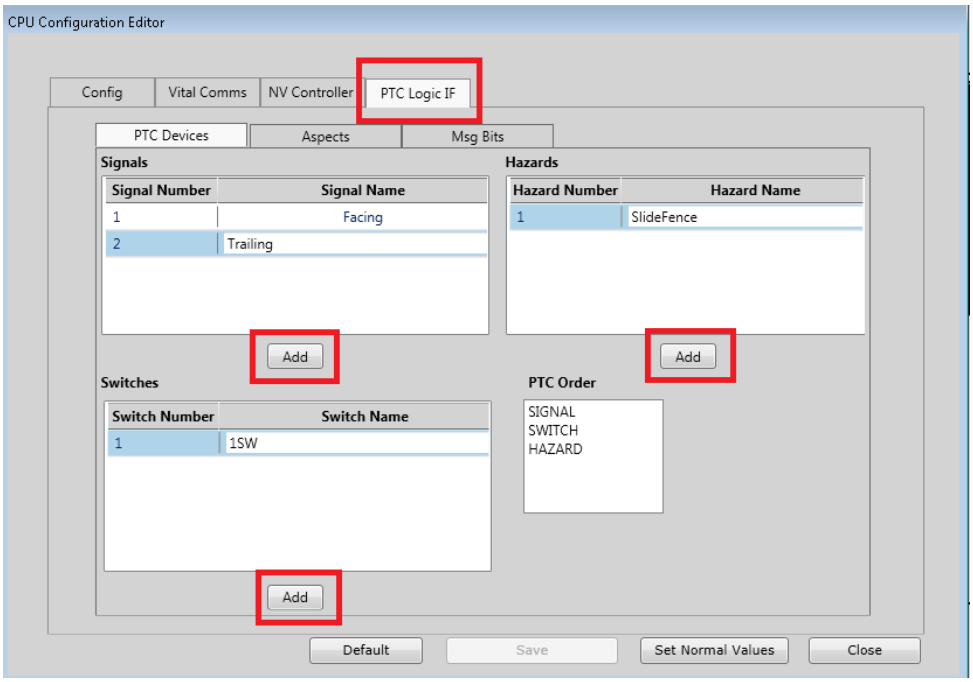


Figure 4-30 Adding Signals, Switches, and Hazards

The default order for reporting PTC devices is Signals, Switch, and Hazard detector as shown in the text box in the lower right corner. Once the PTC devices have been entered, select **Save**, then, if signals are used, select the **Aspects** tab. On this tab the user can add in the signal aspects and their respective PTC Codes that are used by all the signals in this application. Two fixed aspects (**Illegal Aspect** and **All Dark Aspect)** are always present. By default they have the codes of 0 and 30 respectively, but these can be changed if required.

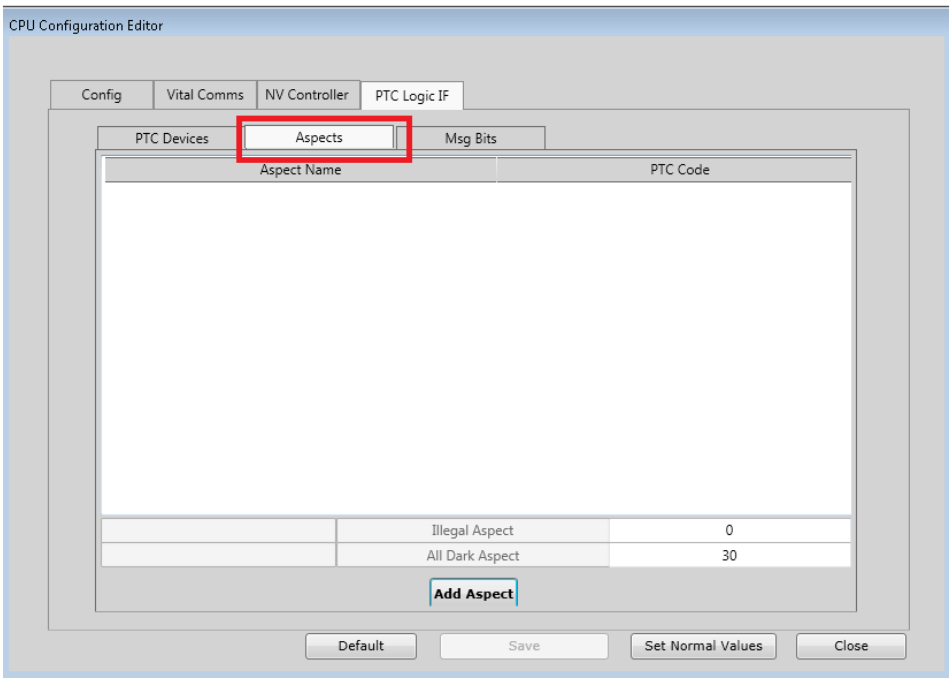


Figure 4-31 Aspects Tab

Press the **Add Aspect** button to add in new aspects and define their PTC codes (see figure below for example).

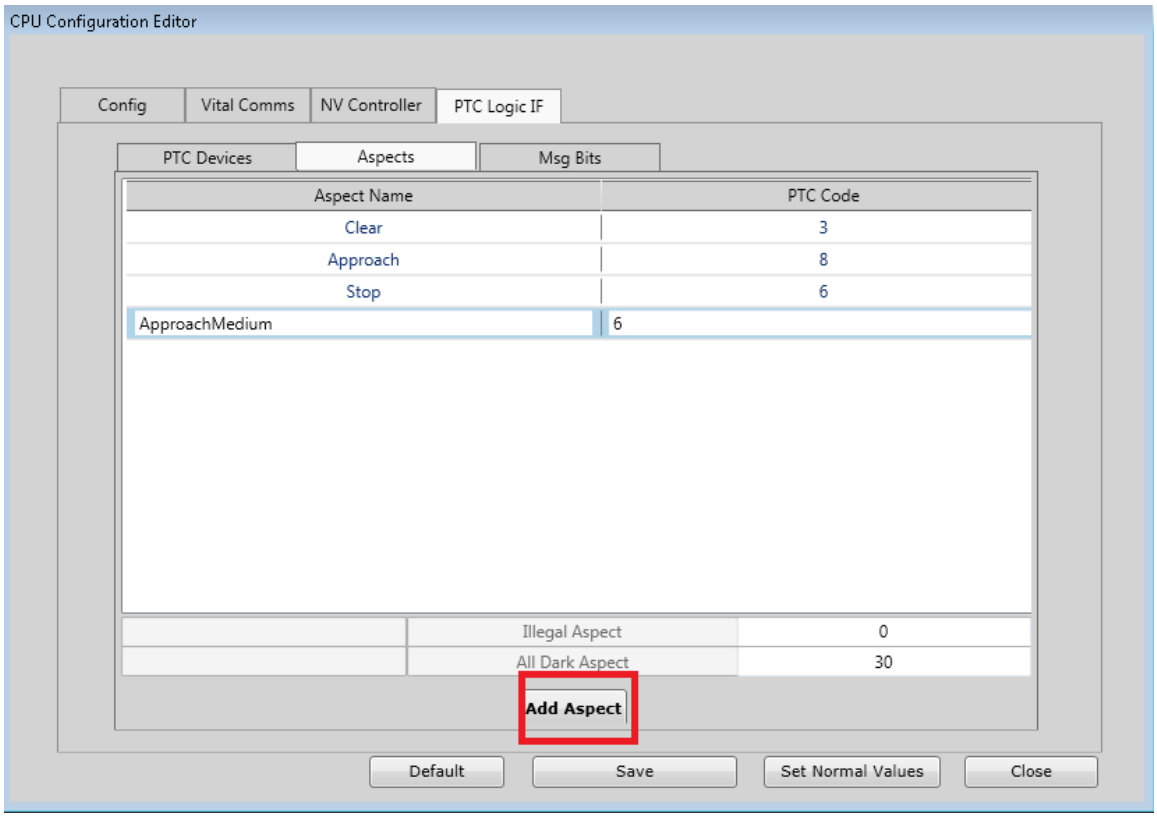


Figure 4-32 Adding Aspects and Defining PTC Codes

Spaces are not allowed in aspect names. Up to 30 aspects can be entered in addition to the **IllegalAspect** and **AllDarkAspect**. Multiple aspects can be entered with the same PTC code, so the same MCF could be used across multiple territories where the aspect is given different names.

The next step is to assign states to control the bits in the PTC message. Select the Msg Bits tab. This will show a list of all the PTC Devices on the left. For each signal, the names of each possible aspect entered in the aspect tab are shown.

Enter a logic state name for each aspect the respective signal is capable of displaying. If the signal does not use an aspect, leave the entry blank.

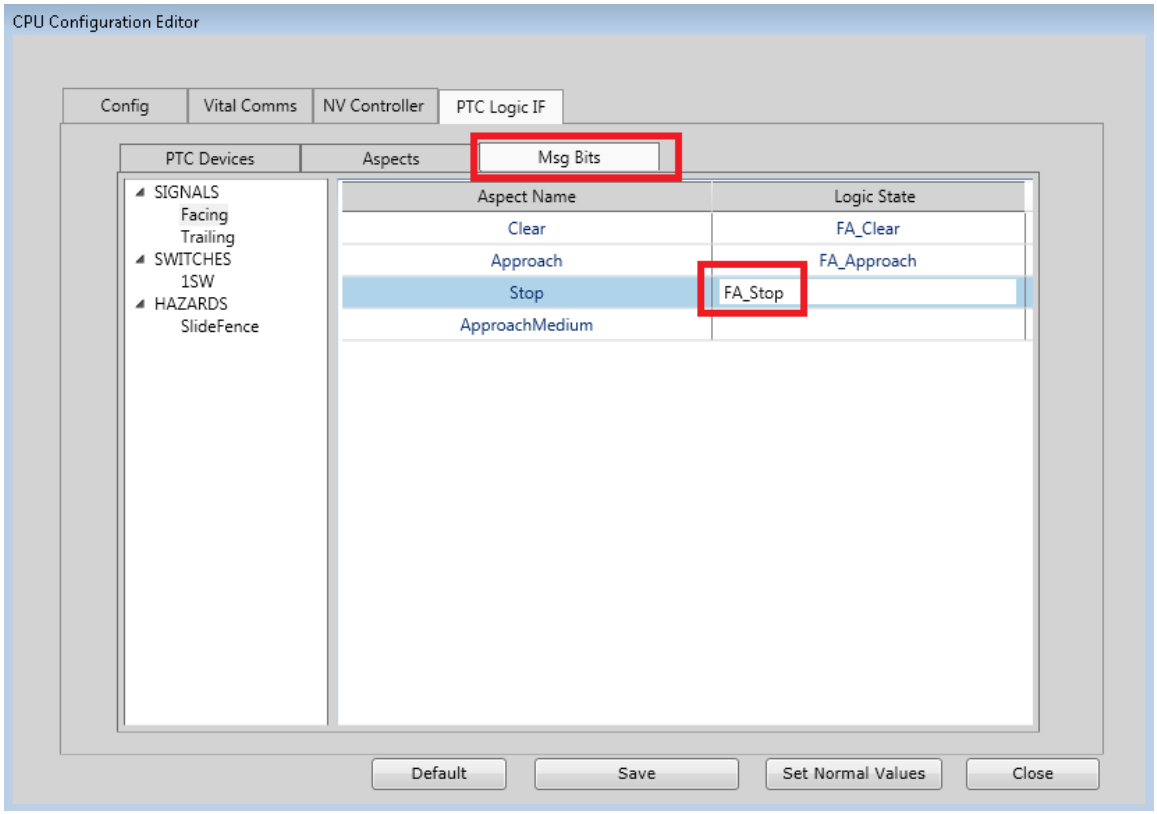


Figure 4-33 Entering Logic State Names

The WCCT will automatically create output logic states for the logic state names. When writing the application logic, set the states of these to the variables that indicate when the signal is displaying this aspect. Only one aspect bit should be set at once. If multiple aspect bits are set, the CPU III will send a PTC code with the value you defined for IllegalAspect. If no aspect bits are set, the CPU III will send a PTC code with the value you defined for AllDarkAspect.

Repeat this for each Signal.

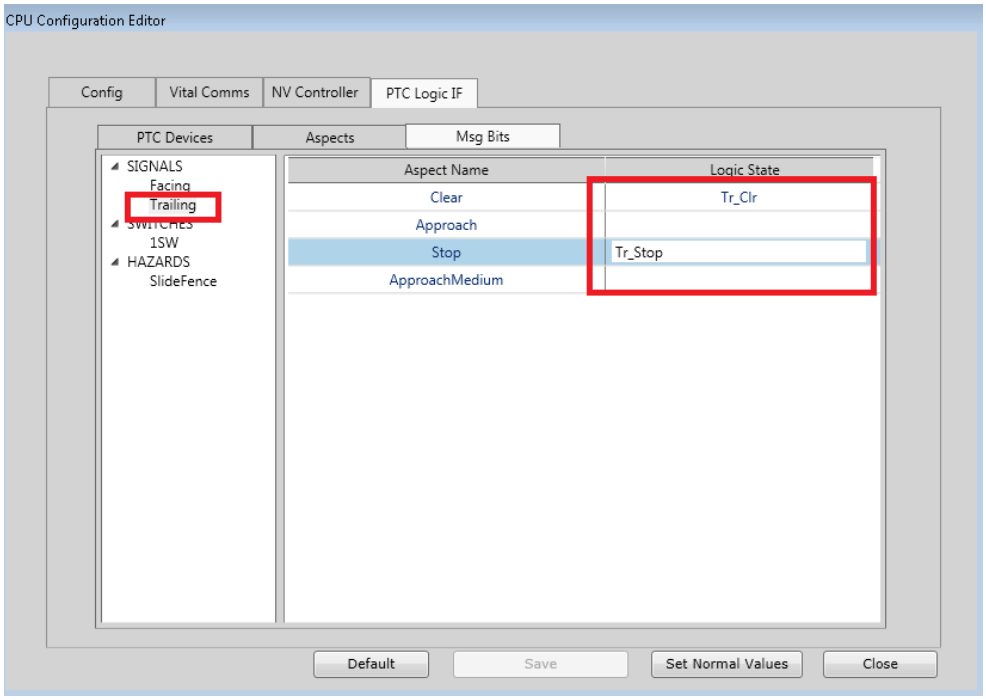


Figure 4-34 Setting Logic State Variables

Once the signals have been completed, select the first switch, and enter the names of the logic states that will be used to report the switch position. Note that this will create new logic output variables with these names. If a duplicate name is entered, the WCCT will give an error and indicate fixing the duplicate is needed. If internal states already exist in the logic that indicate normal or reverse, write logic equations to set the logic state names entered on this tab to the values that already exist.

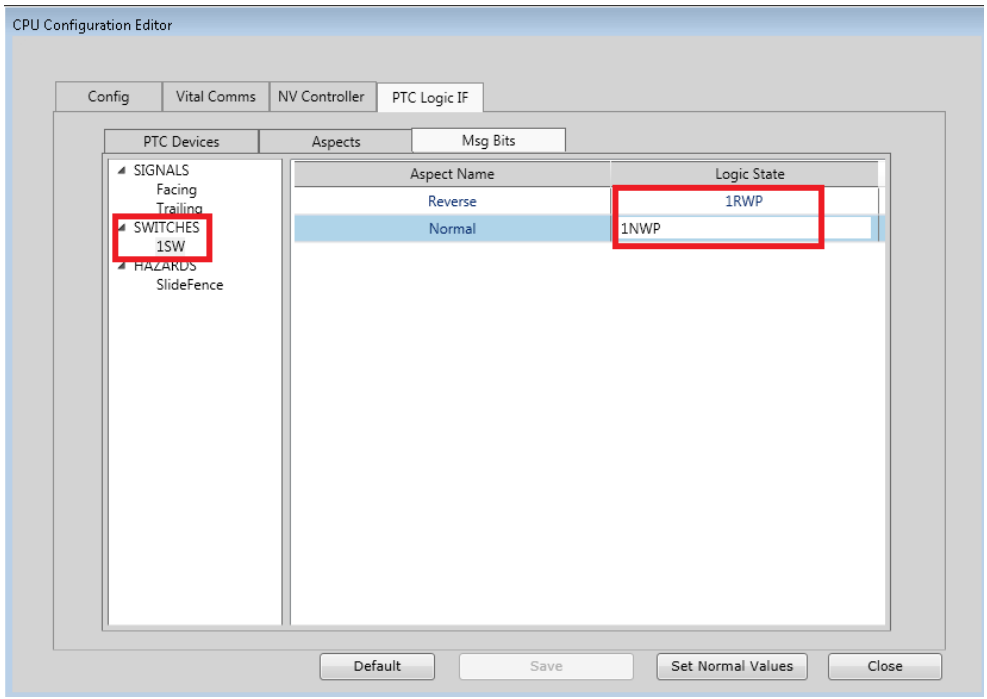


Figure 4-35 Setting Logic State Names for PTC Logic IF

Once all switches are complete and the data saved, select each **Hazard Detector** and enter the names for the fault bit associated with the Hazard Detector.

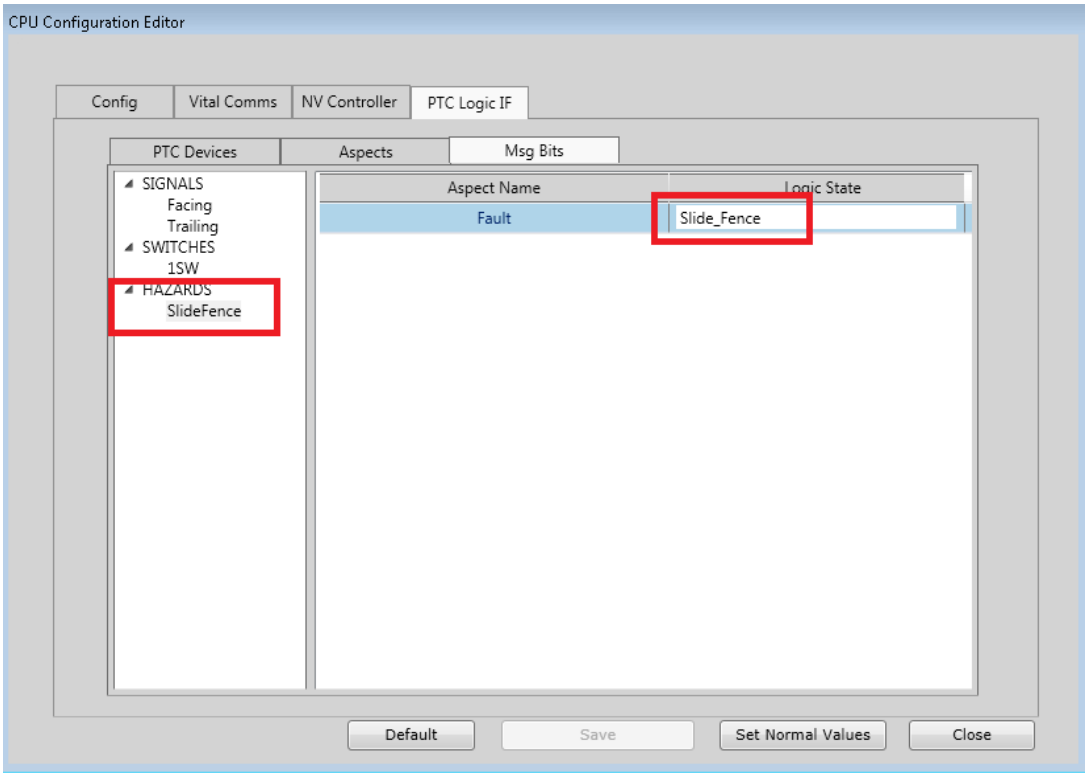


Figure 4-36 Entering the Fault Bit Name for Hazard Detector

The user can set the normal values to be displayed for these PTC inputs and outputs by selecting the **Set Normal Values** button.

4.8 CREATING BOOLEAN VARIABLES

To add Boolean variables, timers, and field configurable properties, bring up the Logic Data Editor by selecting the **Logic Data** button or clicking **Logic Data** from the Edit menu or use the **Ctrl L** shortcut.

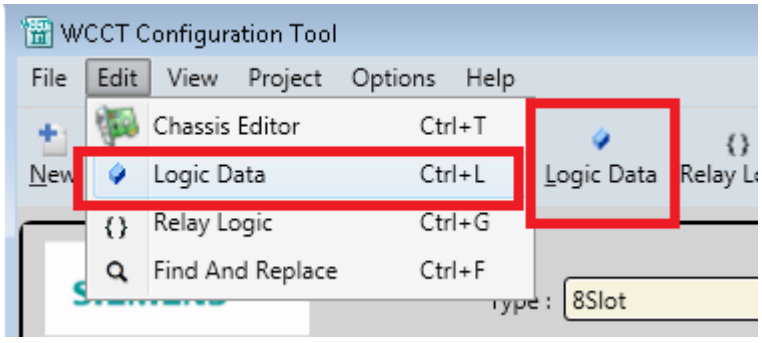


Figure 4-37 Logic Data Editor

This will then open the Logic Data Editor at the Boolean Variables tab.

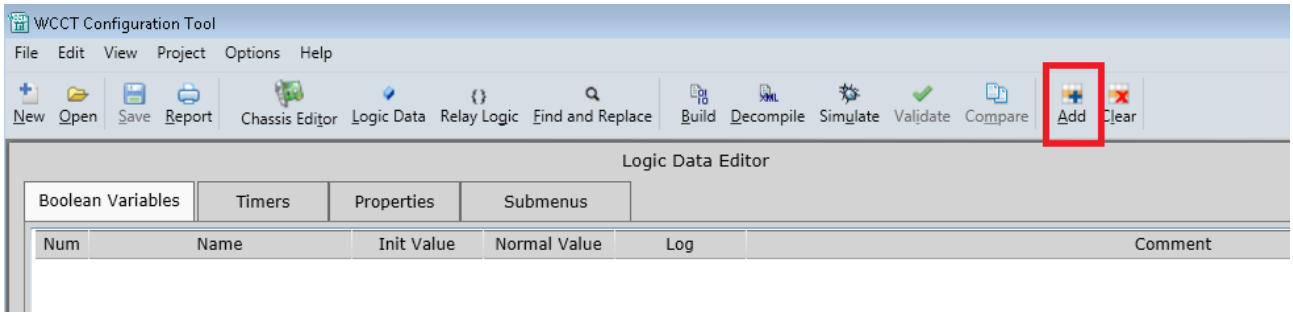


Figure 4-38 Boolean Variables Window

Select **Add** to add new internal Boolean logic states. Enter the name, the initial value (default is false), and whether the variable should be logged on change of state in the CPU log. A comment can be added if necessary next to each variable, multi-line comments are allowed.

The WCCT allows up to 3000 variables. The names can be up to 16 characters long. If a VLP3 is used, up to 300 variables can be marked as being logged. If a VLP2 is used, up to 50 variables can be marked as being logged. When a variable is marked as logged, the CPU will create an entry in its log when that variable changes state. Note that input and output variables defined in the module editors are always logged on change of state.

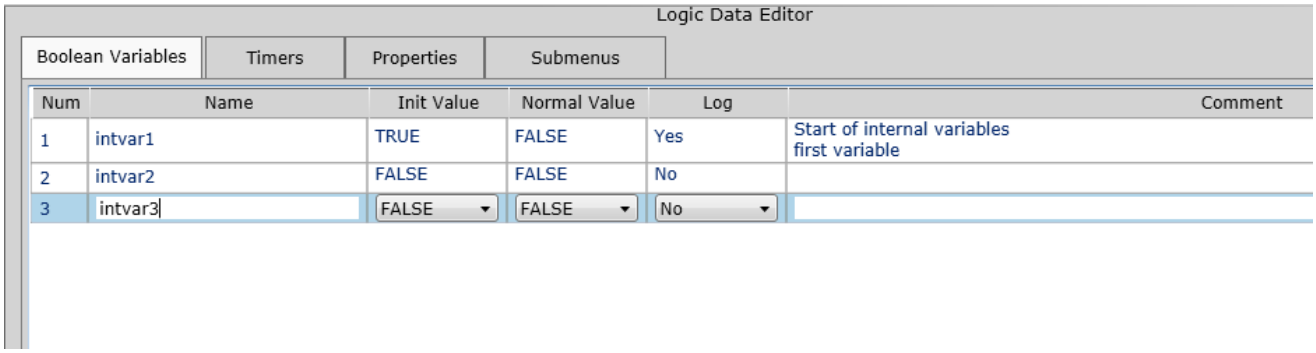


Figure 4-39 Adding Internal Boolean Logic States

Use the **Save** button (or Ctrl S) to save changes. The save button is enabled whenever there are pending changes.

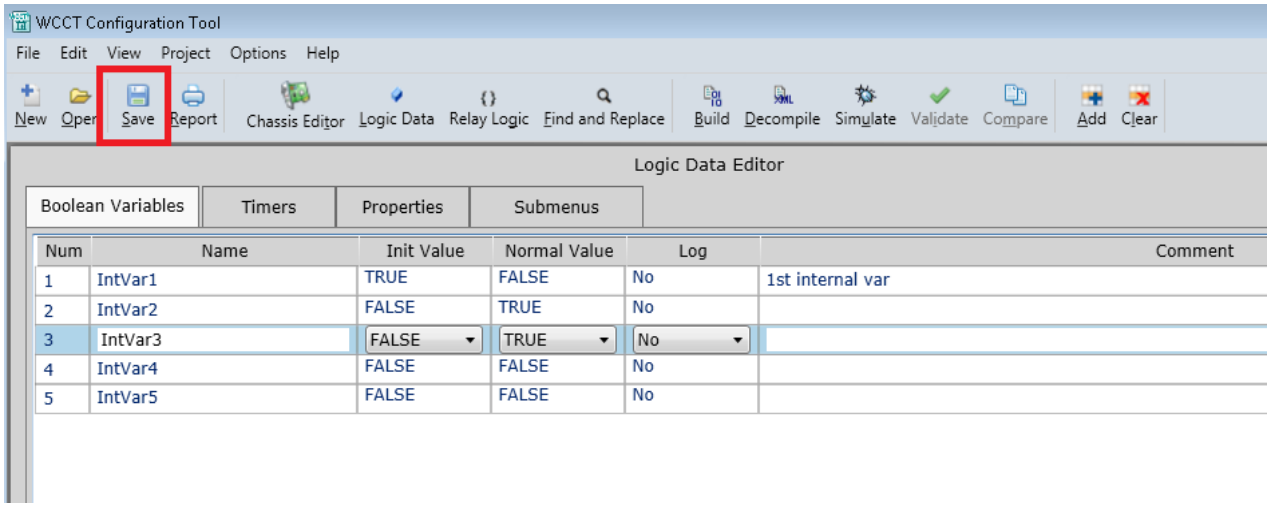


Figure 4-40 Saving Logic Data

The normal values of the relays corresponding to these variables shown on the relay view can be set by selecting the **Normal Value**. If you right-click on the Num column, a drop down menu is shown that allows rows to be moved up, down, new rows inserted, copied or pasted. Note the standard **Ctrl C** and **Ctrl V** short cuts are very useful for copying and pasting whole rows.

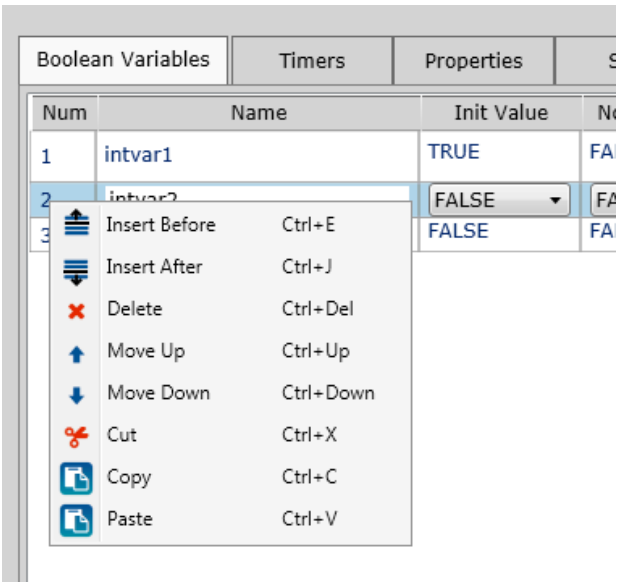


Figure 4-41 Options in Boolean Variables Window

4.9 ADDING TIMERS

Select the **Timers** tab from the Logic Data Editor and then press **Add** to add new timers.

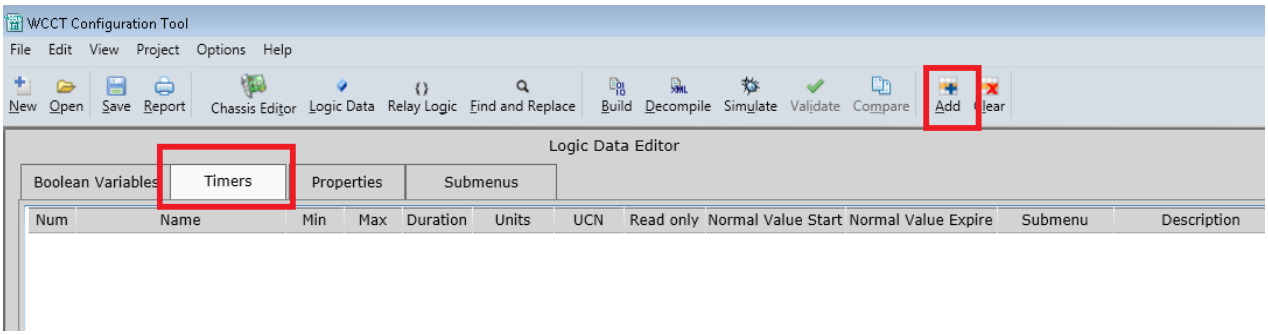


Figure 4-42 Adding New Timers

For each timer, enter:

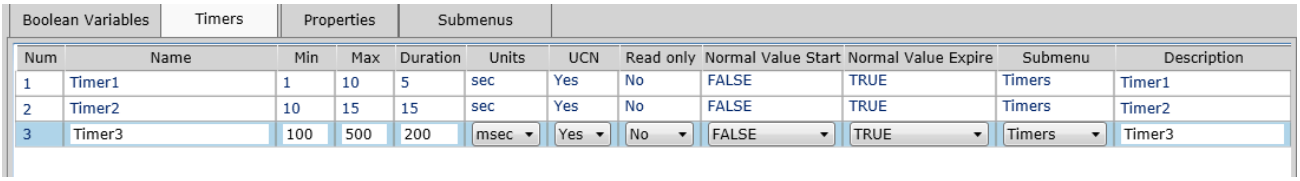
- Name (this is the name that will be used in the logic equations, so no spaces are allowed).
- Minimum and maximum values for the timer.
- Duration in seconds or milliseconds (depending on units).
- Units –select either seconds or milliseconds. Note that if milliseconds is selected as the units, the timer duration and range have to be entered in 100ms increments.
- Select whether the timer value is included in the UCN calculation or not.
- Select whether a timer is read-only or not. When the timer is read-only it is shown in the menu system of the WayConnex/GEO, but is not allowed to be edited.
- Set the Normal Value for Timer Start and the Normal Value for Timer end. These are used to display the correct relay coil/contact state in the relay editor.
- Enter a description for the timer. The description is the name the timer will have in the user interface menus. Spaces are allowed in this text. A description has to be added. If the user presses **Save** without entering a description, the WCCT will create a default description for the timer matching the user entered timer name.
- Select the submenu that the timer will appear on in the menu system of the WayConneX. This will default to the predefined **Timers** menu. To have the timer displayed on a different submenu, first go to the submenu tab and create a new submenu.
- A comment can be added for each timer if required. The comment is optional.

⚠ WARNING

WARNING

TIMERS WHOSE INCORRECT SETTING COULD CAUSE A SAFETY ISSUE MUST BE INCLUDED IN THE UCN TO PREVENT UNAUTHORIZED CHANGES.

The DT, OCE (for GEO), or OCE (for WayConneX) will be used to define the timer and property settings for a particular installation of an MCF, and obtain the required UCN. The VLP3 used in WayConneX supports up to 120 timers and the VLP2 used in GEO supports up to 32 timers.



Num	Name	Min	Max	Duration	Units	UCN	Read only	Normal Value Start	Normal Value Expire	Submenu	Description
1	Timer1	1	10	5	sec	Yes	No	FALSE	TRUE	Timers	Timer1
2	Timer2	10	15	15	sec	Yes	No	FALSE	TRUE	Timers	Timer2
3	Timer3	100	500	200	msec	Yes	No	FALSE	TRUE	Timers	Timer3

Figure 4-43 Setting Timer Parameters

See the Timer Editor Section 3.17.2 for more details. If you right-click on the Num column, a drop-down menu is shown that allows rows to be moved up, down, new rows inserted, copied or pasted.

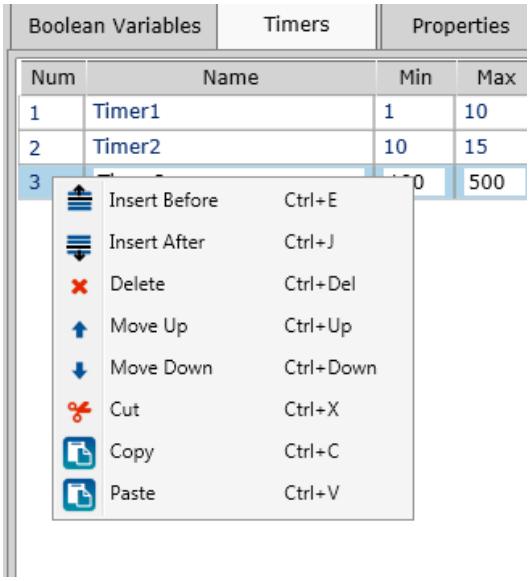


Figure 4-44 Options in Timer Window

4.10 ADDING FIELD CONFIGURABLE PROPERTIES

An application that can be used in multiple field sites may require different field configurable settings to tailor the logic to that specific site. To create these field configurable settings, select the **Properties** tab on the Logic Data Editor.

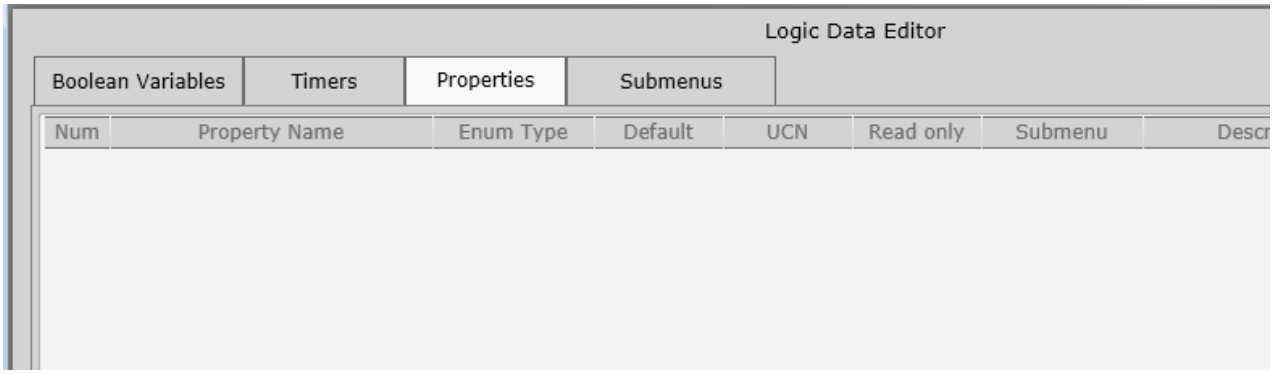


Figure 4-45 Adding Field Configurable Properties

Click **Add** to enter new properties.

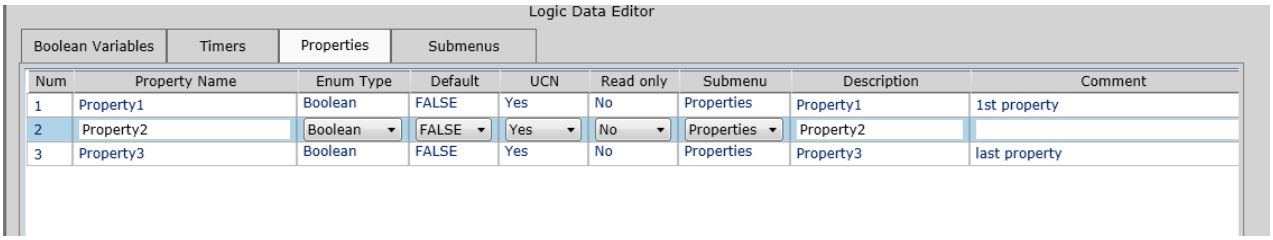


Figure 4-46 Adding New Properties

For each property:

- Enter the property name – this is what will be used to reference the state of this property in the logic.
- The enum type will default to Boolean. Currently, this is the only supported type.
- Select the required default from the drop down menu; default is true.
- Select whether the property value is included in the UCN calculation or not.
- Select whether a property is read-only or not. When the property is read-only it is shown in the menu system of the WayConnex but is not allowed to be edited.
- Select the submenu that the property will appear on in the menu system of the WayConnex. This will default to the predefined **Properties** menu. To have the property displayed on a different submenu, first go to the submenu tab and create new submenus.
- Enter a description for the property - The description is the name the property will have in the user interface menus. Spaces are allowed in this text. A description has to be added. If the user presses **Save** without entering a description, the WCCT will create a default description for the property matching the user-entered property name.
- A comment can be added for each property if required. The comment is optional.

⚠ WARNING

WARNING

PROPERTIES WHOSE INCORRECT SETTING COULD CAUSE A SAFETY ISSUE MUST BE INCLUDED IN THE UCN TO PREVENT UNAUTHORIZED CHANGES.

The DT or OCE (for GEO) or OCE (For WayConneX) will be used to define the timer and property settings for a particular installation of an MCF, and obtain the required UCN.

The WCCT supports up to 100 properties.

All properties will appear on one menu screen along with timers, unless otherwise specified. In a CPU III application, if more than 25 (or 16 for a VLP2 application) properties and timers are used, these need to be placed on separate submenus. See Submenu Section 3.18 for details on how to create submenus.

If you right-click on the Num column a drop down menu is shown that allows rows to be moved up or down, new rows to be inserted, copied, or pasted.

4.11 ADDING LOGIC EQUATIONS USING RELAY LOGIC EDITOR

To add logic equations to a project, select the **Relay Logic** button, or the **Relay Logic** option from **Edit** menu, or use the **Ctrl G** shortcut.

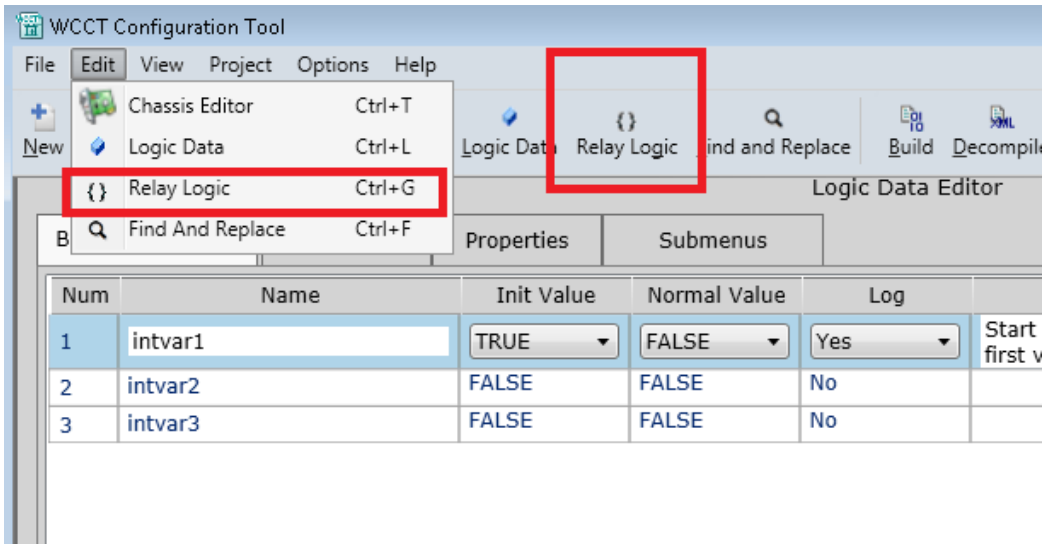


Figure 4-47 Relay Logic Editor - Selecting Project

When there are no equations in a project, the Relay Logic editor will appear as illustrated in the graphic below.

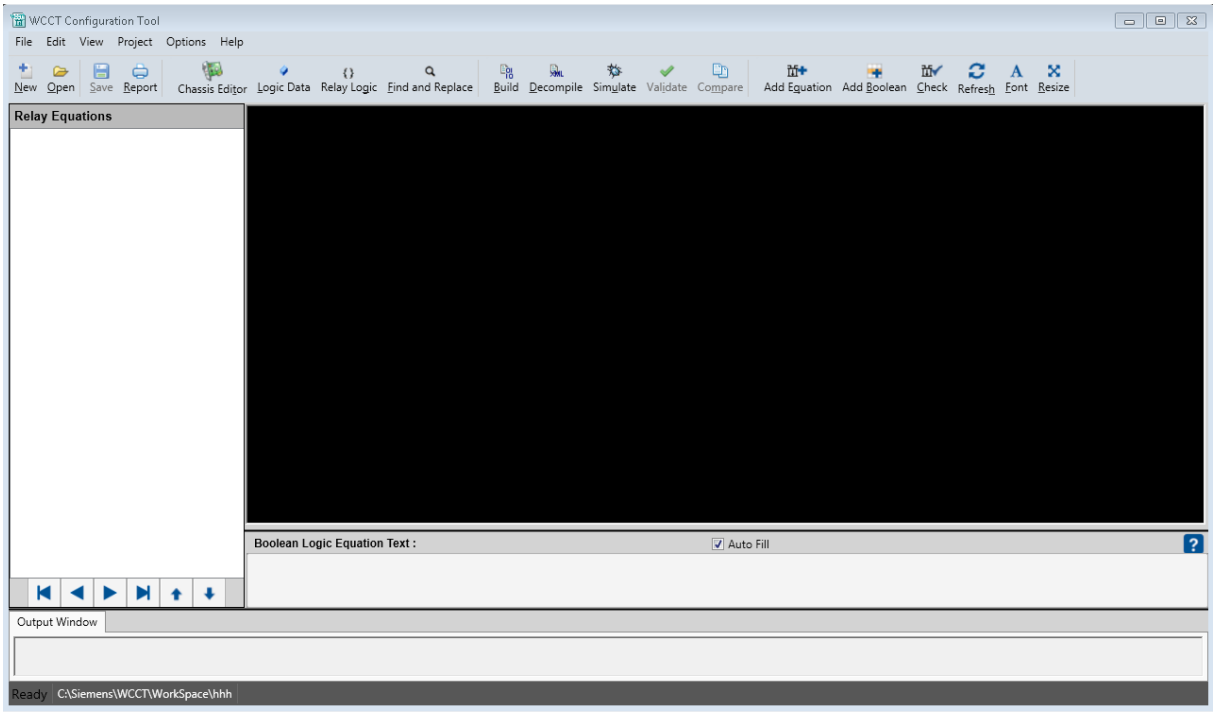


Figure 4-48 Relay Logic Editor Screen - No Equations

To enter an equation, select the **Add Equation** button or use the short cut **Ctrl + ALT + Q**. Type the equation in Boolean format in the Boolean Logic Equation Text box, and press save (Ctrl S) and the equation will appear in the left box and the relay logic equivalent in the top box; for example:

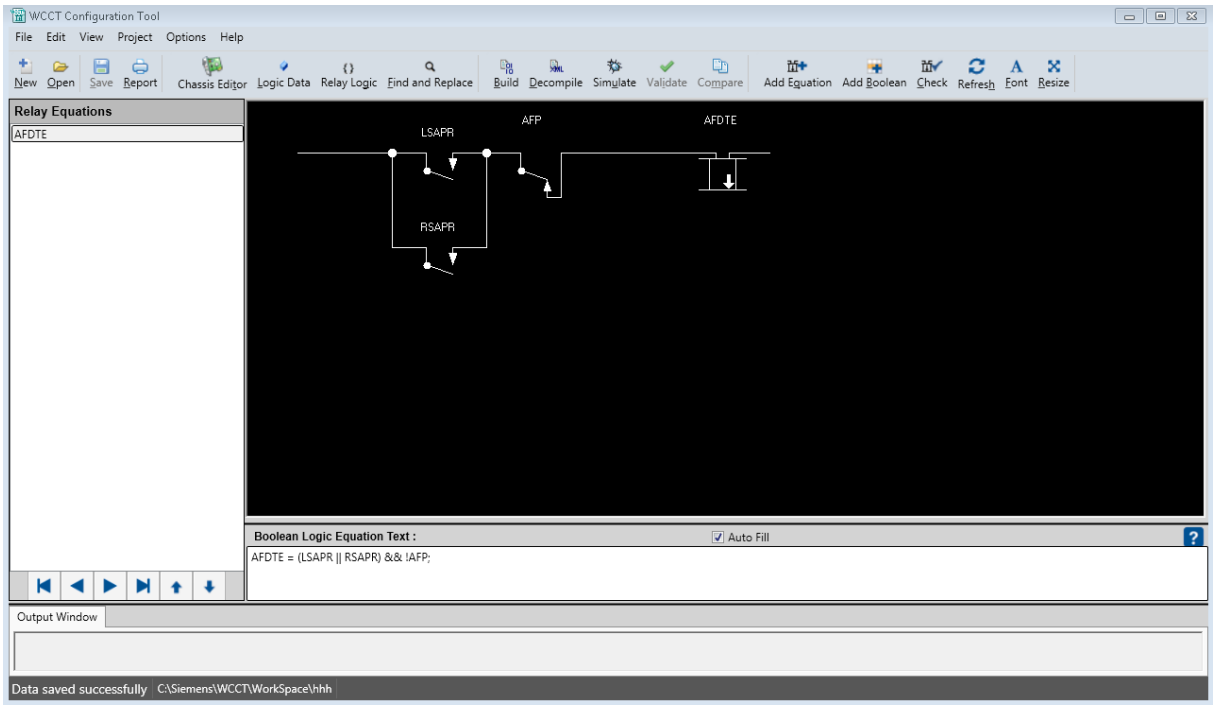


Figure 4-49 Relay Logic Editor - Equation Present

Use the following symbols in the equations:

&& AND
|| OR
! Not
= Assign
( )

To start a timer, use the following:

StartTimerCond ({expression}, {timername})

Where {expression} is a variable or a Boolean expression and {timername} is a timer that was created in the Logic Data Editor timer tab.

To check if a timer has expired use: **TimerExpired** ({timername}).

Select the **?** in the lower right corner to bring down a help box which shows the syntax.

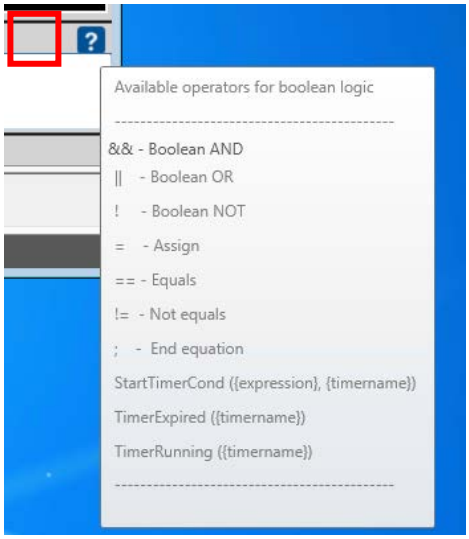


Figure 4-50 Bringing Up a Help Box

To add a new equation to the end of the list, select the **Add Equation** icon or use the short cut **Ctrl + Alt + Q**.

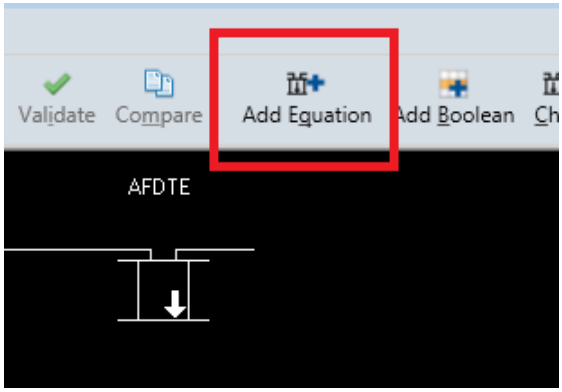


Figure 4-51 Adding a New Equation

To insert an equation before or after an existing one, right-click on the logic state name on the left and select **Insert Before** or **Insert After**. Carriage returns and white space can be used in the text to make it more readable. Comments can be added in the text box preceding the equation by using a pound sign (#) to start the comment. If the relay equation becomes too large, scroll bars will appear so you can see the entire equation.

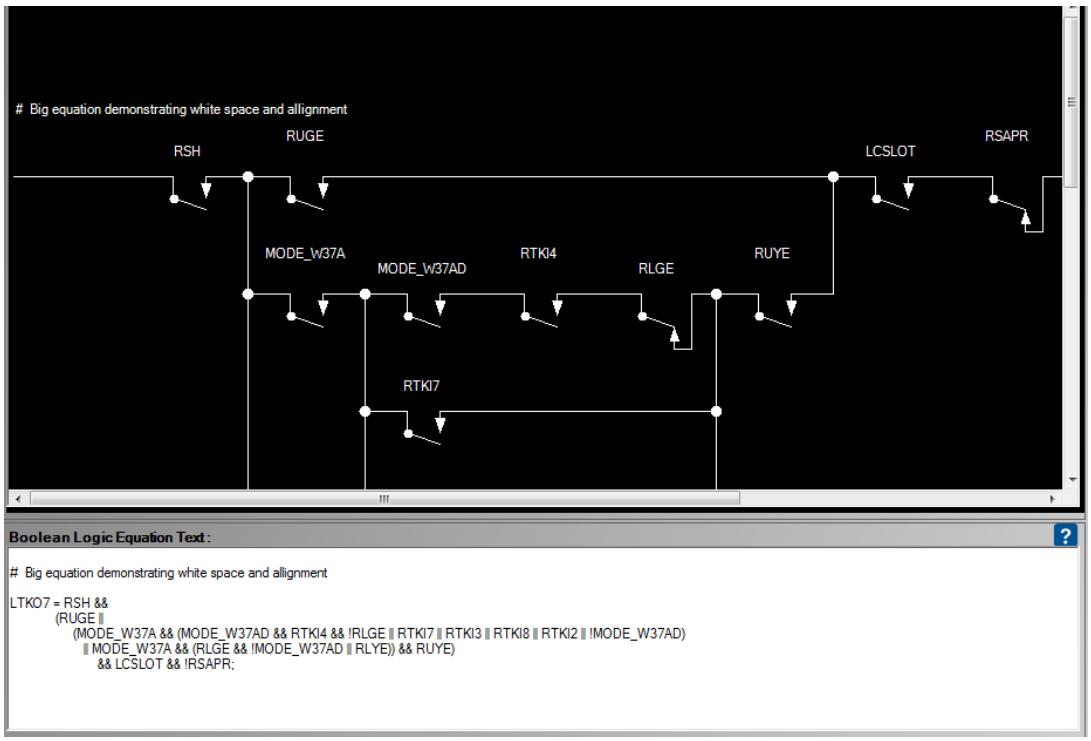


Figure 4-52 Inserting an Equation - Before or After Existing Equation

Continue to enter new equations. Note: equations can be cut and pasted from another source into the Equation text box using standard Ctrl C (copy), Ctrl V (paste). Multiple equations can be copied into the box at one time.

As the list of equations grows, you may want to change the order of equations, or insert a new equation between existing ones. To do this, right-click on the relay coil name list on the left to get the drop down menu shown in the following figure.

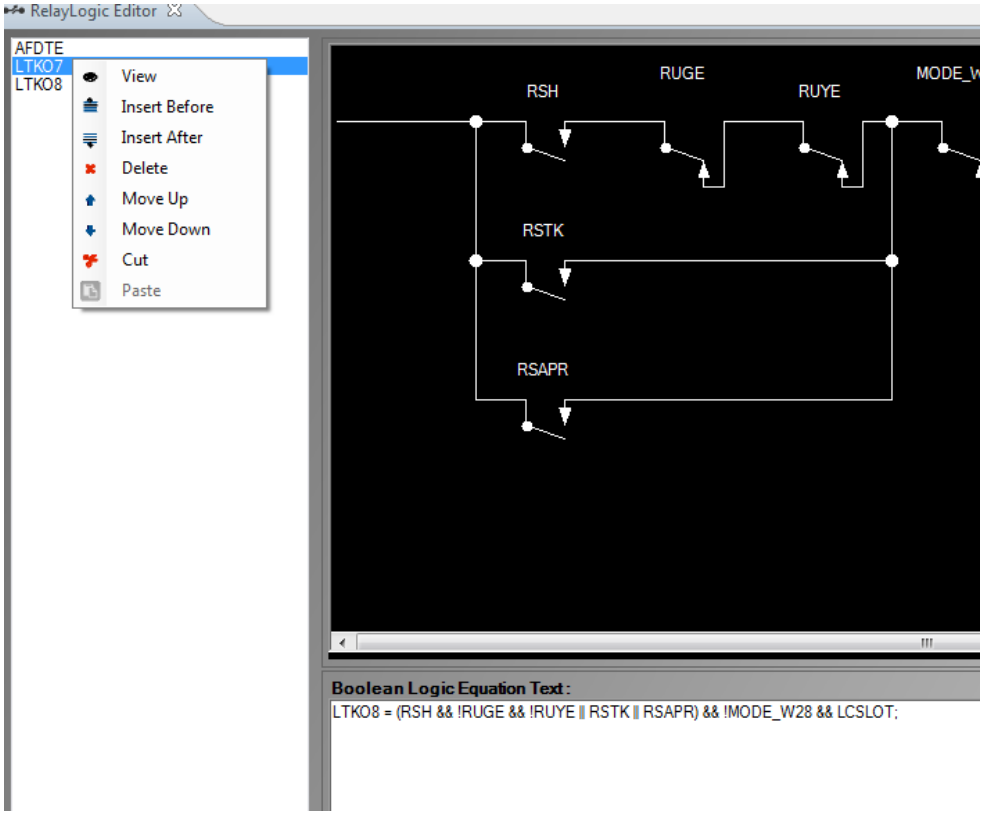


Figure 4-53 Changing the Order of Equations

To view an equation, double-click it in the list on the left, or right-click and select **View**. Once the project has many equations, (see next figure), it is helpful to use the **Find** function to locate where logic states are used.

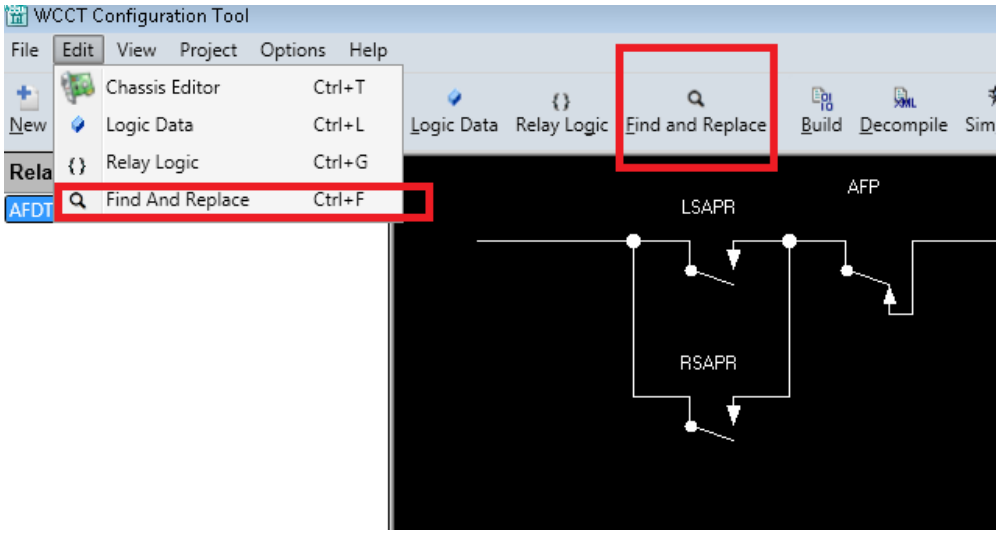


Figure 4-54 Using the Find Function

Select the **Find and Replace** icon or use **Ctrl F**. This will bring up the **Find and Replace** window shown in the following figure.

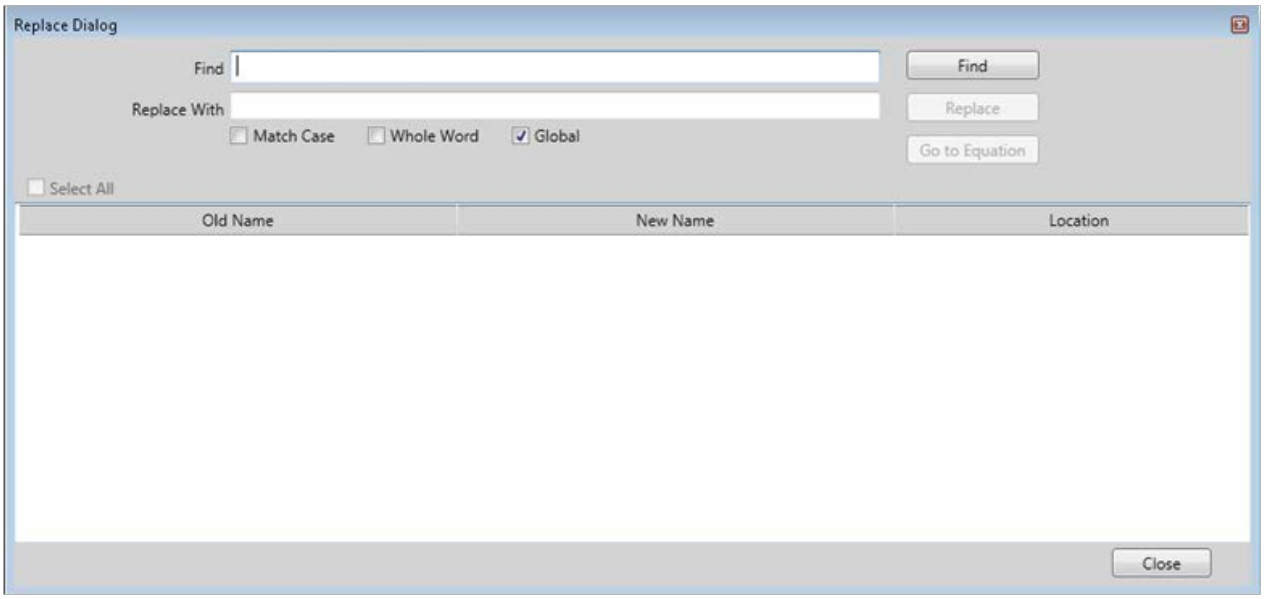


Figure 4-55 Find Equation

Type all or part of the logic state name in and click **Find.**

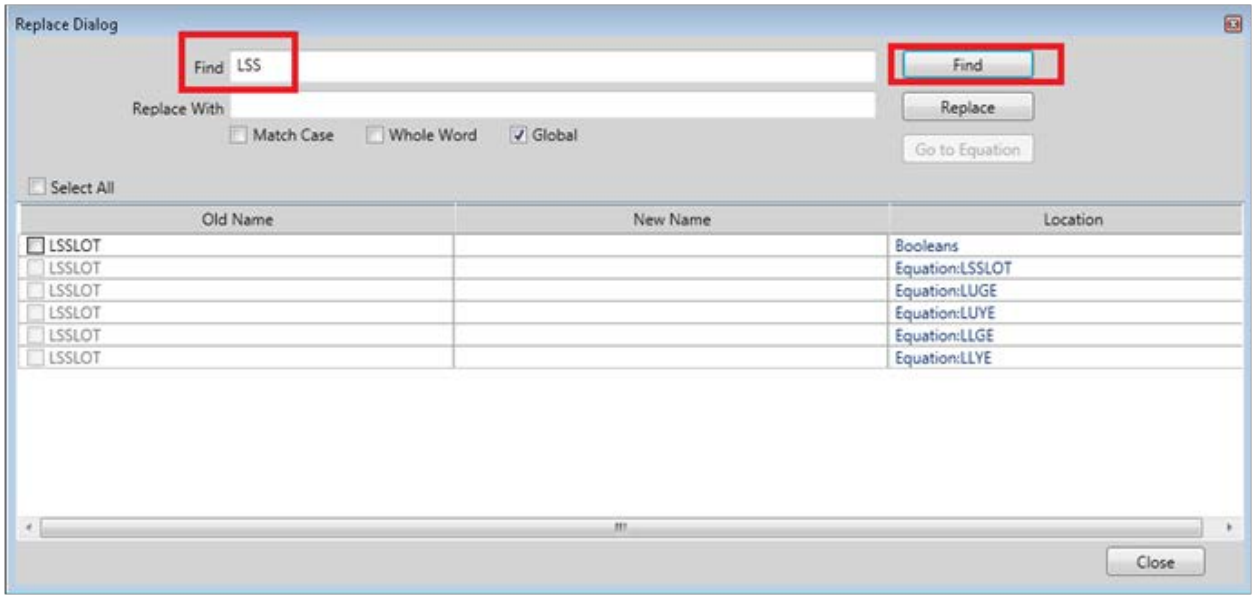


Figure 4-56 Find Name

The window will show all the matching names in the Old Name column, the New Name column will be blank initially.

The Location column will show all the places where the matching name is found:

- Booleans indicates it's found in the Boolean tab of the Logic Data editor
- Equations: LSSLOT, for example, indicates it is found in the equation for LSSLOT
- Slot 2, for example, indicates it is found in module I/O for slot 2

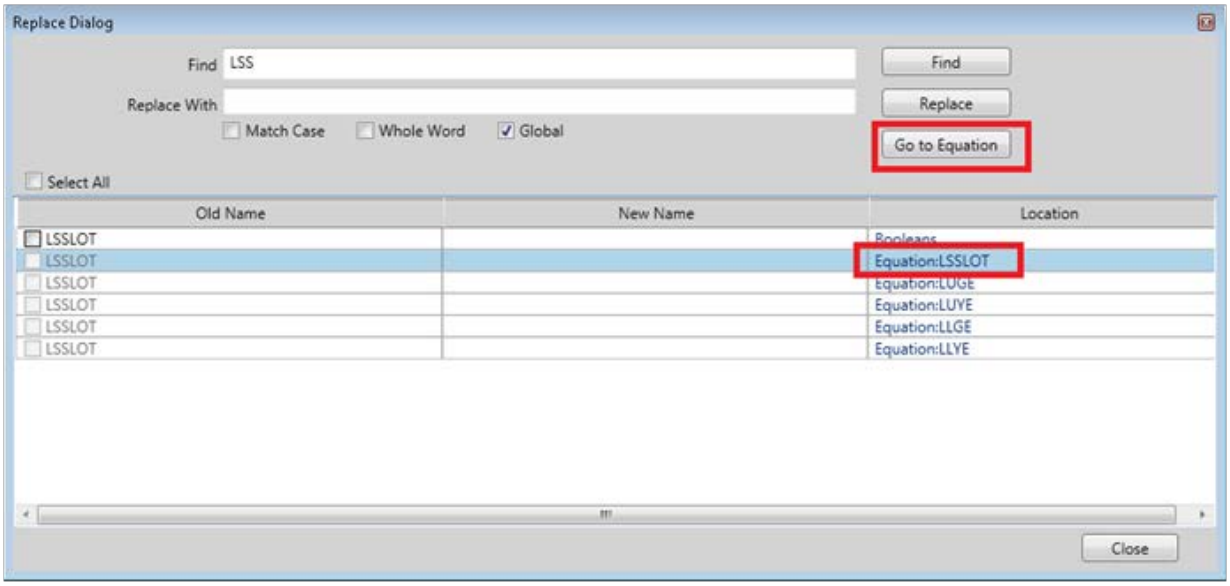


Figure 4-57 Go to Equation

If you double-click on the required equation or select it and press **Go to Equation**, the WCCT will open the equation in the editor.

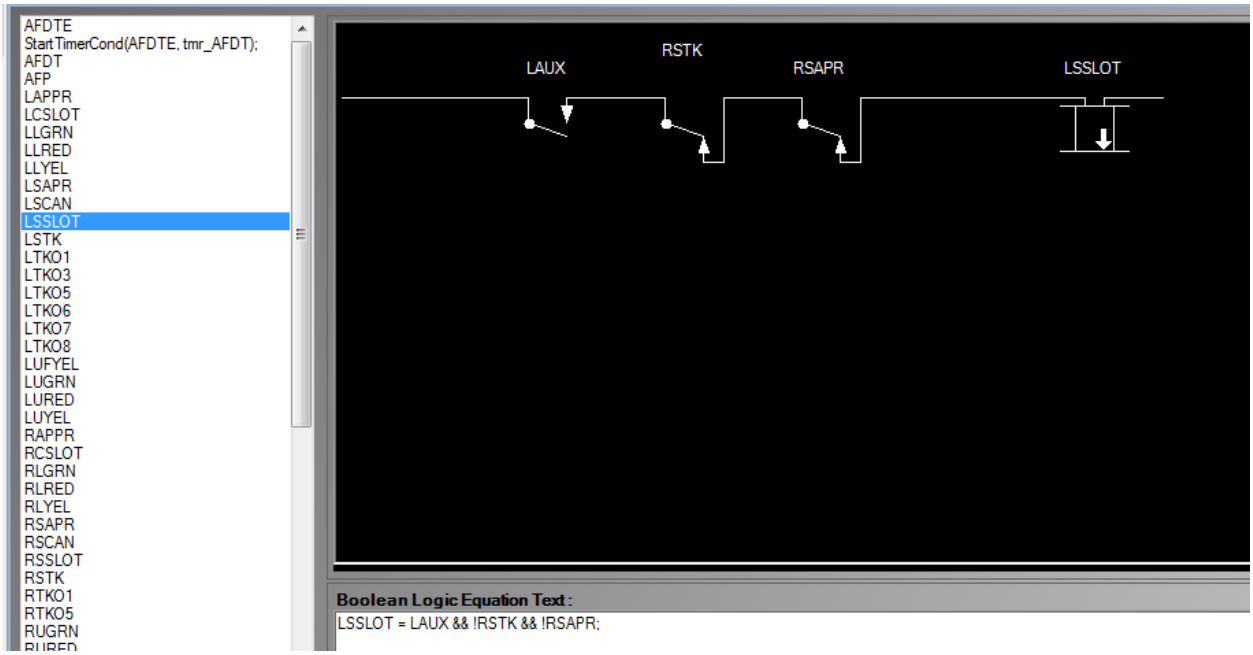


Figure 4-58 Relay Editor Display

See the Relay Logic Editor section for more details.

SECTION 5 SIMULATION

5.0 SIMULATION

The WCCT provides a relay logic based simulation that allows the user to simulate their Boolean/Relay logic.

NOTE

NOTE

The simulation only reproduces the Boolean logic and does not simulate built in functions of the I/O modules. See section 5.8 for features of the I/O modules that may want to be considered while simulating or writing logic.

NOTE

NOTE

The simulation is performed using the logic equations in the MCF, but the equations displayed are the ones in the current Relay Logic view. This means that the project has to be compiled before it can be simulated. If the source files have changed and are newer than the MCF, when the user starts the simulation the WCCT will give a warning indicating that the project files have been modified. At this point the user can choose to continue the simulation or stop and recompile. If the user continues the simulation the results can be unpredictable if the current Boolean equations don't match those in the compiled MCF.

To start the simulation, select the **Project>Simulate** option from the menu, the **Simulate** icon, or the **Ctrl U** shortcut.

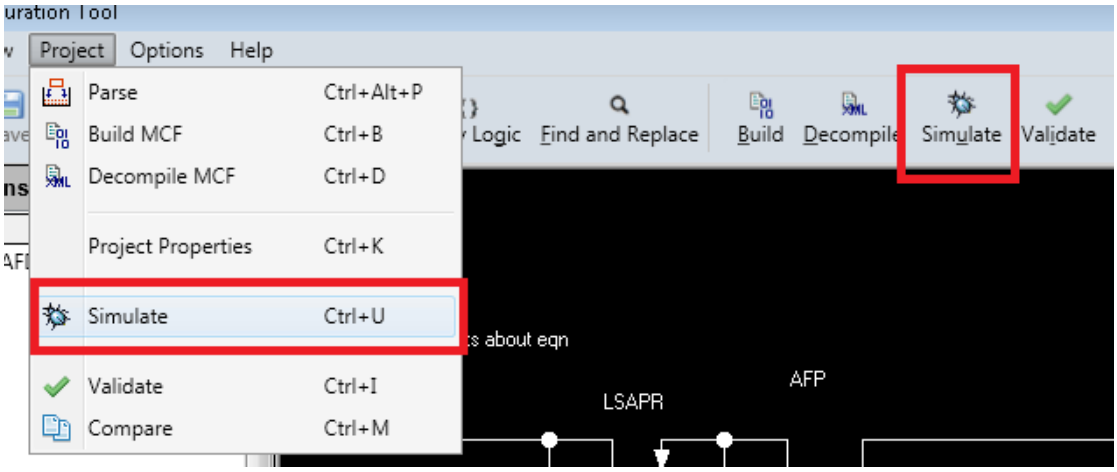


Figure 5-1 Start Simulation Windows

Starting the simulation will bring up the simulation window as shown in the following figure.

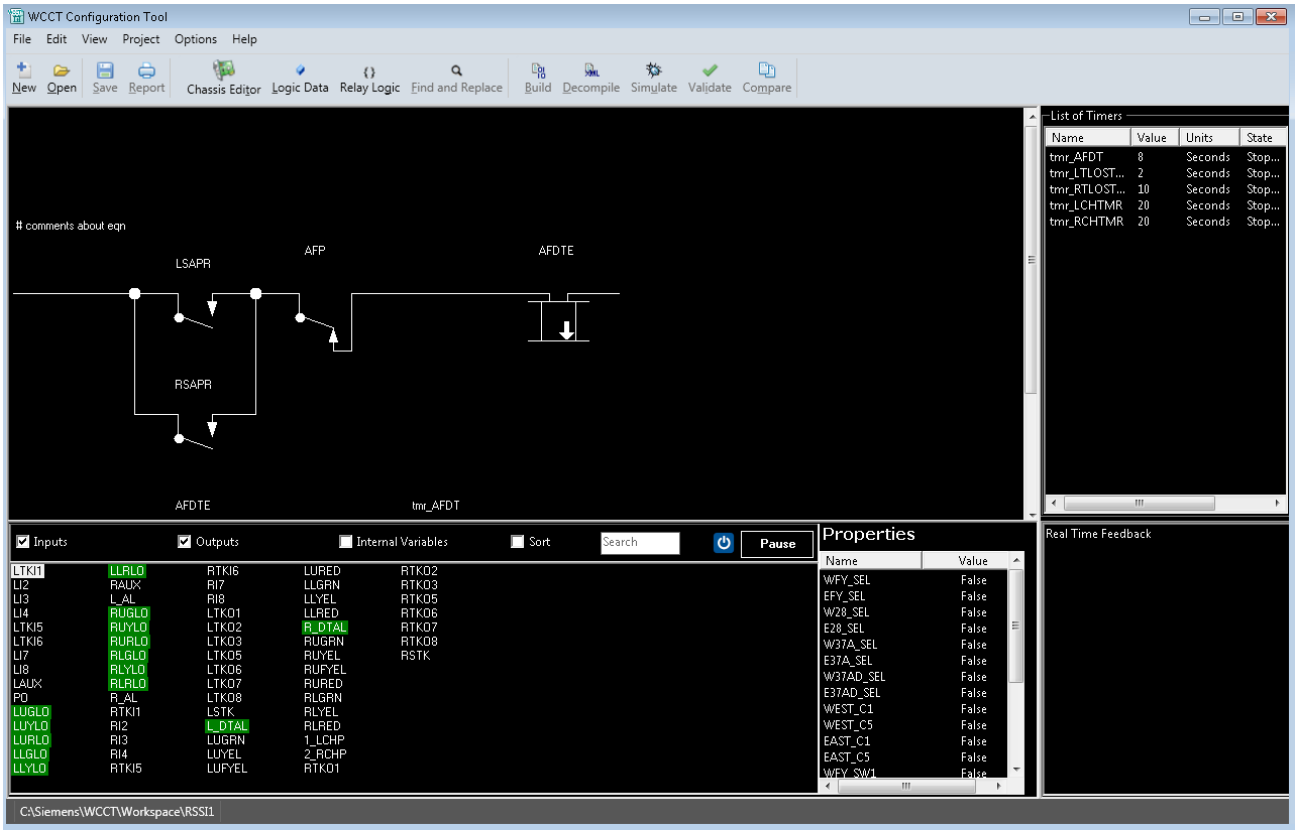


Figure 5-2 Simulation Opening Window

The simulation window is divided into the following sub windows:

- Logic state list Window:
- Property Window
- Timer Window
- Real Time Feedback Window
- Relay Equation Window

5.1 LOGIC STATE LIST WINDOW

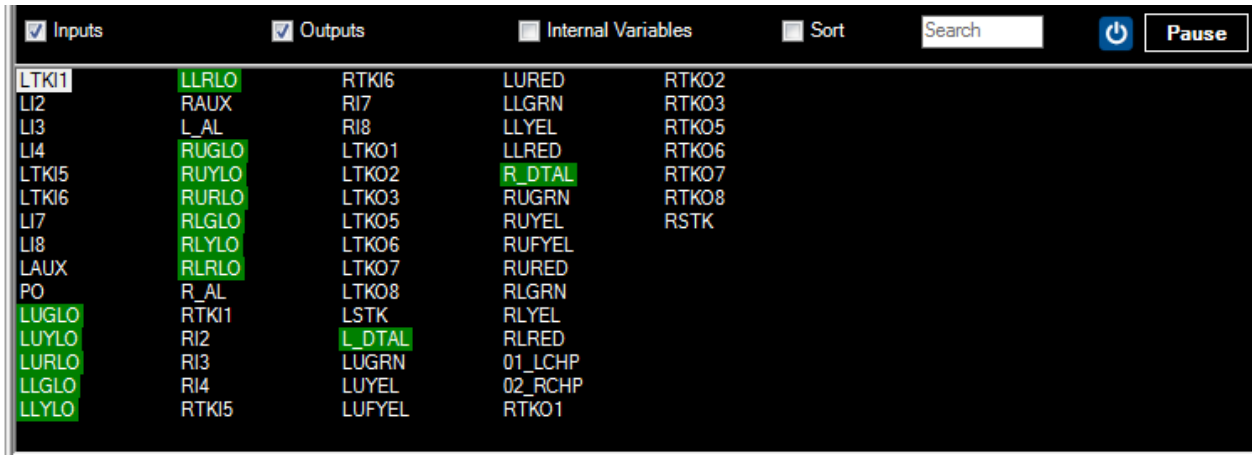


Figure 5-3 Logic State List Window

This shows the list of logic states used in the project. The inputs/outputs/internal variables check boxes allow you to view selected states.

Names shown with a green background represent energized variables.

Names shown without a green background represent de-energized variables.

Selecting the **Sort** check box will list the variables alphabetically.

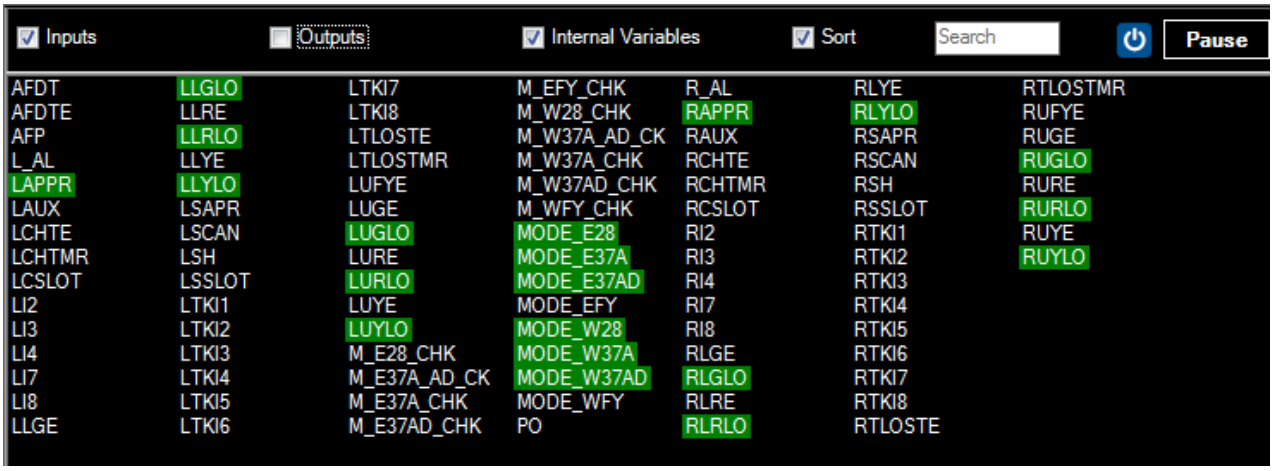


Figure 5-4 Logic State Window – Selection Options

If the **Sort** check box is not checked, the variables are listed in the order they appear in the Logic Data editor. To quickly find a variable without reading through the list, type all or part of the name in the search box.

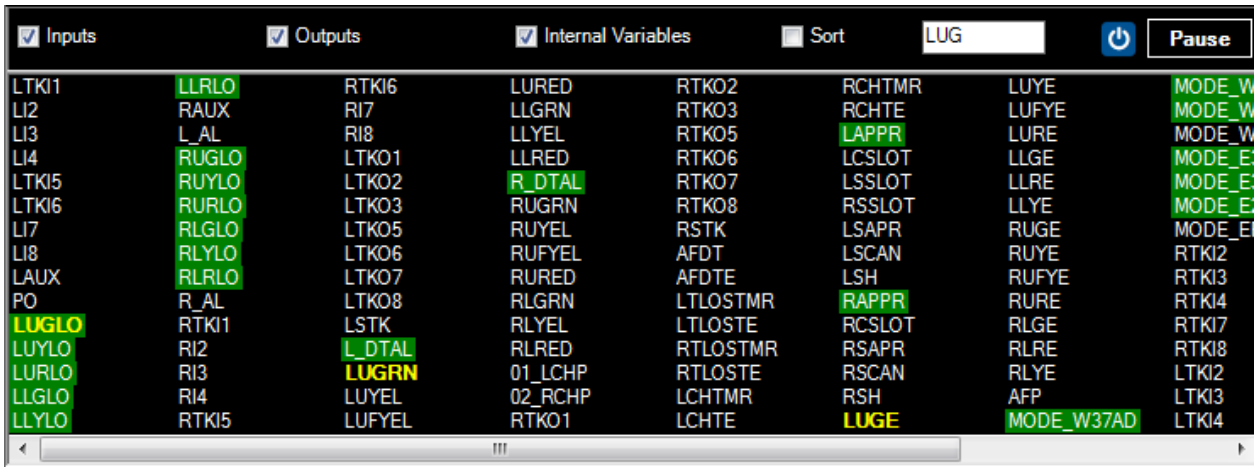


Figure 5-5 Sort Box Not Checked

Right-clicking on a variable will show a drop-down menu with the following options:

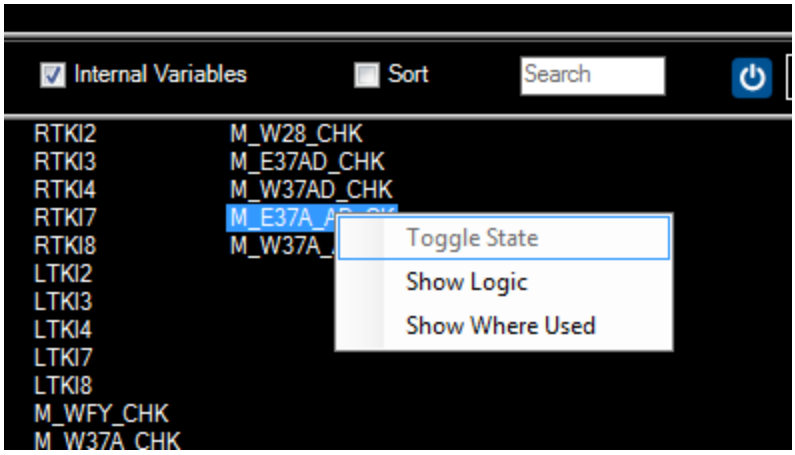


Figure 5-6 Variables Drop-down Menu

If the variable is an input, the following are enabled:

- Toggle state
- Show Where Used

If the variable is an output, the following is enabled:

- Show Logic
- Show Where Used

If the variable is an internal state, the following are enabled:

- Show Where Used
- Show Logic

Selecting **Toggle Input** will change the input state to the opposite value.

Selecting **Show Logic** will add the relay equation that defines this logic state to the Relay Equation window.

Selecting **Show Where Used** will add all the relay equations where this logic state is used to the Relay Equation window.

Note: The above actions result in new equations being appended to the existing ones shown in the Relay Equation window. To clear the existing equations that are shown, right-click in the Relay Equation window and select either the option to **Clear Equation** or **Clear Display**.

For example, **Show Where Used** for LAUX displays the following:

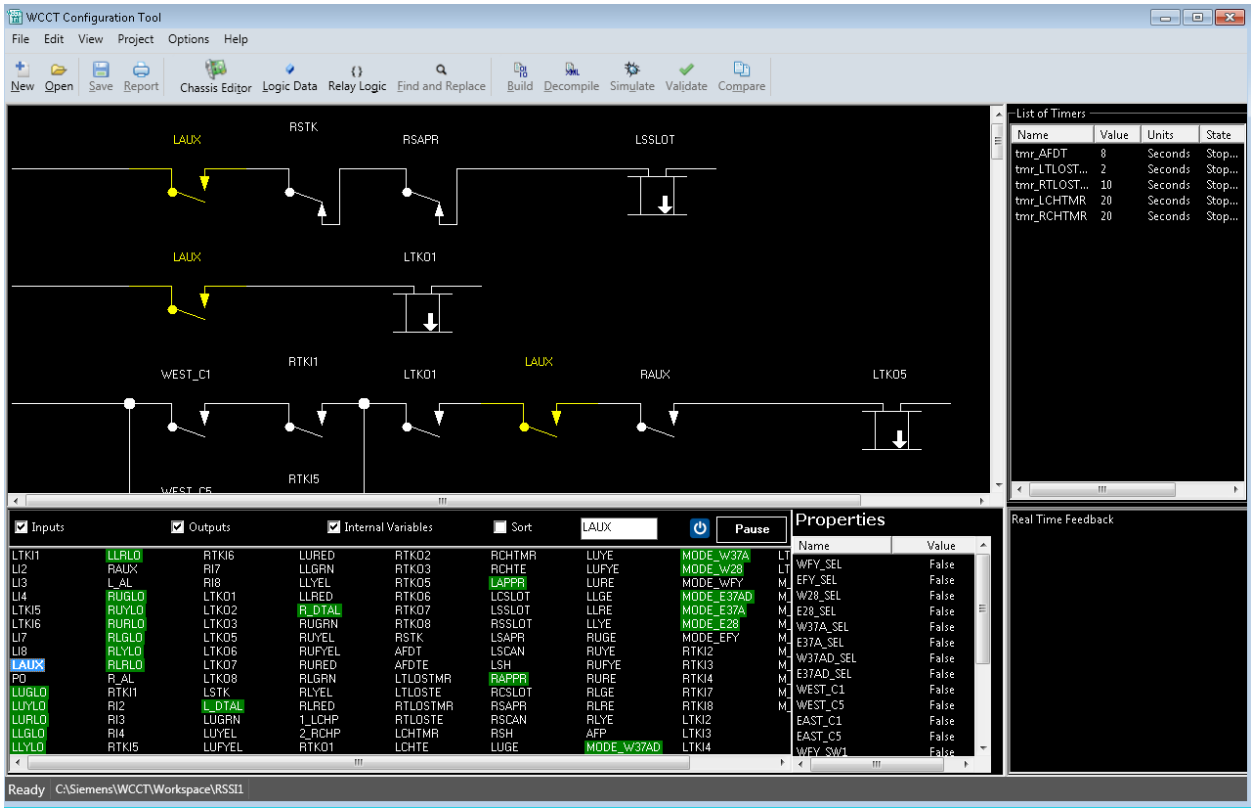


Figure 5-7 Show Logic Window

To reset the simulation back to its initial state, press the blue **Reset Simulator** button.

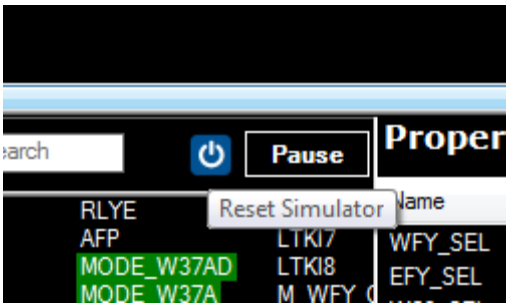


Figure 5-8 Reset Simulator Button

The normal mode of the simulator is to run the logic equations on change of state of an input (or timer or property) until they are stable, then wait for the next change.

In some cases, it may be necessary to change multiple inputs at the same time and then have the changes run through the logic together. For example, this may be needed in order to simulate how a coded track card changes from receiving one vital code to another, without dropping the vital code, or having two vital code bits present, e.g. a transition from receiving code 1,7 to receiving code 1,4.

In order to do this in the simulator, press the pause button.

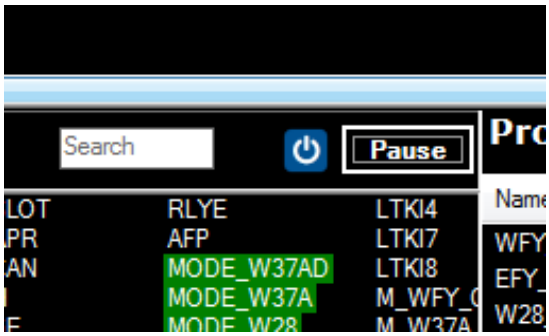


Figure 5-9 Simulator Pause Button

This will change to show **Resume** when the simulator is paused.

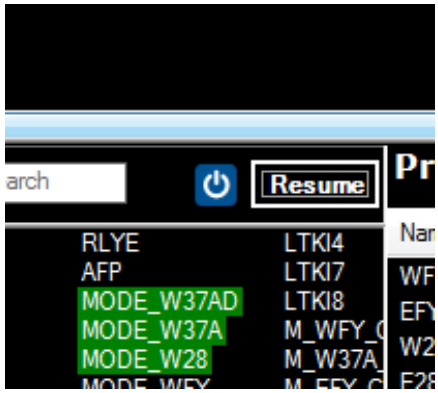


Figure 5-10 Simulator Resume Button

At that point, multiple inputs can be changed in this window and the logic won't be executed until **Resume** is pressed.

Note that:

a) Currently, changing the state of an input from the relay diagram is not supported under pause mode.
b) If a property is changed, the logic will be executed even when paused.
c) If a timer is expired, the logic will be executed even when paused.

5.2 PROPERTIES WINDOW

The Properties window allows the values of configuration properties to be changed in the simulation.

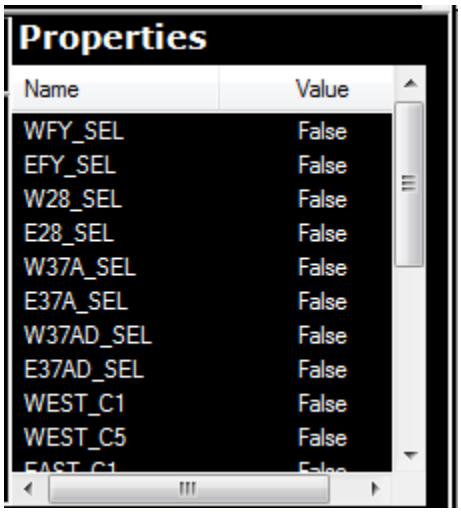


Figure 5-11 Properties Window

Double-click on the true/false values to change them or right-click and select **Toggle State**.

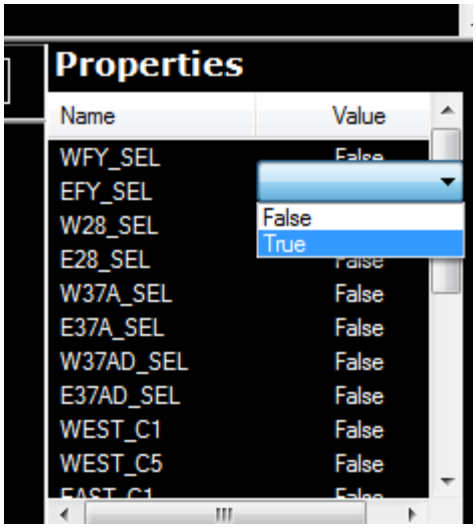


Figure 5-12 Properties Window - Selecting Values - True/ False

Right-click on a property to bring up the menu option to show where this property is used.

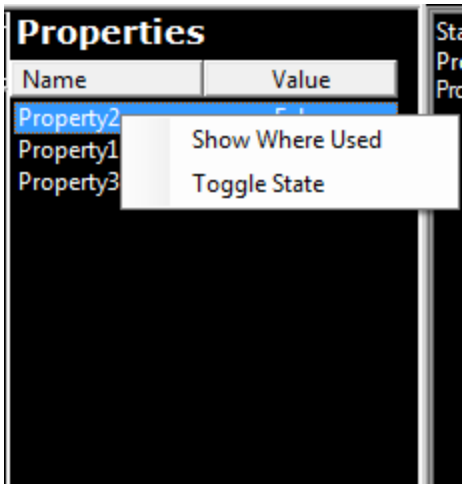


Figure 5-13 Properties Menu Options

If this is selected, the relay diagram will show all equations that use this property (as shown below).

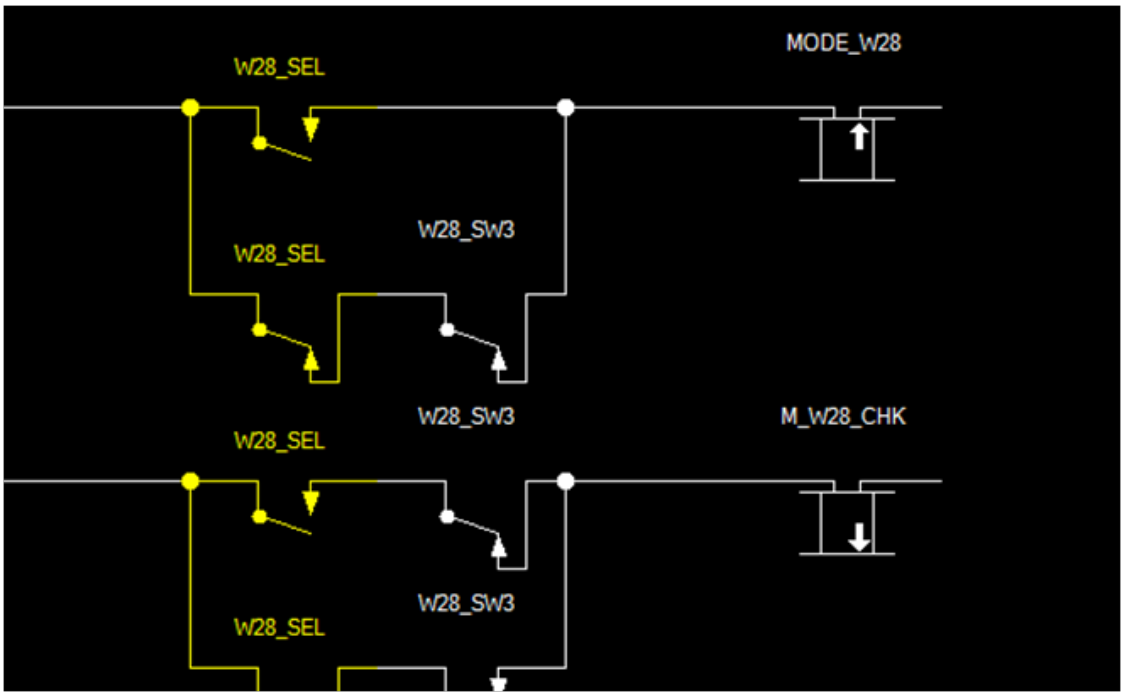


Figure 5-14 Relay Diagram Window - All Equations Used in Selected Property

5.3 LIST OF TIMERS WINDOW

The **List of Timers** window shows the timers used in the program and the following specifications:

- Timer Name
- Timer Value
- Timer Units
- State of Timer (stopped, running, or expired)
- Time to Run on Timer in Seconds (scroll right to see this)

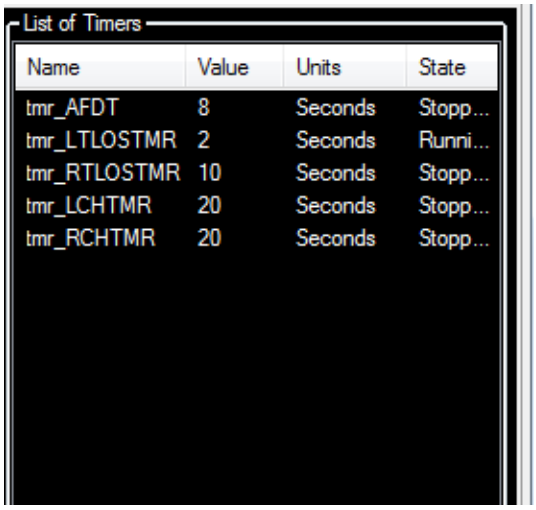


Figure 5-15 List of Timers Window

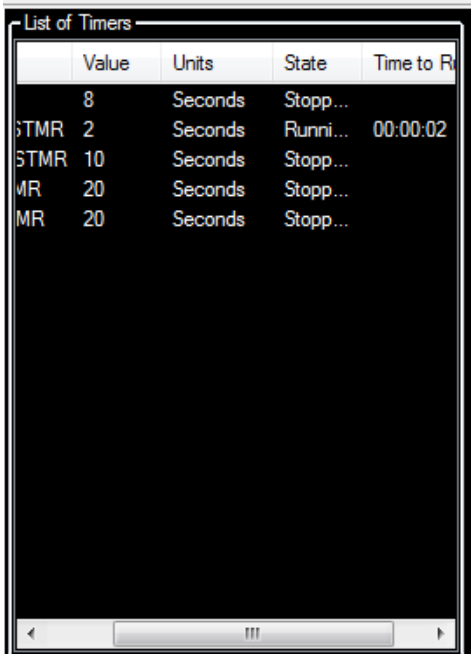


Figure 5-16 List of Timers - Runtime Status

The value of the timer can be changed by double-clicking on the value column and entering a new value. The timer value cannot be changed when the timer is running, only when it is stopped or expired.

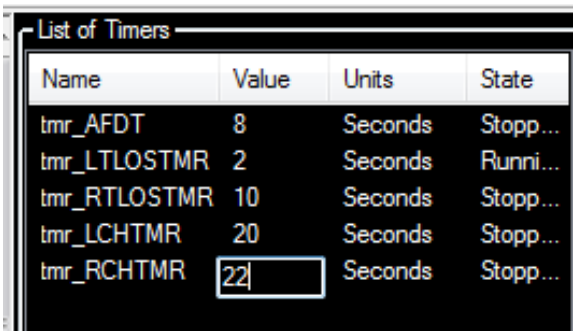


Figure 5-17 Changing Timer Values

If the timer is in milliseconds, the WCCT will enforce that you enter it into 100 ms increments. Right-click on a timer to bring up the options menu which contains:

- Expire Timer – sets a running timer to expired (only visible when timer is running)
- Show Logic – shows where timer coil is set
- Show Where Used – show all equations timer contact is used in

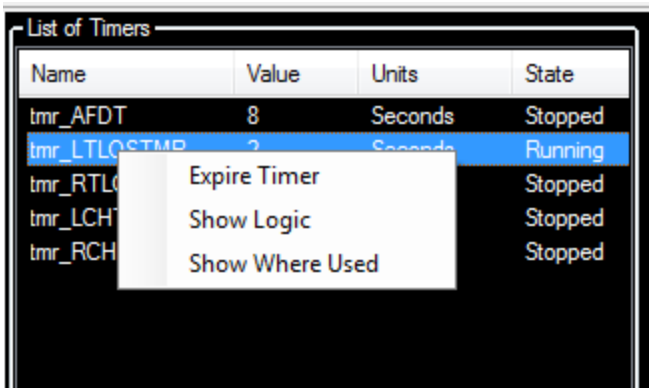


Figure 5-18 Timer Menu Options

5.4 EXPIRING TIMERS

The simulation does not run real time. When an input changes, the logic executes until stable, then stops. When a timer is started, its time to run will remain at its configured value until a timer is expired. This method allows the user to debug the logic more easily, as timers are not expiring in real time while the user is trying to look at things.

When the user right-clicks on a running timer and selects the **Expire Timer** option, the simulation clock is incremented by the amount of time remaining on that timer + 0.1s and this timer will change to **Expired**.

If multiple timers are running, and the user selects one timer and expires it, the time to run for all the running timers is decremented by the time to run for the user-selected timer, so any timers with a shorter time to run will expire as well. This will ensure timers expire in their natural order.

For example, if there are three timers, all which started at the same time with the following values:

	Value	**State**	**Time to Run**
Timer A	1	Running	1:00
Timer B	5	Running	5:00
Timer C	10	Running	10:00

If the user expired Timer C, 10s will elapse on the simulation clock so all three timers will be set to **Expired**. If the user selects to expire timer A, 1.1 s will elapse on the simulation clock so the timer list will show:

	Value	**State**	**Time to Run**
Timer A	1	Expired	0:00
Timer B	5	Running	3:90
Timer C	10	Running	8:90

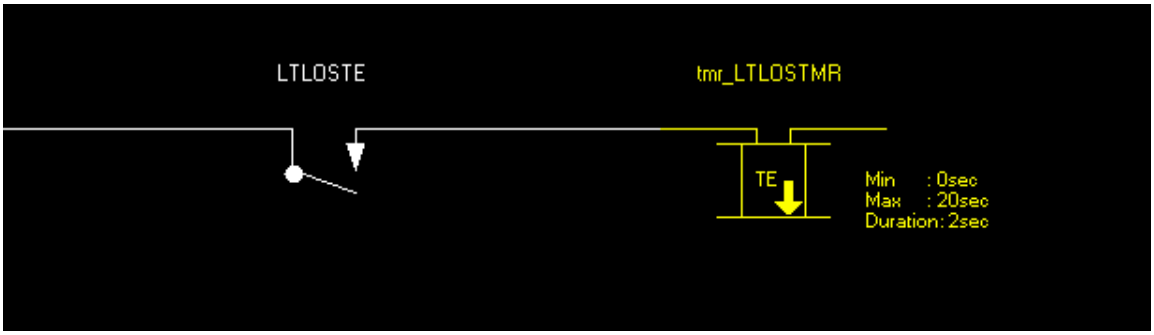
If the user then selects to expire timer B, 3.90 + 0.1 s will elapse on the simulation clock so the timer list will show:

	Value	**State**	**Time to Run**
Timer A	1	Expired	0:00
Timer B	5	Expired	0:00
Timer C	10	Running	4:90

5.5 TIMERS IN RELAY LOGIC

A timer will be show in the logic with its default parameter next to it and a TE in the relay coil. Note: the default value is that from MCF, not the value set by the user during simulation, meaning this value will not match the value shown in the List of timers if the user has changed the value.

When the timer is stopped, it will show a down arrow as follows on the relay coil.



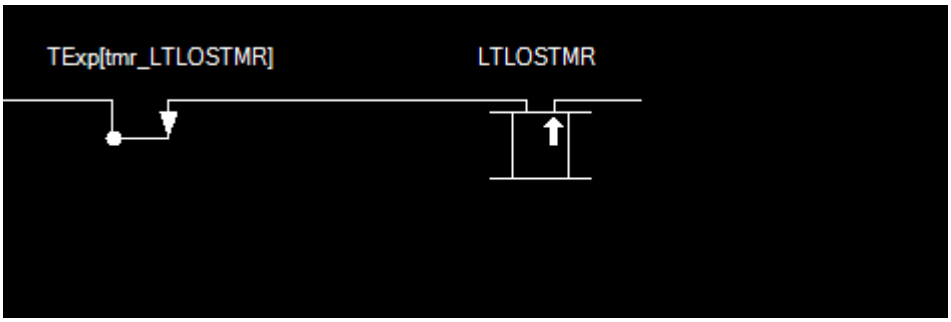
The relay contact is labeled with TExp and then the timer name.

When the timer is running, the timer coil shows an up arrow for energized,

and the relay contact still shows down while the timer is still running.

When the timer goes to expired, the relay coil still shows energized.

And now the timer contact also shows energized.



5.6 RELAY EQUATION VIEW

In this view the user can see the relay logic. Right-click on a relay contact or coil to get a drop-down menu that shows:

- Toggle State (for inputs only)
- Show Logic (for outputs and internal states only)
- Show Where Used
- Clear Equation
- Clear Display

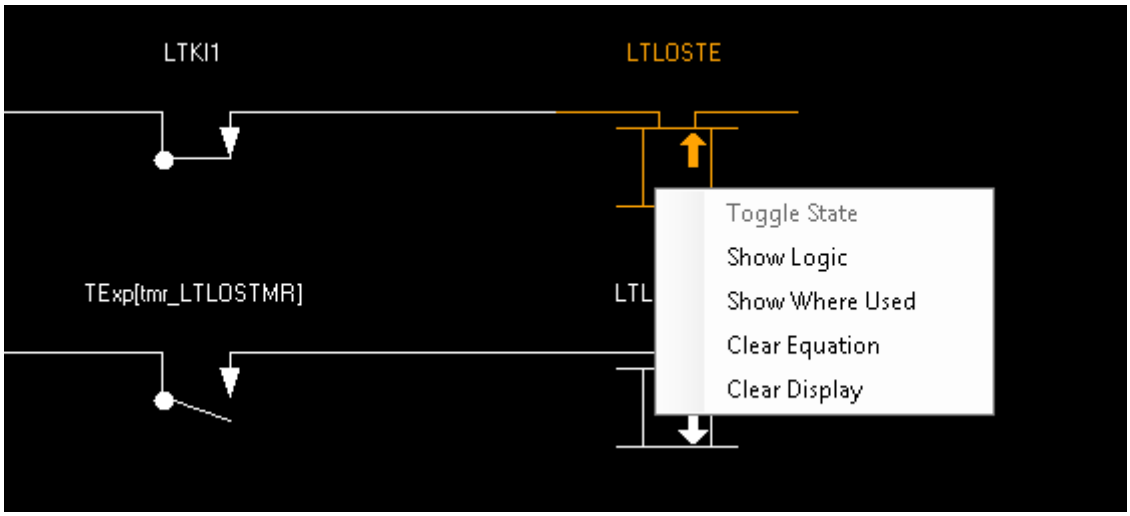


Figure 5-19 Relay Logic View Drop-Down Menu

Selecting toggle state will change the state of the input. Double-clicking on the relay contact will also toggle its state.

Show Logic will add the equation that defines the logic state to the current relay display view.

Show Where Used will add all the equations in which the selected logic state is used to the current relay display view.

Clear Equation will remove the selected equation from the current relay display view.

Clear Display will remove all equations from the current relay display view.

5.7 STATE CHANGES WINDOW

The State Changes window lists the changes in the states of relay contacts, coils, and change of timer states that occur during simulation.

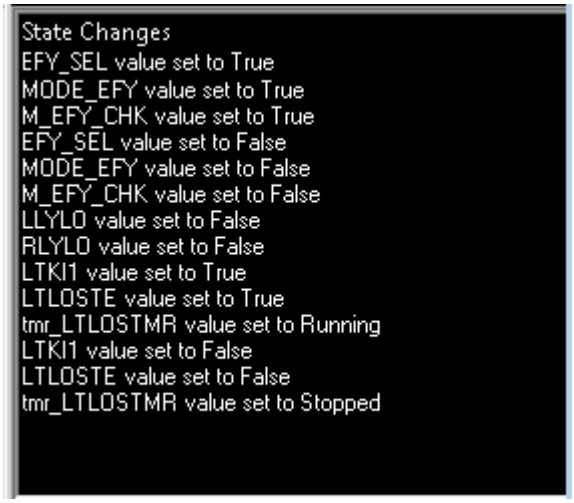


Figure 5-20 State Changes Window

To clear the log entries in the window, right-click in the window and select the **Clear Log** option.

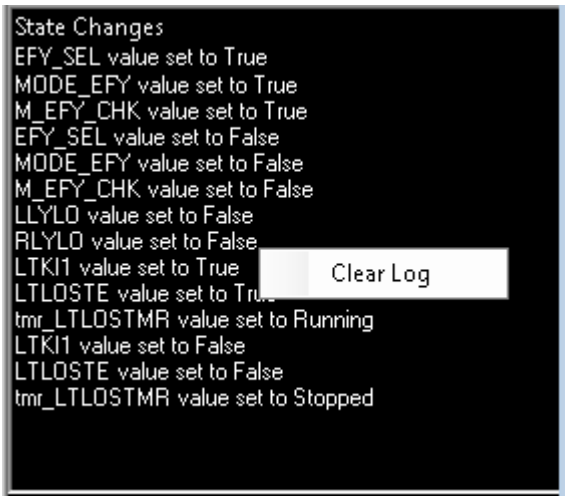


Figure 5-21 Clearing State Changes Window

5.8 I/O MODULE SPECIFIC SIMULATION CONSIDERATIONS

5.8.1 Code Track / Coded Line Module

- When transitioning from receiving one vital code to another, use the **Pause** button to avoid having the vital code bit drop out or two vital code bits being set at once.
- When simulating a track being shunted, first clear the Non-Vital Rx bit and Vital Rx bit, then **Pause** the simulation, clear any vital or non-vital codes and the code 1, then **Resume** the simulation.
- When simulating clearing a track shunt, first set the Non-Vital Rx bit and Vital Rx bits (could **Pause** and set both bits, then **Resume,** or set non-vital Rx first), then **Pause** the simulation, then set any vital or non-vital codes and the code 1, then **Resume** the simulation.

5.8.2 WayTraX Module

- When transitioning from receiving one code to another, use the **Pause** button to avoid having the vital code bit drop out or two vital code bits be set at once.

5.8.3 Colorlight Module

- When foreign energy is detected by a Colorlight card, it will set all its lamps to LOR, so when simulating foreign energy, manually set all the lamps to LOR.
- When foreign energy is detected by a Colorlight card, it will turn off its DC-DC Converter and report the converter status as de-energized, so if using the red-retaining options and simulating foreign energy, also clear the Colorlight Convertor status.

5.8.4 Searchlight Module

See section 3.10.

- When simulating a Searchlight, the mechanism feedback positions need to be manually entered; for example, if the mechanism is set to green, manually set the green feedback bit.

This page intentionally left blank

SECTION 6 CHECKING FILES

6.0 PARSING A PROJECT

In general, the Editors where the data is entered will provide error messages if illegal or inconsistent data is entered. However, there are some checks done when the project is compiled. These checks can be performed without actually doing the compile by selecting **Project >Parse** from the menu.

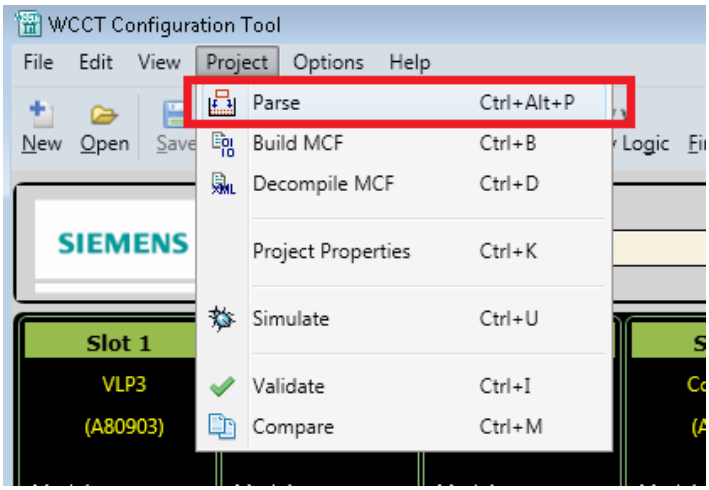


Figure 6-1 Parse Project Menu

The parse errors show up in the output window. Note that the output window can be resized by dragging the top border up or down.

Figure 6-2 File Error Indication on Console Window

The Parse errors gives a reason for the error, e.g. ‘undefined name [x2]’ and refers to a line number in the source file, e.g. test2.gl (62) in the example above.

To find where the undefined name is, use the **Find and Replace** function.

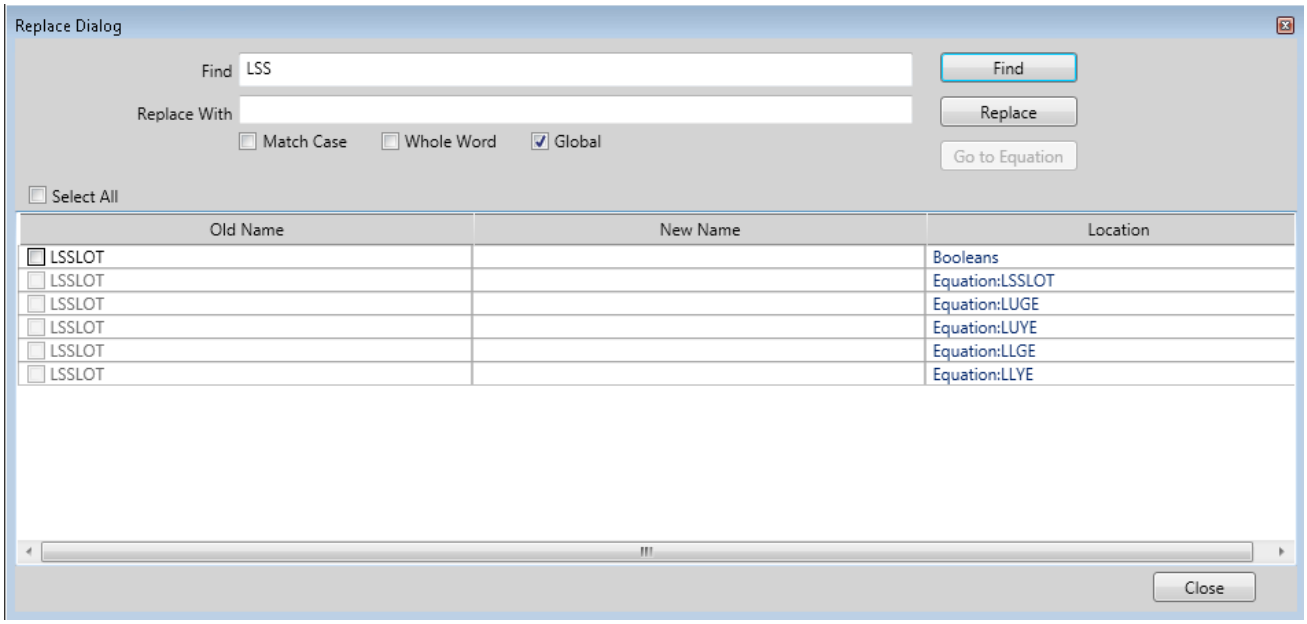


Figure 6-3 Finding Undefined Variable

6.1 BUILDING THE MCF

To build the project MCF, select the **Build** icon, **Project>Build MCF**, or use the short cut **Ctrl B**.

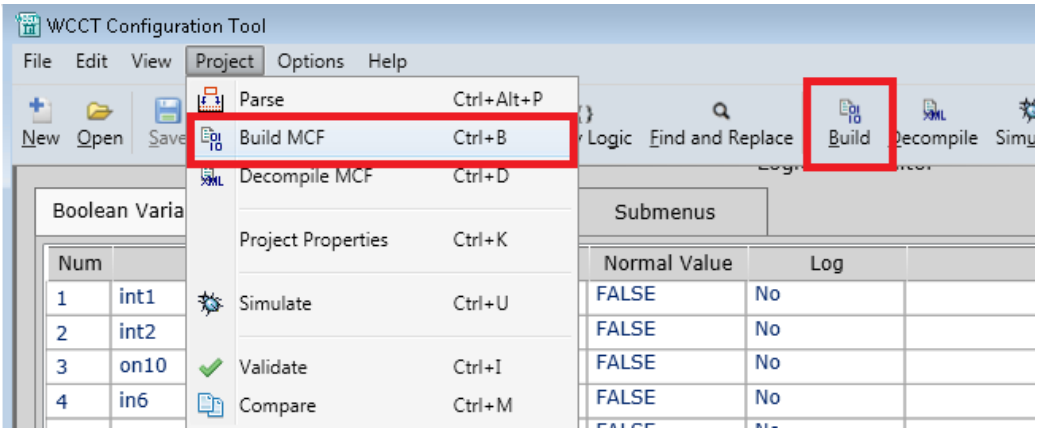


Figure 6-4 Building the Project

As the project is compiling, the output window shows the progress.

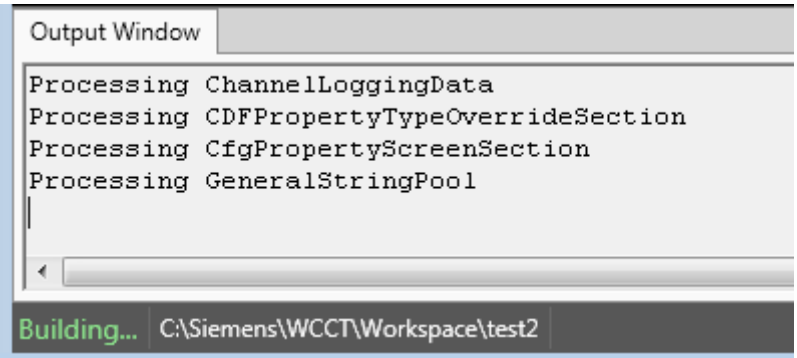


Figure 6-5 Build Progress

When the compilation has finished, you will see the message near the end of the text in the console window:

```
XML To Binary MCF Generation Complete
Done
        1 file(s) copied.
        1 file(s) copied.


{main_unit}: Number of used logic states 8008 (maximum 50000)
```

The MCF file will be created in the MCF_Dir subdirectory of the project workspace.

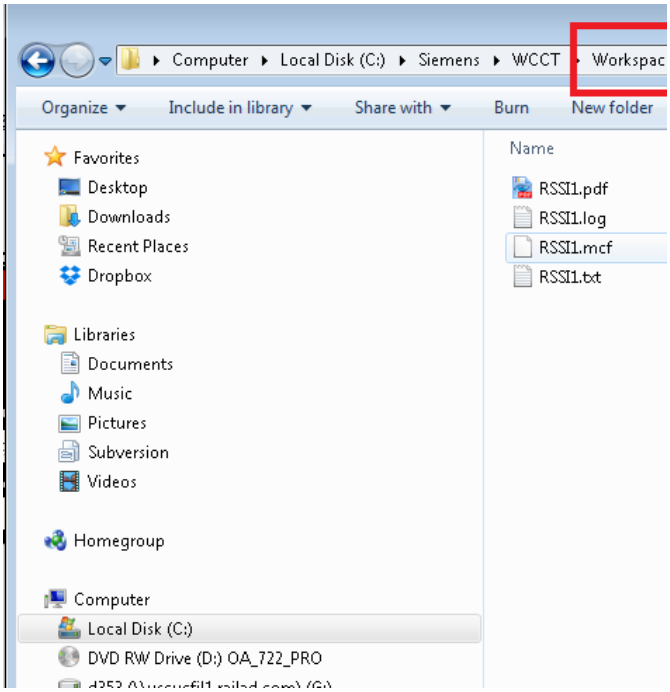


Figure 6-6 MCF Directory Window

A Log file is created in MCF_Dir folder, this is a text file and contains the CRC of the MCF, for example:

MCF CRC : 0x9CAB5836

This is the MCF CRC that is used to verify that the MCF is not corrupt and is the one the user loads into the WayConneX/GEO.

The MCF file and reports are named after the project and may or may not have the 3 digit version number appended to it, depending on how the user has set **Include MCF File Version** in the **Options>Settings** menu.

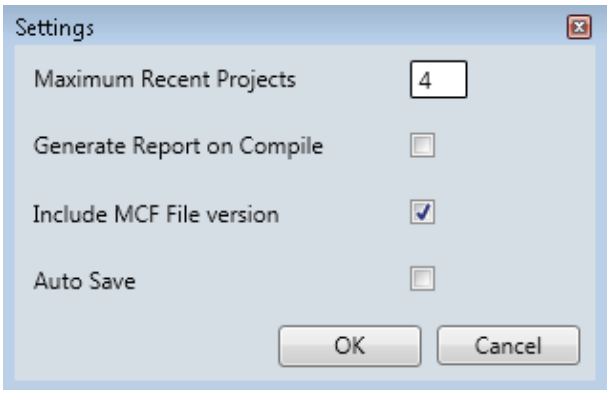


Figure 6-7 Options/Settings Window

The MCF version is set in the **Project>Properties** window.

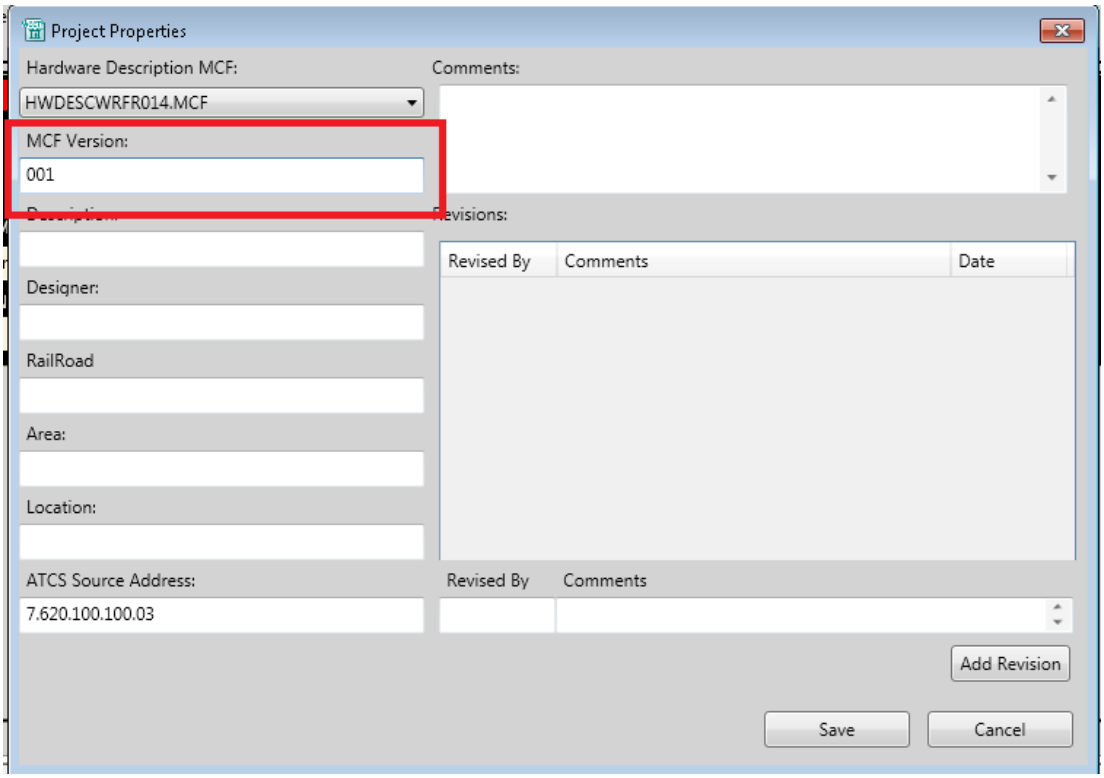


Figure 6-8 Project Properties - MCF Version

The build process also creates a text file (.txt) containing the program listing.

The tool may be configured so that the build process creates a PDF report containing the program listing and the relay logic diagrams. To turn this on, go to the **Options>Settings** window (see previous Options/Settings window) and check the **Generate Report on Compile** check box.

When the build process is started, the WCCT first performs the Parse check described in the previous section. It then performs some pre-compilation checks. The compilation may fail for other reasons, for example a module has been added, but is not used, or a limit has been exceeded. These errors are not detected until the final compilation stages, and may result in somewhat cryptic error messages. Appendix A lists the more common error messages.

6.2 DECOMPILING

The MCF file contains all the information inside it so that it can be decompiled back to its original source files. To decompile an MCF select the **Project>Decompile MCF** from the menu, the **Decompile** icon from the toolbar, or use the **Ctrl D** shortcut.

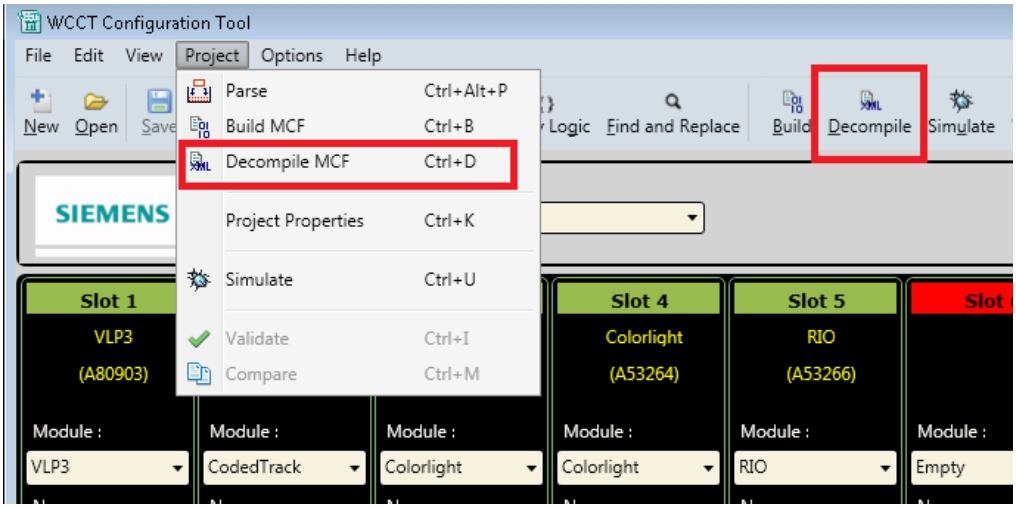


Figure 6-9 Decompile MCF Option

This brings up a browse window, from here select the file path of the MCF.

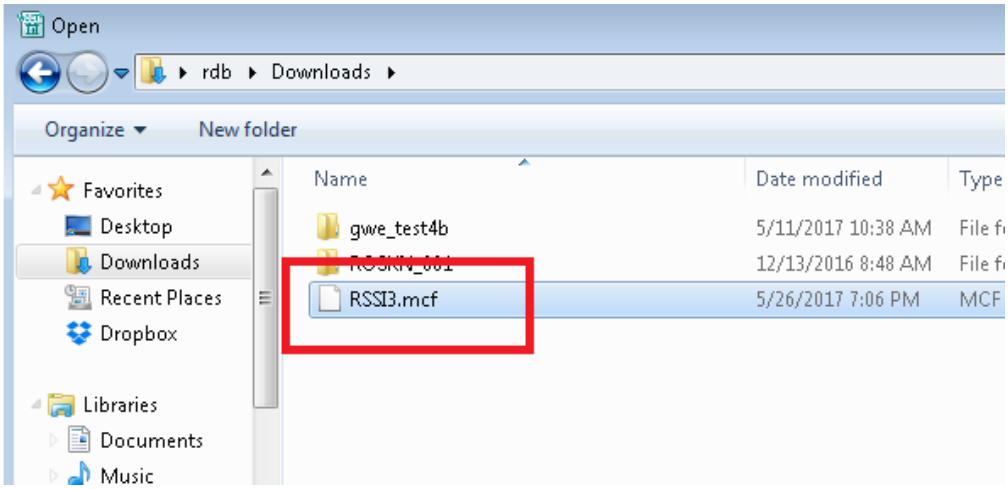


Figure 6-10 Selecting the MCF to Decompile

The MCF will be decompiled into a folder under the default Workspace and given the name {projectname}_Decomp001

The project will be left open after it is decompiled.

6.3 VALIDATING

A key part of creating an MCF is to validate it. The validation process performs the following functions:

- decompiles the MCF back to its source files
- compares the source files with the original source files
- recompiles the decompiled project
- compares the newly compiled MCF with the original MCF
- creates validation report containing
    - the difference between the original and decompiled source files
    - the binary differences the between the original and newly compiled MCF

To validate a project select **Validate** from the Project menu, select the **Validate** icon, or use the **Ctr I** shortcut.

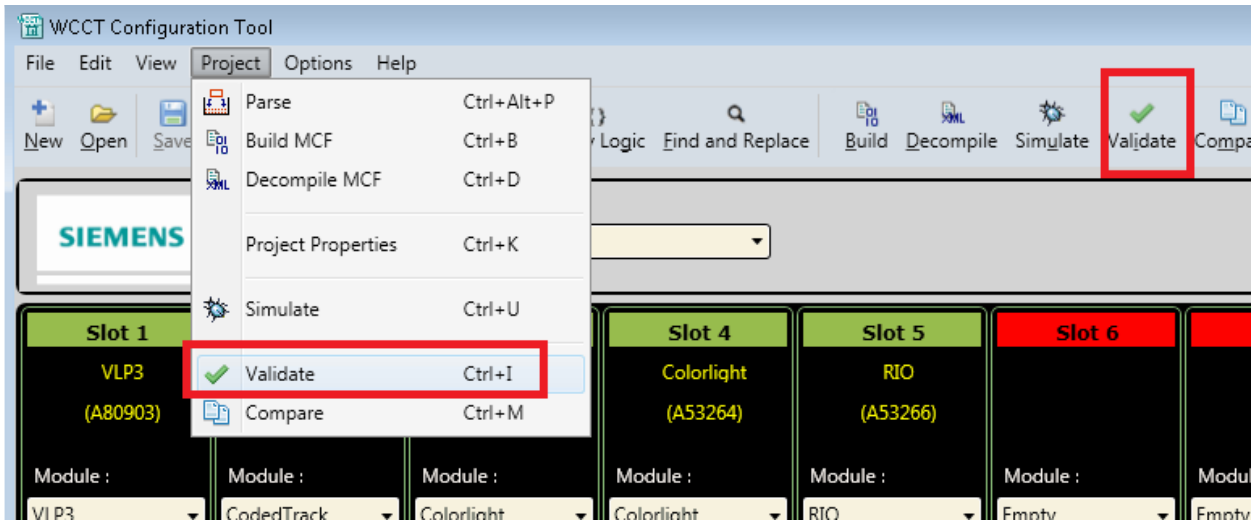


Figure 6-11 Selecting Validate Option

When **Validate** is selected it will first decompile the MCF. If there is no MCF, the validate button will be disabled. If the project source files have changed since the MCF was last built, the WCCT will give the following error message:

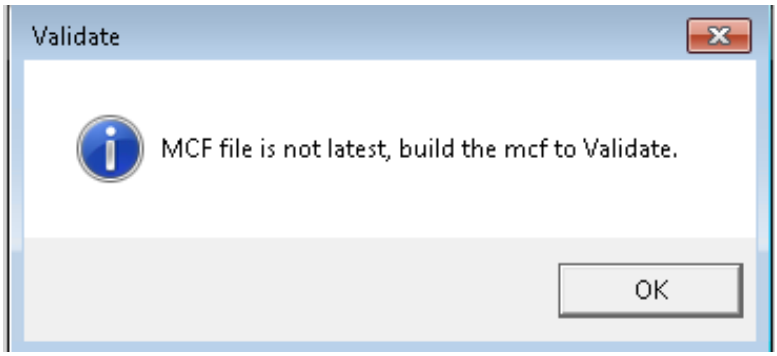


Figure 6-12 MCF Not Current Error

When the **Validate** operation completes successfully, the WCCT will show the following in the output window:

```
Validation : Passed
```

The tool will also open up a text window containing the validation comparison and report.

This report has 4 sections; the first section shows the project name, MCF CRC

```
--------------------------------------------------------------------------------
WCCT    : 1.5.2
Project :test12
MCF Name :test12001.mcf
MCF CRC : 0x40100272
--------------------------------------------------------------------------------
Comparison Report Between :test12 and test12_v_DComp001
Generated: 5/29/2017 6:34:48 PM
```

The next two sections show the results of comparing the project source file, and should have **Number of Differences 0.**

```
-----------------------------------Compare:BEGIN---------------------------------
[C:\Siemens\WCCT\Workspace\test12\test12.gc]
[C:\Siemens\WCCT\Workspace\test12_v_DComp001\test12.gc]
Files C:\Siemens\WCCT\Workspace\test12\test12.gc and
C:\Siemens\WCCT\Workspace\test12_v_DComp001\test12.gc are identical
Number of Differences : 0
-----------------------------------Compare:END-----------------------------------

-----------------------------------Compare:BEGIN---------------------------------
[C:\Siemens\WCCT\Workspace\test12\test12.gl]
[C:\Siemens\WCCT\Workspace\test12_v_DComp001\test12.gl]
Files C:\Siemens\WCCT\Workspace\test12\test12.gl and
C:\Siemens\WCCT\Workspace\test12_v_DComp001\test12.gl are identical
Number of Differences : 0
-----------------------------------Compare:END-----------------------------------
```

The last section shows the results of a binary comparison of the original MCF and the recreated MCF. There will be two known and acceptable differences listed here: the project date/time and the MCF CRC.

```
-----------------------------------Compare:BEGIN---------------------------------
[C:\Siemens\WCCT\Workspace\test12\MCF_Dir\test12001.txt]
[C:\Siemens\WCCT\Workspace\test12_v_DComp001\MCF_Dir\test12001.txt]
13,14c13,14
< * MCF File Time/ Date           :  29-May-2017 18:33
< * MCF CRC                       :  0x40100272
---
> * MCF File Time/ Date           :  29-May-2017 18:34
> * MCF CRC                       :  0xC67CB06
16c16
< * Report Date / Time            :  29-May-2017 18:33
---
> * Report Date / Time            :  29-May-2017 18:34
Number of Differences : 2
-----------------------------------Compare:END-----------------------------------

Total Number of Differences Found : 2
```

```
Validation : Passed
```

The validation report is stored in the project workspace in the MCF_Dir folder and has the name: {projectname}_Validate.txt

6.4 PROGRAM LISTING

When the project is compiled, the WCCT creates a program listing file with the MCF name and a .txt extension.

This file has a header that shows the WCCT version information, the project name, the MCF name, the CRCs, and the information entered in the project properties window; for example:

```
********************************************************************************
** WCCT Version                 :  1.5.2
** Part Number                  :  9vD24
** Copyright  ©  2017 Siemens
********************************************************************************

********************************************************************************
* Project Name                 :  RSSI1
* Version                      :  002
* Project Type                 :  Wayside Controller Configuration Tool
* MCF File Name                :  RSSI1002.mcf
* MCF File Time/ Date          :  29-May-2017 18:49
* MCF CRC                      :  0x82861E37
* Hardware Description MCF     :  HWDESCWRFR014.MCF
* Report Date / Time           :  29-May-2017 18:49
* Description                  :  my desc
* Designer                     :  rdb
* RailRoad                     :  cunca
* Area                         :  NZ
* Location                     :  tasman
* ATCS Source Address          :  7.620.100.100.03
* Comments                     :  test locn
********************************************************************************
* Revised By                   :  rdb
* Comments                     :  comment 1 comment 2
* Date                         :  11-May-2017

* Revised By                   :  rdb
* Comments                     :  comment 3
* Date                         :  11-May-2017

* Revised By                   :  hhh
* Comments                     :  h
* Date                         :  12-May-2017

********************************************************************************
```

The report then contains the chassis information, for example:

```
********************************************************************************
* Type: 8Slot
* Name: gwe_Main
```

```
* SLOTS:
*            Module              Name
*            ------------------- -------------------
*    Slot 1  VLP3                sl1
*    Slot 2  CodedTrack          Etrack
*    Slot 3  Colorlight          sl3
*    Slot 4  Colorlight          sl4
*    Slot 5  RIO                 sl5
*    Slot 6
*    Slot 7
*    Slot 8  CodedTrack          sl8

********************************************************************************
```

For each module the report shows the configuration information and module I/O, for example:

```
--------------------------------------------------------------------------------
Slot 2 - CodedTrack
--------------------------------------------------------------------------------

Configuration:
    TrkTxVoltage                    : 1500 mV
    Code5                           : Alternating           [Default]
    EC4Compatibility                : EC4Plus
    NonVitalCodeChangeCycles        : 1                     [Default, UCN]
    VitalCodeChangeCycles           : 2                     [Default, UCN]
    ShuntDropCycles                 : 2                     [Default, UCN]
    ShuntPickCycles                 : 5                     [UCN]
    CurrentLimit                    : 10000 mA

CTIO:
    Channel Name: LTK
    CodedTrack Outputs:                     CodedTrack Inputs:
    -------------------                     ------------------
    Code 1 Tx :LTKO1                        Code 1 Rx :LTKI1
    Code 2 Tx :LTKO2                        Code 2 Rx :LI2
    Code 3 Tx :LTKO3                        Code 3 Rx :LI3
    Code 4 Tx :                             Code 4 Rx :LI4
    Code 5 Tx :LTKO5                        Code 5 Rx :LTKI5
    Code 6 Tx :LTKO6                        Code 6 Rx :LTKI6
    Code 7 Tx :LTKO7                        Code 7 Rx :LI7
    Code 8 Tx :LTKO8                        Code 8 Rx :LI8
    Code 9 Tx :                             Code 9 Rx :
    Code M Tx :                             Code M Rx :
                                        Vital Code Rx :LTKVCP
                                        Non Vital  Rx :LTKCP

VRO Outputs:

Stick LED:
    Stick LED Control : LSTK
```

The report then shows the sections from the Logic Data editor

```
********************************************************************************
Logic Data
```

```
********************************************************************************
--------------------------------------------------------------------------------
Boolean Variables
--------------------------------------------------------------------------------
Num Name                      Initial        Log       Normal      Comment
--------------------------------------------------------------------------------
1   AFDT                      FALSE          No        FALSE
2   AFDTE                     FALSE          No        FALSE
3   LTLOSTMR                  FALSE          No        FALSE


--------------------------------------------------------------------------------
Timers
--------------------------------------------------------------------------------
Num Timer Name           Min  Max  DUR  Units Display                  ROnly UCN Cmnt
--------------------------------------------------------------------------------
1   tmr_AFDT             8    8    8    sec   AFDT                     No    No  [1]
2   tmr_LTLOSTMR         0    20   2    sec   LTLOSTMR                 No    No
3   tmr_RTLOSTMR         0    20   10   sec   RTLOSTMR                 No    No
4   tmr_LCHTMR           20   20   20   sec   LCHTMR                   No    No
5   tmr_RCHTMR           20   20   20   sec   RCHTMR                   No    No
--------------------------------------------------------------------------------
Timer Comments:
1.  tmr_AFDT            :TIMERS
```

Comments are shown in the report as references to a list of comments below the table, as shown above. The report finishes by listing the equations in Boolean text format.

```
********************************************************************************
Equations
********************************************************************************
# comments about eqn
AFDTE = (LSAPR || RSAPR) && !AFP;

StartTimerCond(AFDTE, tmr_AFDT);
```

6.5 COMPARING FILES

To compare two projects, first open one of them, then select **Compare** from the project menu, the **Compare** icon, or the **Ctrl M** shortcut. In order to compare projects, the projects must have been compiled so that the program listing files are available.

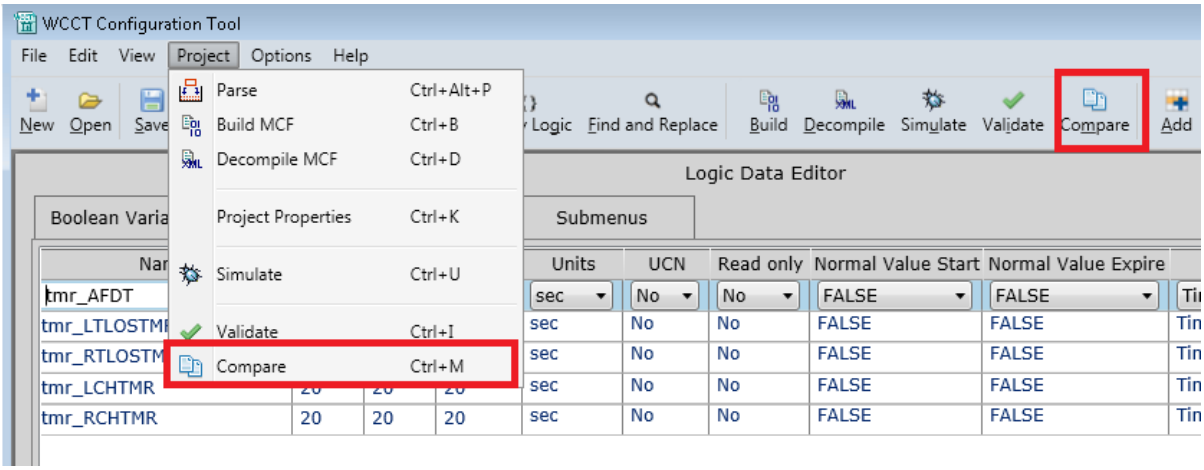


Figure 6-13 Selecting the Compare Option

When **Compare** is selected, the WCCT will open a browse window. From here select the second project's .**project.xml** file.

The WCCT will do a comparison of the program listing files (see previous section for details) from the two projects, and bring up a text window with the differences between the files.

If the listing files are not up to date (i.e the project source files are newer than the listing file) the WCCT will give the following warning:

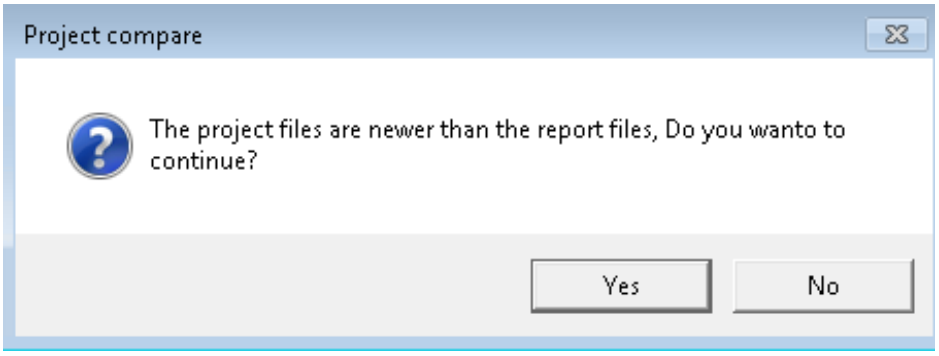


Figure 6-14 Listing File Not Current Warning

6.6 PROJECT REPORT

The WCCT creates a text version of the Program Listing that can be opened in a text editor, but this doesn't contain the actual relay logic diagram. The WCCT can, however, also create a PDF version that will contain the relay logic diagrams.

To create the report, select the **Report** icon or use the **Ctrl R** shortcut.

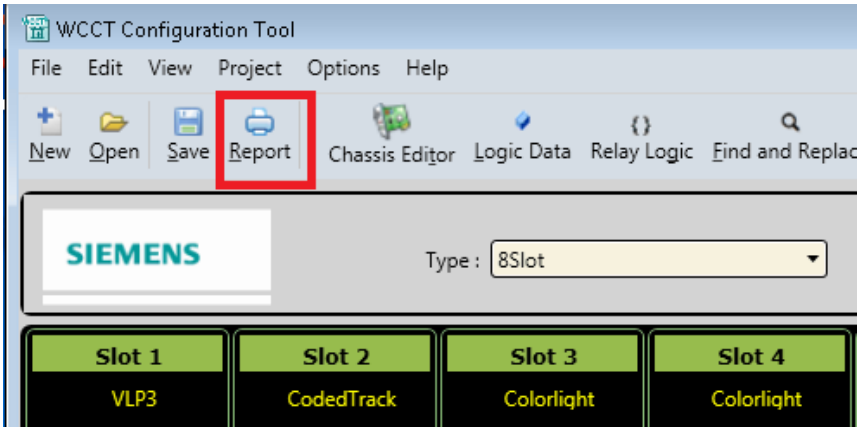


Figure 6-15 Selecting the Report Option

The WCCT also has the option to create a new report whenever the MCF is built. To turn this option on or off, open the settings window from the **Options>Settings** menu and check the **Generate Report on Compile** check box if this is required.

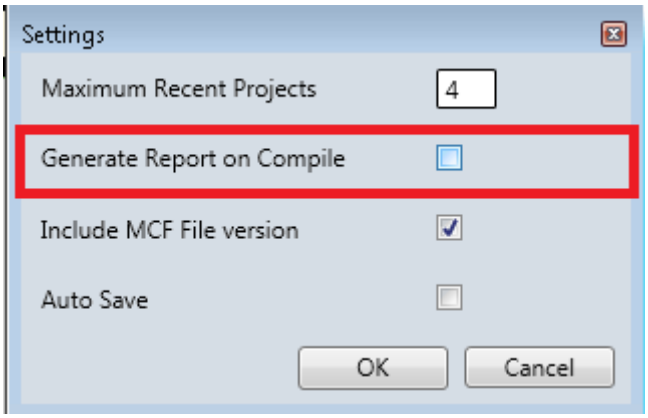


Figure 6-16 Selecting to Generate a Report on Compile

If **Generate Report on Compile** is checked, when the user presses **Build** the WCCT will create the MCF and then create a {project}.pdf report file. The PDF report file will be stored in the usual location: {Workspace}/{project}MCF_Dir.

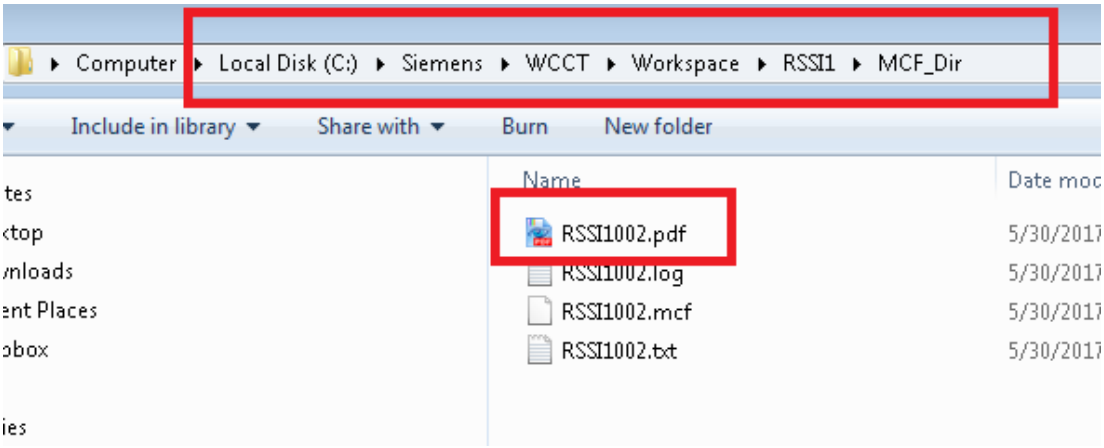


Figure 6-17 Build Products

When the user presses the **Report** icon, a print preview window is created. Drag the corners of the window to re-size it. Use the **Zoom** to increase magnification.

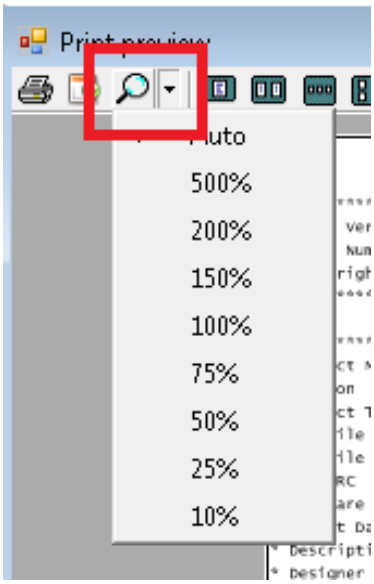


Figure 6-18 Print Preview Zoom

The report will start off with the identical information to the program listing. Use the page-up/down or the page selector in the top-right to navigate the report.

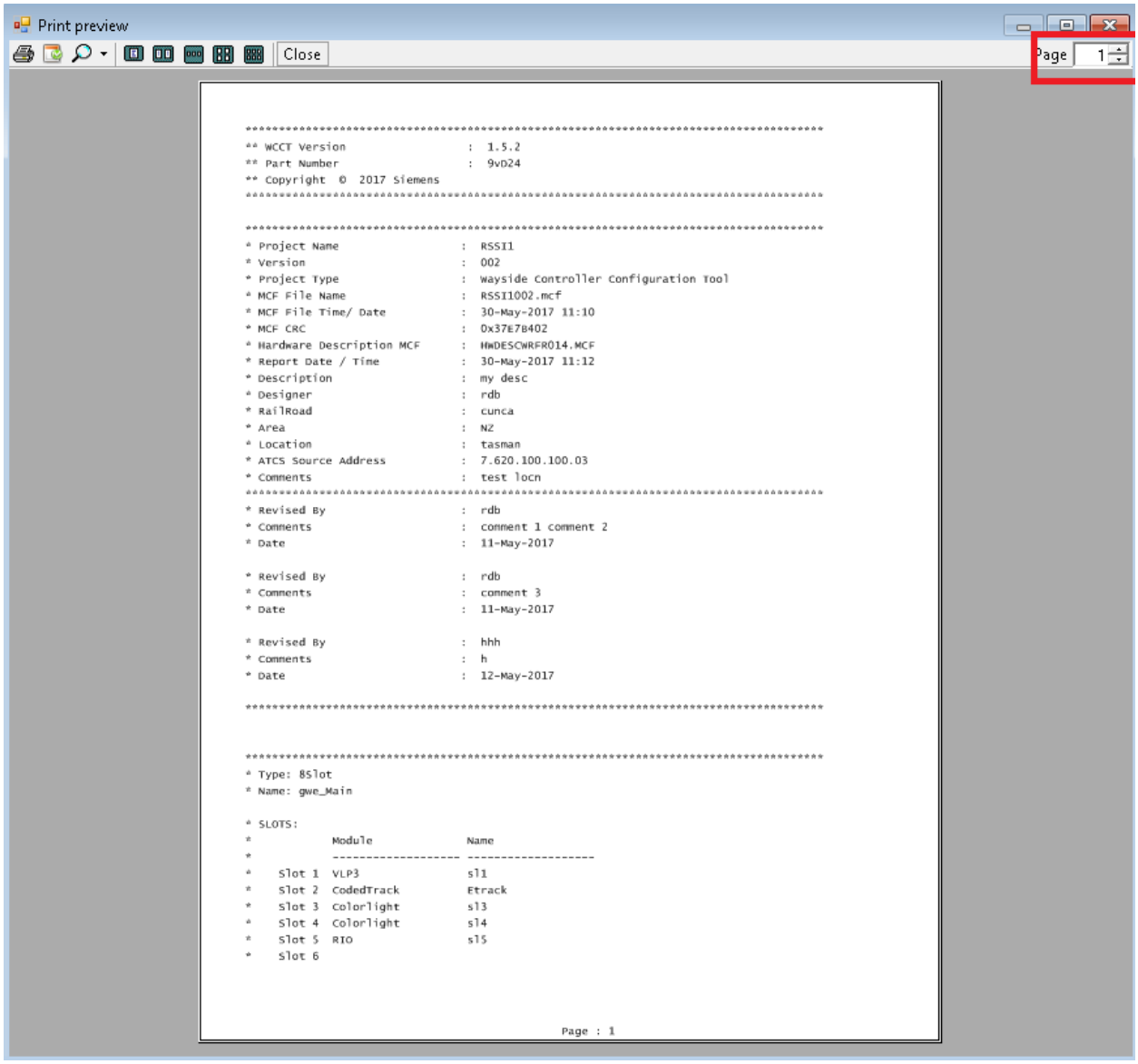
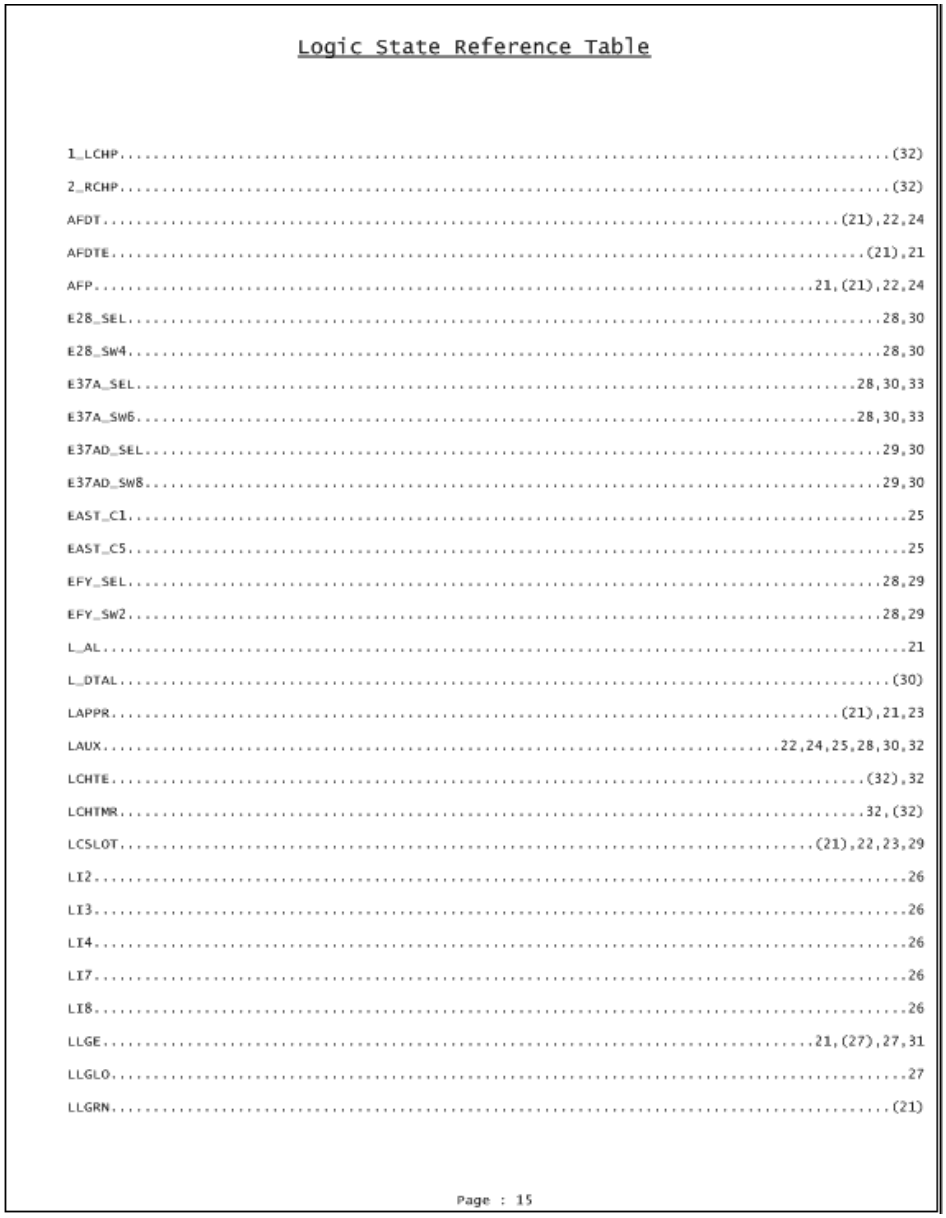


Figure 6-19 Print Preview

After showing the textual Boolean format for the logic, the report will show a logic state reference table. This shows the page numbers on which logic states are used. If the page number is in parentheses, this indicates the page number on which the logic state is defined.



Logic State Reference Table

1_LCHP.....(32)
2_RCHP.....(32)
AFDT.....(21),22,24
AFDTE.....(21),21
AFP.....21,(21),22,24
E28_SEL.....28,30
E28_SW4.....28,30
E37A_SEL.....28,30,33
E37A_SW6.....28,30,33
E37AD_SEL.....29,30
E37AD_SW8.....29,30
EAST_C1.....25
EAST_C5.....25
EFY_SEL.....28,29
EFY_SW2.....28,29
L_AL.....21
L_DTAL.....(30)
LAPPR.....(21),21,23
LAUX.....22,24,25,28,30,32
LCHTE.....(32),32
LCHTMR.....32,(32)
LCSLOT.....(21),22,23,29
LI2.....26
LI3.....26
LI4.....26
LI7.....26
LI8.....26
LLGE.....21,(27),27,31
LLGLO.....27
LLGRN.....(21)

Page : 15

Figure 6-20 Logic State Reference Screen

After the logic state references, the report shows the relay logic diagrams.

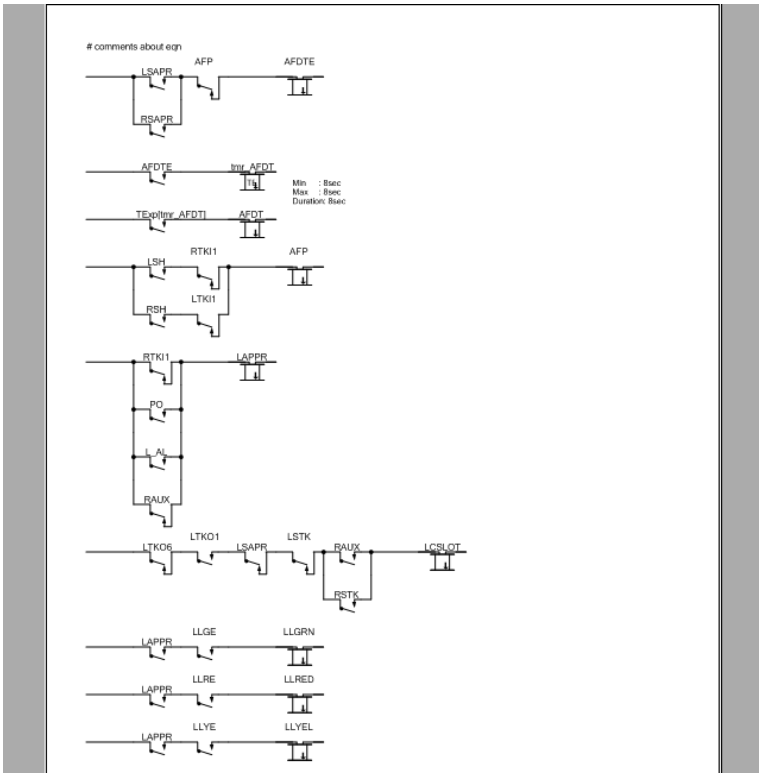


Figure 6-21 Relay Logic Diagram in Report

If equations are too deep or wide to fit on the page, they will get split across pages. If this happens, the WCCT will bring up a window showing the **Split Equation Information**. This window shows which equations are split and what page they appear on. Split type is V (vertical), H (horizontal) or VH (vertical and horizontal).

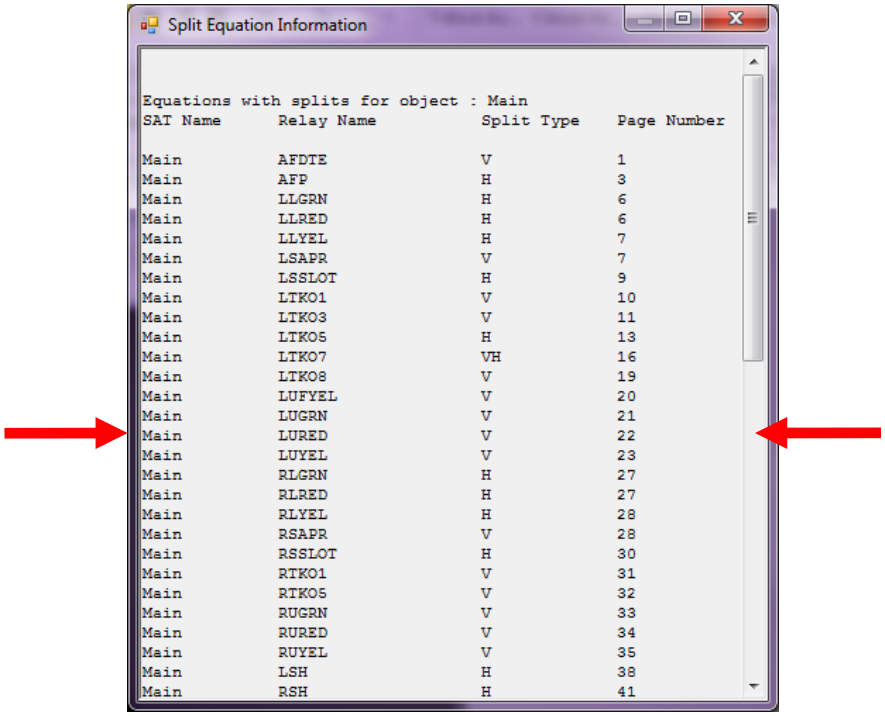


Figure 6-22 Split Equation Report

As shown below, this equation is split vertically and horizontally.

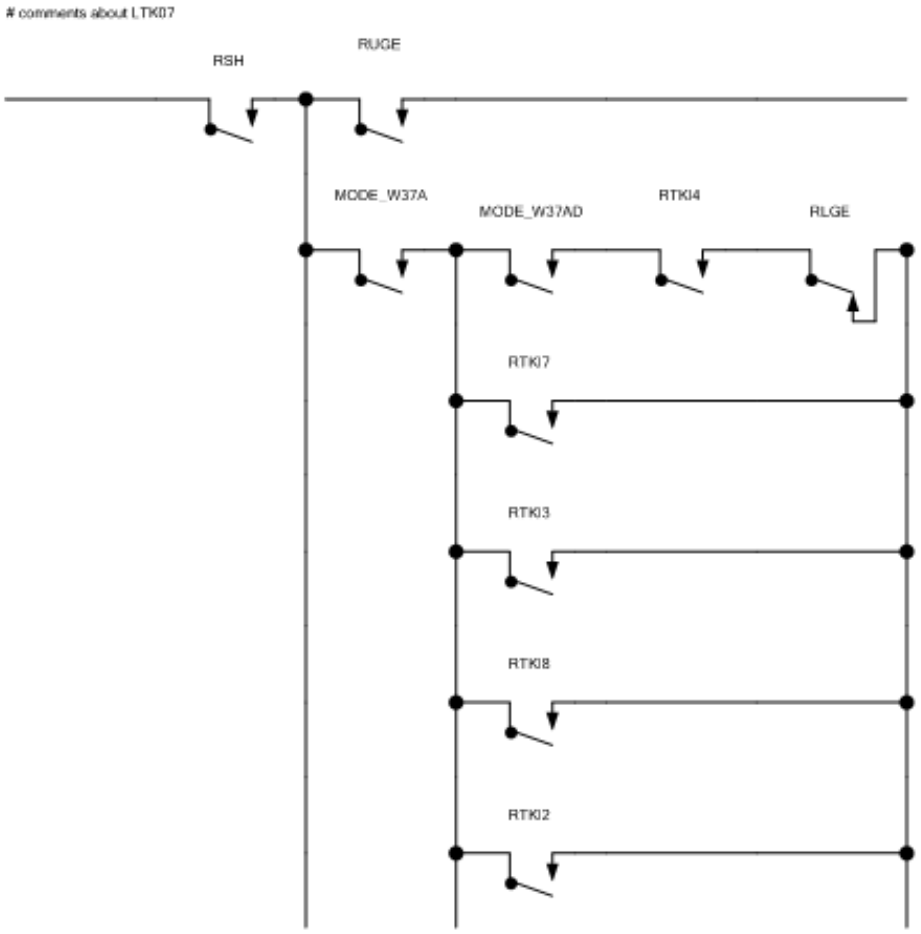


Figure 6-23 Split Equation Window Example

The user should inspect the print preview to see if it is readable enough. If too many equations are split, or the equations are too small, the relay sizes can be adjusted. To do this, close the report and go to the Relay Logic Editor and select the **Resize** icon or the **Ctrl +ALT + R** shortcut. See section 6.6.2 for details.

To change the paper size, select the **Printer Properties** from the print preview.

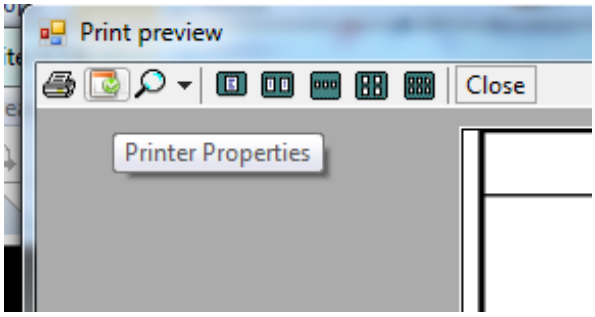


Figure 6-24 Printer Properties

Then select the appropriate orientation and paper size.

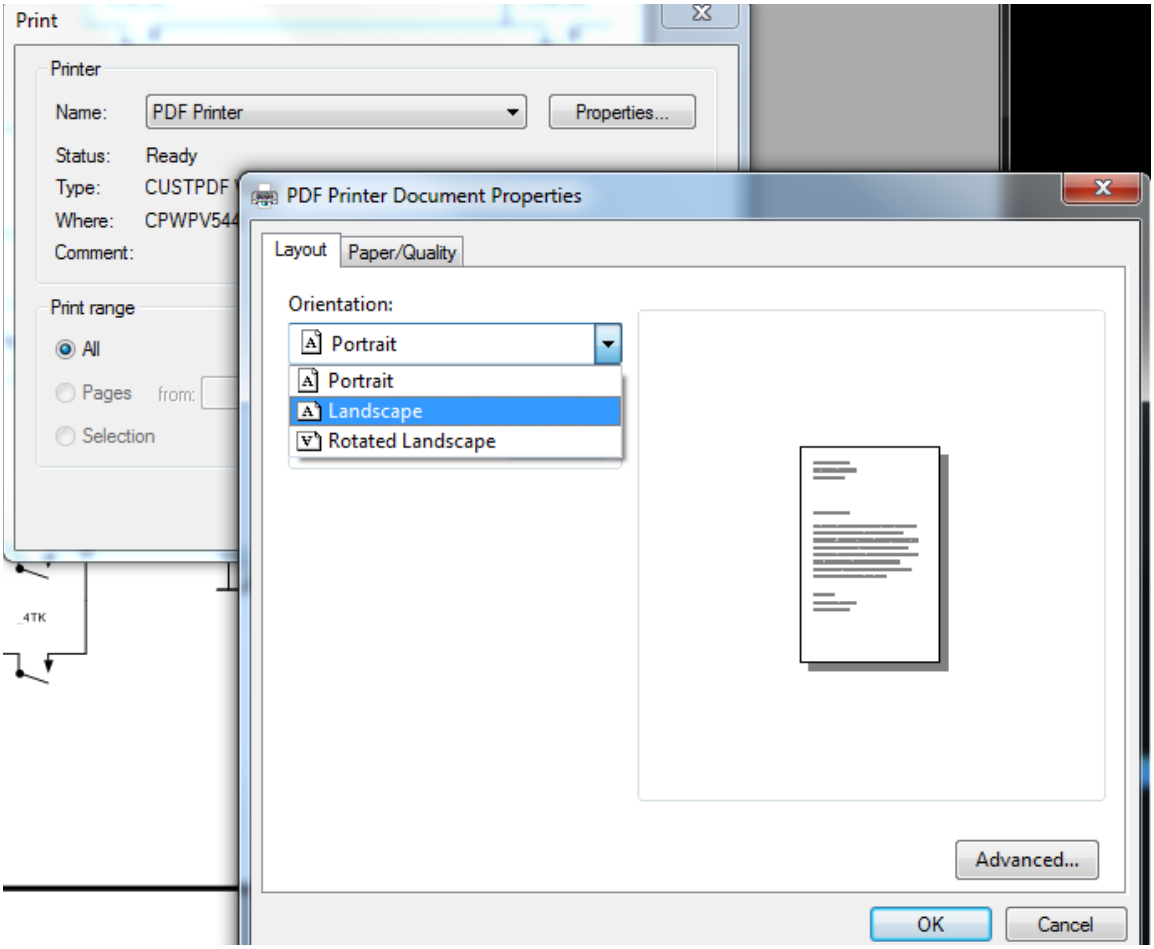


Figure 6-25 Setting Printer Properties

For example, changing the landscape made the equation illustrated below a lot more readable.

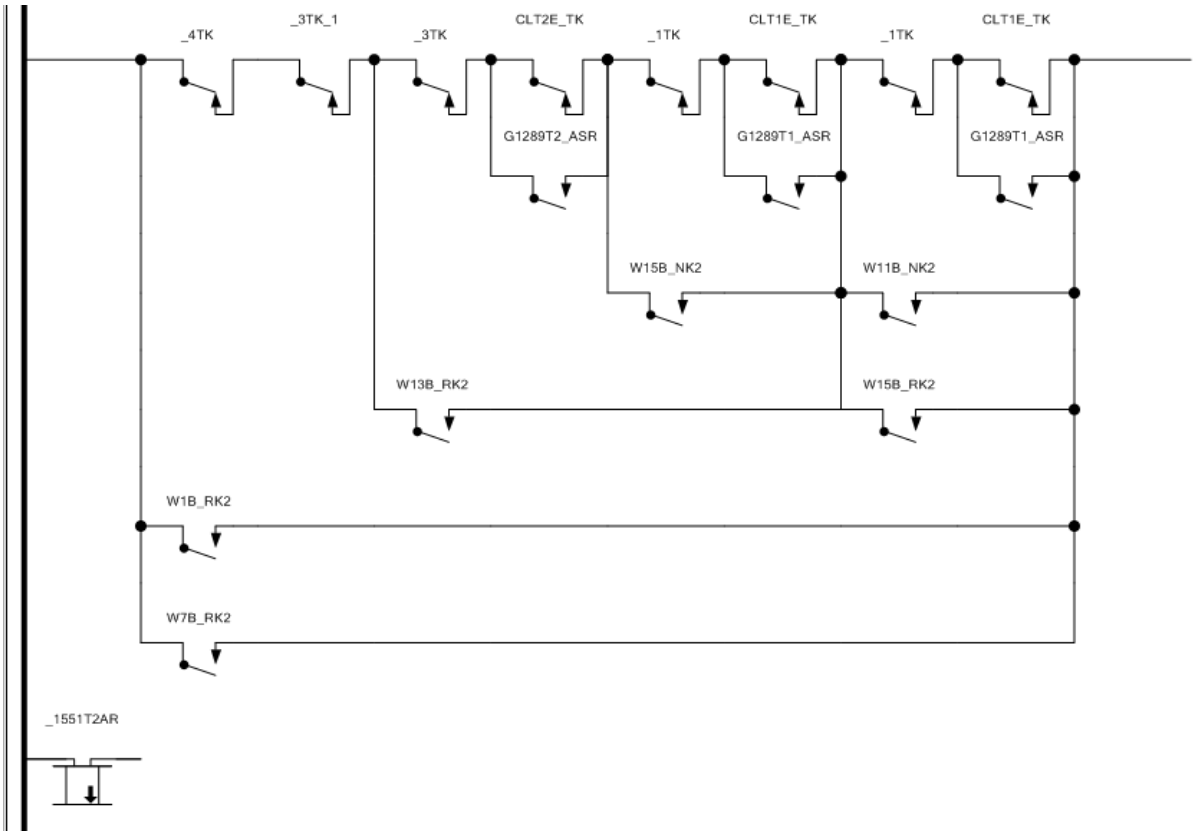


Figure 6-26 Large Split Equation

By reducing the relay size from 100% to 90%, the entire equation is able to fit on one page.

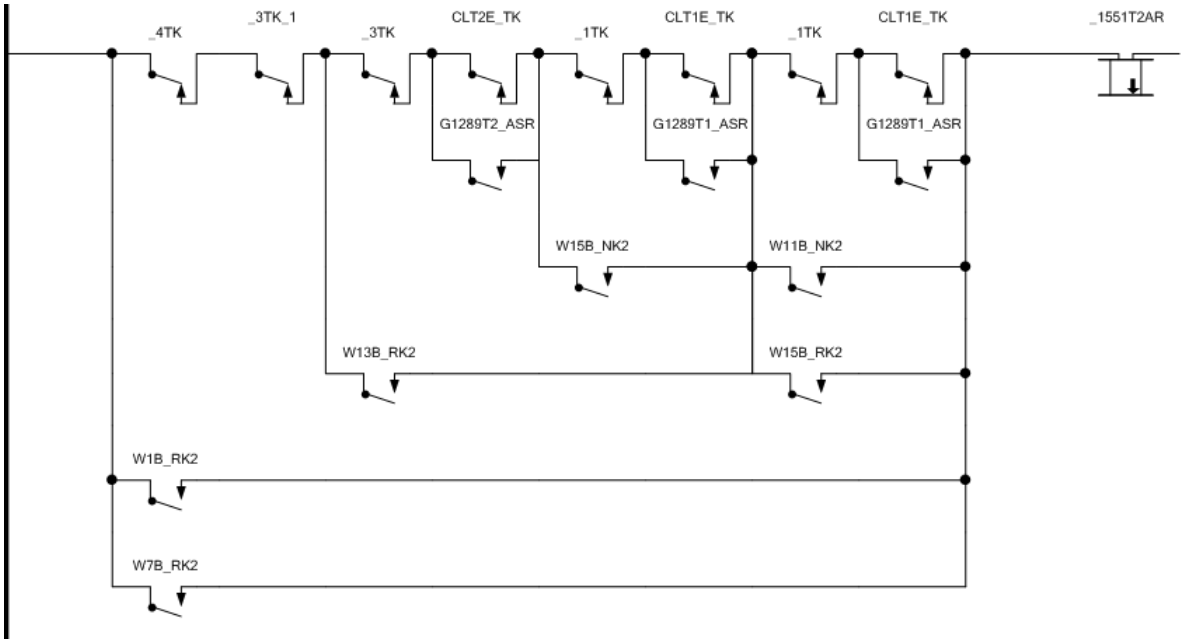


Figure 6-27 Page Landscape and Sizing Example

Before printing, setup the printer using the **Print Setup** button.

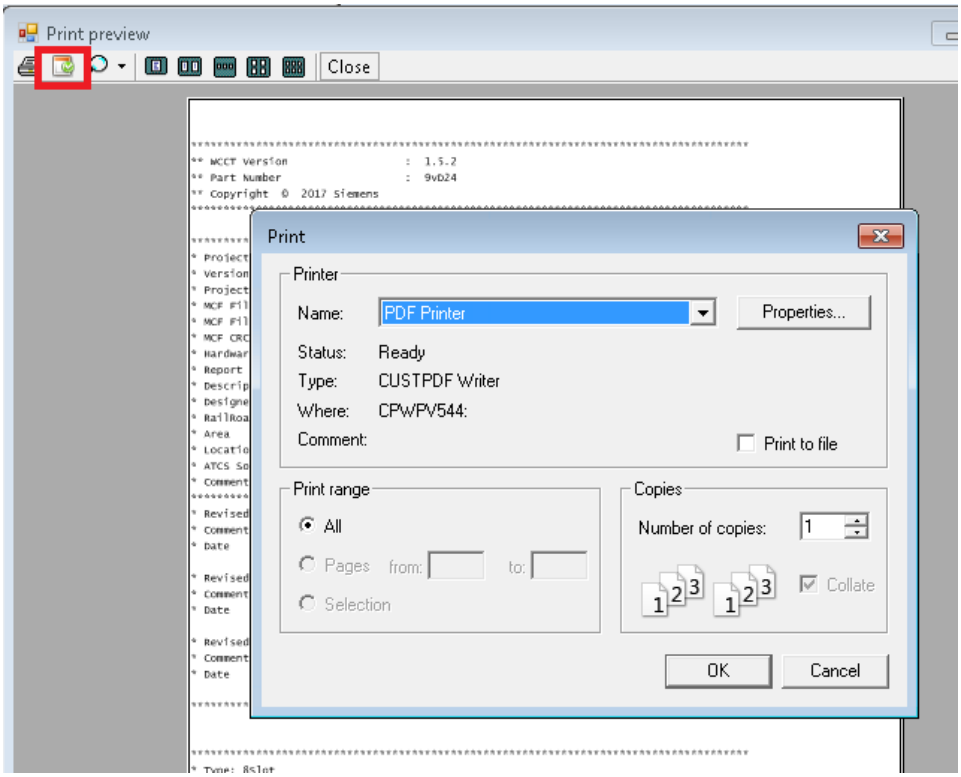


Figure 6-28 Print Setup

To create the PDF document from the **Print Preview** select the **Print** icon.

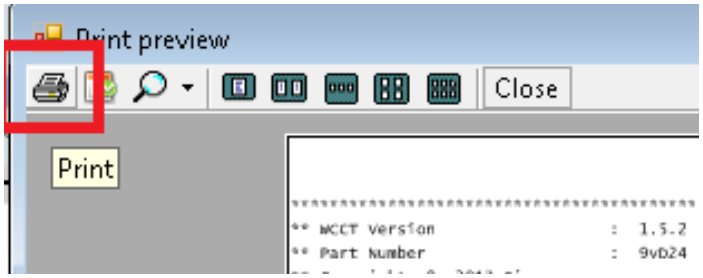


Figure 6-29 Printing the PDF

6.6.1 Changing the Font Size

Select the **Font** icon in the top-right of the Relay Editor to change the fonts.

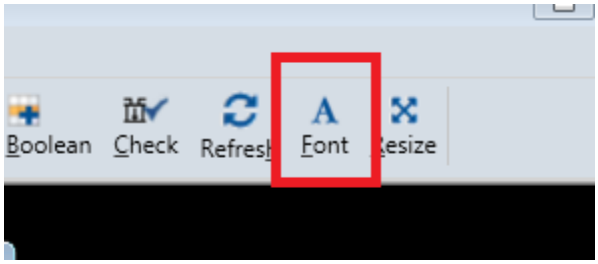


Figure 6-30 Changing Font

The **Font** selection window will appear as shown in the following figure.

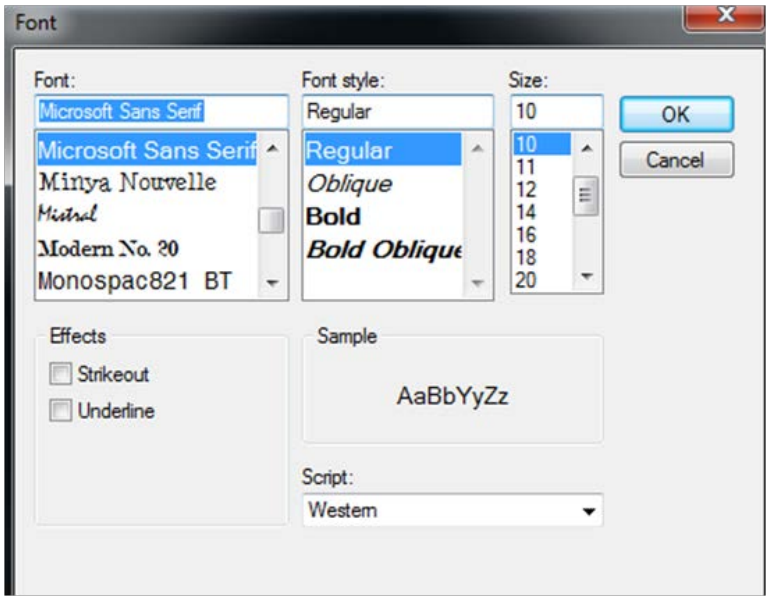


Figure 6-31 Font Selection window

6.6.2 Changing Relay Sizes

In order to be able to plot the relay diagram to a readable PDF, it may be necessary to change the relay sizes. Select the **Resize** icon or use the **Ctrl + Alt + R** shortcut to do this.

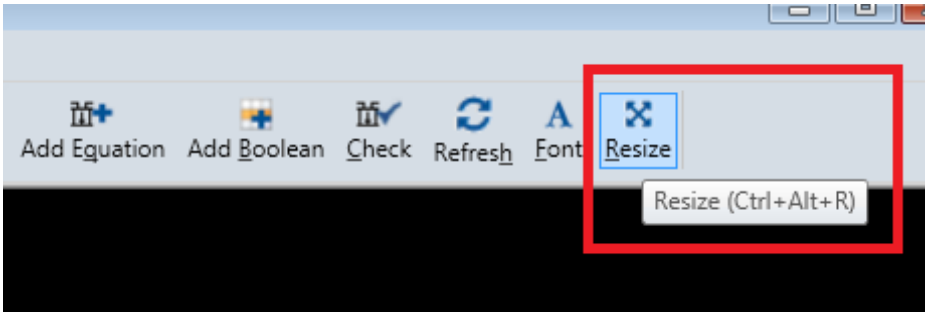


Figure 6-32 Changing Relay Size

The **Change Resolution Editor** will come up. The width and height of the relay can be changed. If the **Lock Aspect** is checked, the relays remain square with the height and width changing together.

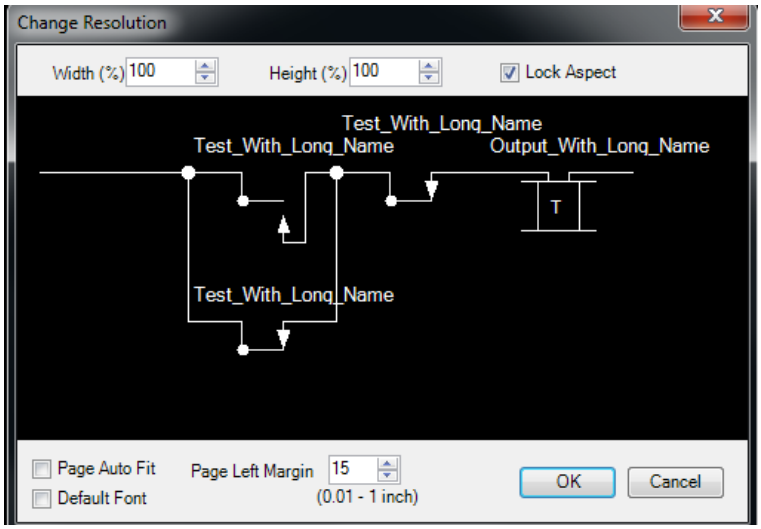


Figure 6-33 Change Resolution Editor

If **Page Autofit** is selected, the WCCT will find the largest equation and scale everything down so that it fits. This may result in equations too small to read, in which case it is better to uncheck **Autofit** and manually adjust the height/width until a suitable compromise between equation size and number of page splits is found.

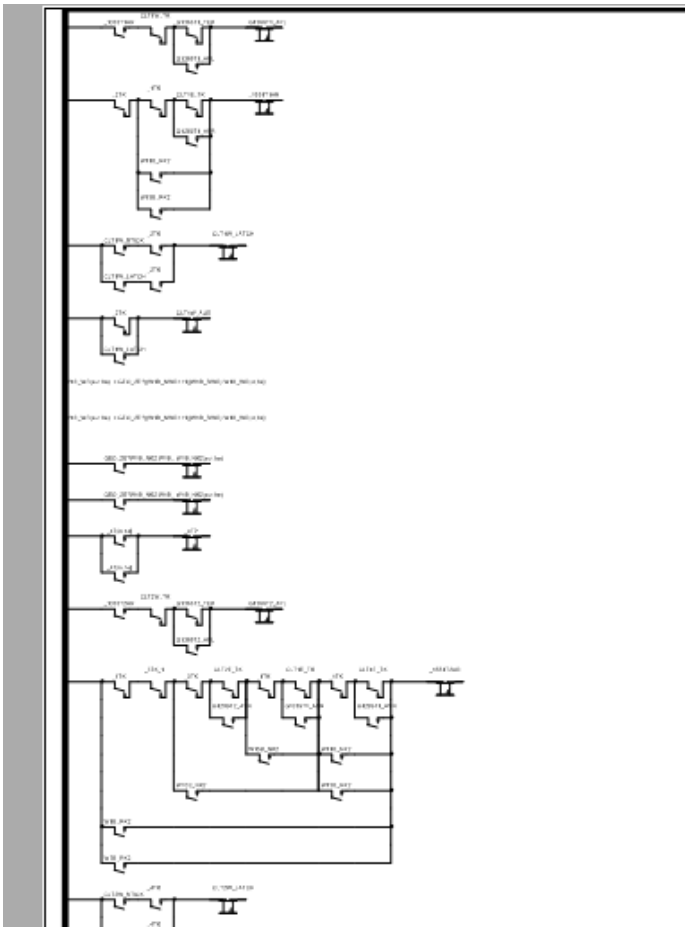


Figure 6-34 Autofit Window

This screen also allows you to set the left margin. If the PDF is to be printed and bound into a book along its left edge, a left margin may be needed to avoid cutting off the plot. Change the value of **Page Left Margin** to adjust the white space on the left side of the page. Selecting the **Default Font** check box will set the fonts back to their default settings.

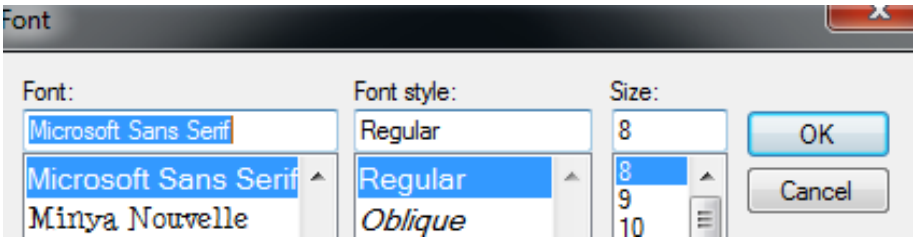


Figure 6-35 Default Font

This page intentionally left blank

APPENDIX A ERROR MESSAGES

ERROR MESSAGE TABLES

The following table lists some of the more common errors that are detected by the parser.

Error Msg	Example and Remedy
(88): error: Failed to find module in slot [5] (slot is empty)	e.g. ASSIGN (Main, L_DTAL) (slot 5, VRO 5) No module has been defined in slot 5. Check that the slot number corresponds to a defined module
(91): error: Invalid channel number [5]	ASSIGN (Main, L_DTAL) (slot 3, VRO 5) The module in slot 3 does not have 5 VRO channels. Check the module type and the number of I/O supported by it.
(13): error: Undefined name [ALL] in SAT scope [objMain]	This error message point to the line ACTIVATE Main (ALL) This is not the real cause, the real cause is a logic state is used in an ASSIGN statement that is not defined as either an input or output in the object, e.g. LA_DTAL does not exist ASSIGN (Main, LA_DTAL) (slot 3, VRO 1)
(92): error: syntax error, unexpected IDENTIFIER, expecting ')'	A WCCT keyword is not recognized e.g. ASSIGN is misspelled ASIGN (Main, R_DTAL) (slot 4, VRO 1)
(57): error: Undefined object name [GEO]	e.g. GWECONFIG GNSCL_RSSI [atcsconfig=GEO, subnode=SN_Main, gwe=GEOIITwoTrackExtended] The atcsconfig name show above did not match the one defined in the ATCSCONFIG section ATCSCONFIG atcsGEO(
(57): error: Undefined name [_Main]	GWECONFIG GNSCL_RSSI [atcsconfig=atcsGEO, subnode=_Main, gwe=GEOIITwoTrackExtended](The subnode name show above did not match the one defined in the ATCSCONFIG section ATCSCONFIG atcsGEO(SUBNODE SN_Main [number=3](

Error Msg	Example and Remedy
(57): error: Invalid GWE type [GEOIITwoTrackExtende]	GWECONFIG GNSCL_RSSI [atcsconfig=atcsGEO, subnode=SN_Main, gwe=GEOIITwoTrackExtende](The chassis name used in the gwe is mistyped. He mod.tmpl.gc file for allowed chassis types
(58): error: syntax error, unexpected MODULE, expecting '('	e.g. GWECONFIG GNSCL_RSSI [atcsconfig=atcsGEO, subnode=SN_Main, gwe=GEOIITwoTrackExtended] MODULE sl1 [slot=1, module=VLP3] '(' is missing
(19): error: Invalid name [WFY_SL]	 e.g. LOGICALCONFIG lcGEO(OBJECT Main(# Vital Configuration WFY_SL [enable] = FALSE WFY_SL is not a property of the Main object
(19): error: Invalid GL value type (enumerator expected)	e.g. LOGICALCONFIG lcGEO(OBJECT Main(# Vital Configuration WFY_SEL [enable] = 1 EFY_SEL [enable] = FALSE W28_SEL [enable] = FALSE Check allowed values for WFY_SEL (from lcGEOtmpl.gc) , in this case it's a Boolean

(61): error: Undefined value [EC5]	e.g.
	MODULE sl2 [slot=2, module=CodedTrack](
	VCOVoltage2 = 0 mV, # min=0, max=4000
	Code5 = Alternating,
	EC4Compatibility = EC5,
	
	EC5 is not a valid value, check the mod.tmpl.gc file for valid values

Error Msg	Example and Remedy
(123): error: syntax error, unexpected ')', expecting '('	CONTROL NV_CTRL ((Main, NWZ) [bit=1], (Main, RWZ) [bit=2], (Main, FGZ) [bit=3],) There should be no comma on the last item
(241): error: Missing VCOM Channel (In and Out).	e.g. VCOMIN VCOM_REMOTE1 ((Main, GAGG) [bit=1], (Main, GBRG) [bit=6]) VCOMOUT VCOM_REMOTE1 ((Main, OSTP) [bit=1]) VCOMCFG VCOM_REMOTE (Version = 1, The VCOMCFG name did not match the names for its VCOMIN or VCOMOUT channel
(187): error: Invalid units [ms] for configuration parameter [MaxTimestampOffset] (expected [s])	e.g. MaxTimestampOffset = 10 ms Units should be ‘s’
(77): error: Undefined name [WEFTyp] in global scope	PROPERTY WEF_SEL [enum=WEFTyp, desc="Flash selection"] The enum name did not refer to a defined enumeration type
(326): error: Undefined name [IN10]	OUT3 = !(IN1 && IN10 && WFY_SEL); IN10 is not a defined VARIABLE
(248): error: Invalid module [VLP2]	MODULE sl1 [slot=1, module=VLP2] Check that the module is allowed in the defined e.g. gwe=GEOIITwoTrackExtended by checking the mod.tmpl.gc file
(262): error: Number of log statements used cannot exceed: 50	VARIABLE LAGG [enum=Boolean, log] The VLP2 module in slot 1 only supports 50 logged variables. Reduce the number of variables marked with ‘log.

Error Msg	Example and Remedy
(99): error: Invalid maxvalue for [T12345678912345678901]	The maximum timer value is 65535 (seconds or ms) So following gives error TIMER TIMER1 [minvalue=1, maxvalue=65536, defaultvalue=10, units="msec", desc="Timer 2", excludeucn]

The following table lists some of the more common errors that are detected when compiling

Error Msg	Example and Remedy
ERROR: WriteMCFBinary:DOM parse error [-1072897660]. . maxInclusive constraint failed. The element: 'NumberOfSATTimers' has an invalid value according to its data type.	The number of TIMERs exceeds the maximum allowed by the selected CPU module in slot 1. The VLP2 supports a maximum of 32 timers
ERROR: WriteMCFBinary:DOM parse error [-1072897660]. . maxInclusive constraint failed. The element: 'MaxSATLogicState' has an invalid value according to its data type.	The number of VARIABLES exceeds the maximum allowed by the selected CPU module in slot 1. The VLP2 supports a maximum of 2000 VARIABLES in one object
ERROR: WriteMCFBinary:DOM parse error [-1072897660]. . maxInclusive constraint failed. The element: 'PropertyNameLength' has an invalid value according to its data type.	The name of a logged variable is too long, logged variables need to be less than 16 chars e.g. following name is 17 chars VARIABLE I12345678911234567 [input, enum=Boolean, log]
ERROR: WriteMCFBinary:DOM parse error [-1072897660]. . maxInclusive constraint failed. The element: 'TimerNameLength' has an invalid value according to its data type.	The name of a timer is too long, maximum length is 20 characters e.g. following name is 21 chars TIMER T12345678912345678901 [minvalue=1, maxvalue=20, defaultvalue=10, units="msec", desc="Timer 2", excludeucn]
Error : Only 16 user editable properties are allowed on one configuration screen, go into Logic Data Editor Properties and Timers screens and move properties and timers onto their own screens such that there are less than 16 per screen using	Too many configuration properties + timers on 1 page, limit for VLP2 is 16 per page and VLP3 25 per page. Uses the "::" syntax rto create submenus

APPENDIX B HWDESC FILES

HARDWARE DESCRIPTION FILES

The Hardware Description File (HWDESCWRFRxxx.mcf, xxx is the version number) is a file supplied with the WCCT that describes the available chassis' and modules in the GEO/WayConneX system. When the user parses or compiles a project, the WCCT checks against information in that file. Whenever the GEO/WayConneX is upgraded to support new module types or changes are made to existing ones that affect their configuration or I/O channels, a new HWDESC MCF is released. The existing projects the user has created will continue to use the HWDESC that was created originally, unless the user specifically upgrades them. If the user needs to update an existing project to use the new feature, they should upgrade to the new HWDESC file. To do this, right-click on the project and select **Change Hardware Description MCF**.

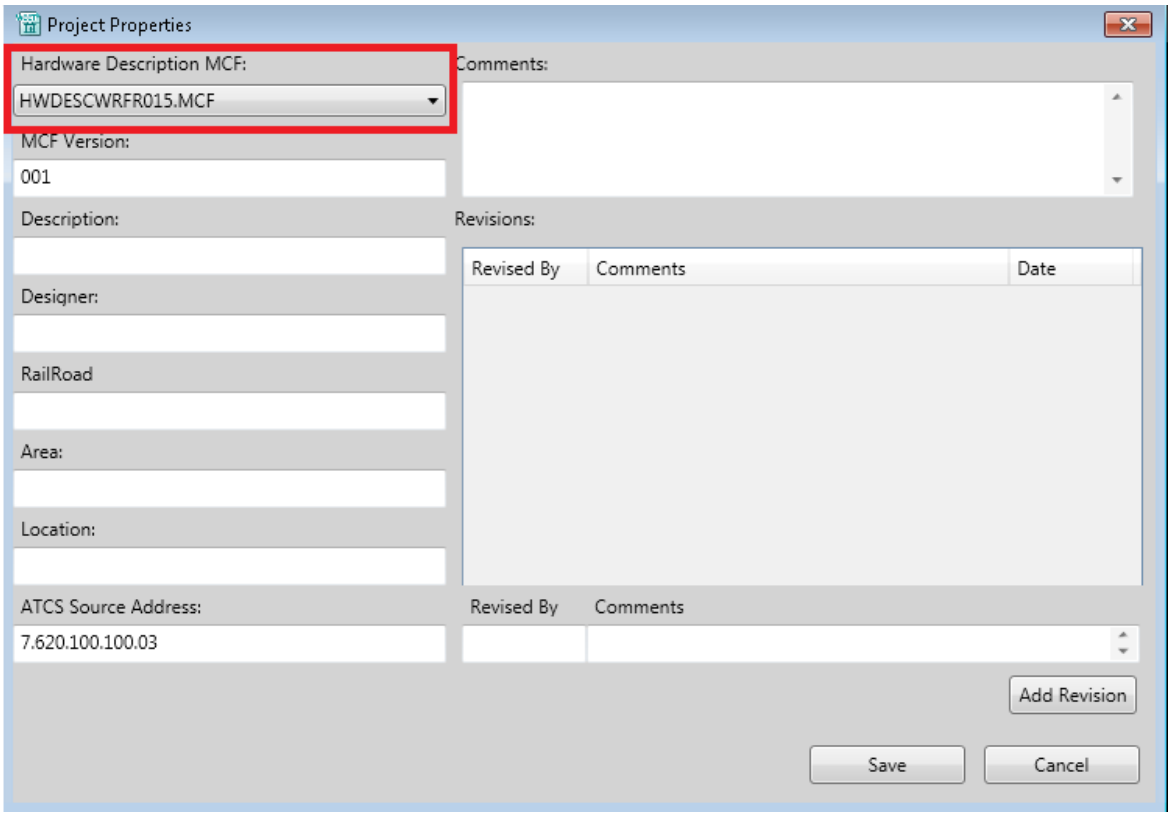


Figure B-1 Selecting HWDESC MCF

A drop down list will appear with any other HWDESC MCFs that are available.

The WCCT only comes with the latest HWDESC MCF. If, in the future, an older HWDESC is required that is not present on the computer that the WCCT is installed on, get the old HWDESC and copy to the new machine as follows:

Navigate to the C:\siemens\WCCT\HWDescMCF\GEO directory and select the old HWDESC MCF file to copy from the old machine, copy the .log and .mcf for the version required to the new machine.

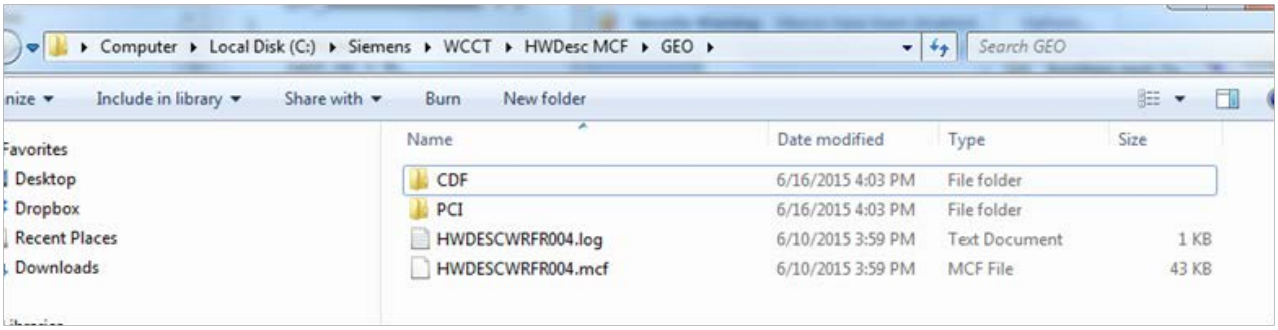


Figure B-2 HWDESC MCF File Location

APPENDIX C LIMITS

PARAMETER LIMITS

Item	CPU II+	CPU III	Notes
Total Number of Variables	3000	3000	
Number of Timers	32	120	
Number of VComms Channels	10	10	
Number of bits in VComms Channel	80	250	
Number of PTC signals	0	20	PTC not supported on CPU II+
Number of PTC switches	0	20	
Number of PTC hazard detectors	0	20	
Number of PTC aspects	0	32	

String formats:

Type	Max Length	Format
Property Names	40 characters	a-z, A-Z, “_”, 0-9. Not all numerical, can’t start with number
Property Description	40 characters	a-z, A-Z, “_”, 0-9, space, following punctuation characters . , ; : _ - !
Timer Name	20 characters	a-z, A-Z, “_”, 0-9. Not all numeric, can’t start with number
Timer Description	40 characters	a-z, A-Z, “_”, 0-9, space, following punctuation characters . , ; : _ - !
Channel Names	20 characters	a-z, A-Z, “_”, 0-9. Not all numerical
I/O Names	30 characters	a-z, A-Z, “_”, 0-9. Not all numerical
Boolean Variable Names	16 characters	a-z, A-Z, “_”, 0-9. Not all numerical
Project Name	13 characters	a-z, A-Z, “_”, 0-9, no spaces
Submenu Names	16 characters	a-z, A-Z, “_”, 0-9., space, following punctuation chars ‘ ; : _ - ! = / Note: comma, := and :: not allowed

This page intentionally left blank

APPENDIX D WCCT SOURCE FILES

SOURCE FILES

The source files for a WCCT project consist of the following:

- .project.xml file
- .GC file
- .GL file.

The .project.xml contains a list of what other files are used

The .GC file contains:

- The definition of the which chassis is used
- Which modules are used in the chassis
- The default values for I/O module configuration parameters
- How inputs and outputs variables are assigned to I/O modules
- Communications interface definitions for:
    - Vital communication channels
    - Non-vital communication channel (controls, indications)
    - PTC interface channels

The .GL file contains:

- The definition of inputs, output and internal variables (aka relays, Boolean states)
- The user logic
- Definition of user timers
- Definition of user configurable properties

WARNING

THE CONTENTS OF THE GC AND GL FILE ARE AUTOMATICALLY GENERATED FROM THE DATA ENTERED INTO THE EDITORS IT IS NOT RECOMMENDED THAT THE USER MANUALLY EDITS THESE FILES AS THIS COULD LEAD TO INCONSTISTENCIES AND LOST DATA IF SUBSEQUENT CHANGES ARE MADE USING THE WINDOW EDITORS.

This page intentionally left blank

APPENDIX E OPENING OLD PROJECTS

OPENING OLD PROJECTS

WCCT versions up to and including 1.3.6 used a different format and file name for the project file. The project file was called .project.

The new WCCT can be used to open old projects. When the project is opened, a copy of it will be created in the current workspace location and it will be upgraded to the new format, so a {projectname}.project.xml will be created.

The old project will have used an older version of the HWDESC MCF in which the names of the chassis are different.

To open the old project, select **Open** as usual. Then go to the location of the old project and change the drop down menu in the bottom right to **Old Project Files (*.project)** as shown below. The .project for the old project will now be visible.

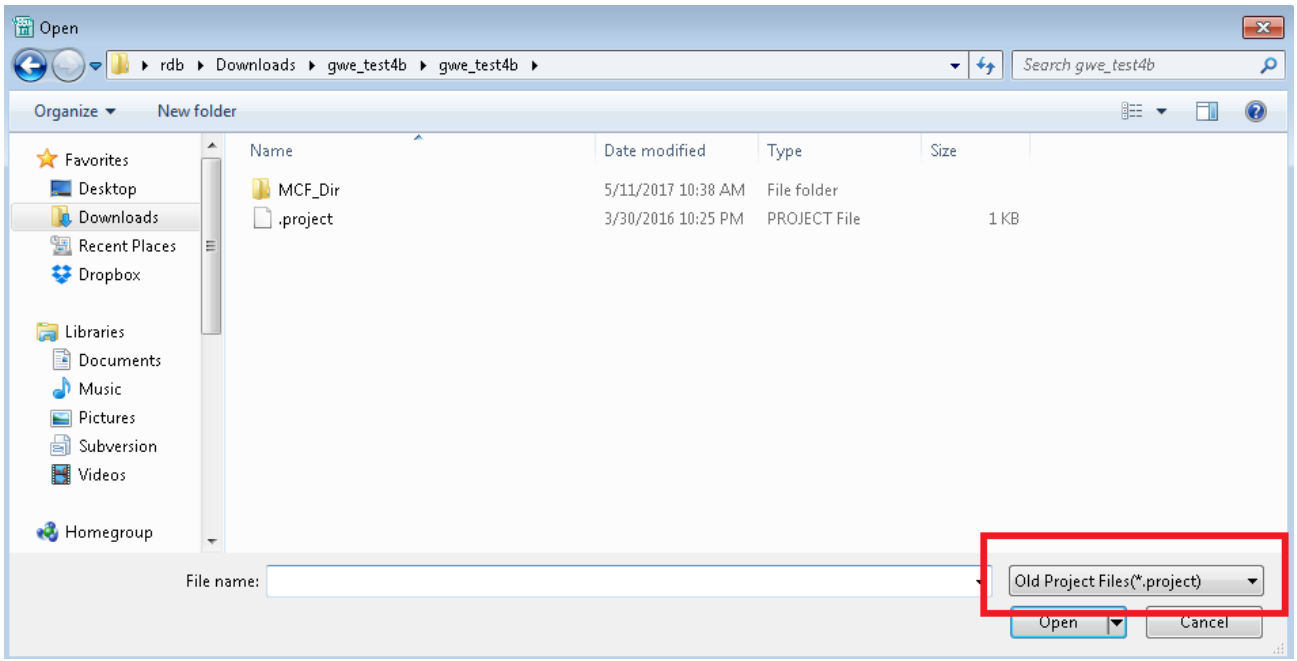


Figure E-3 Opening an Old Project

Select the .project file and double-click or select the **Open** button. The WCCT will show a message like the following:

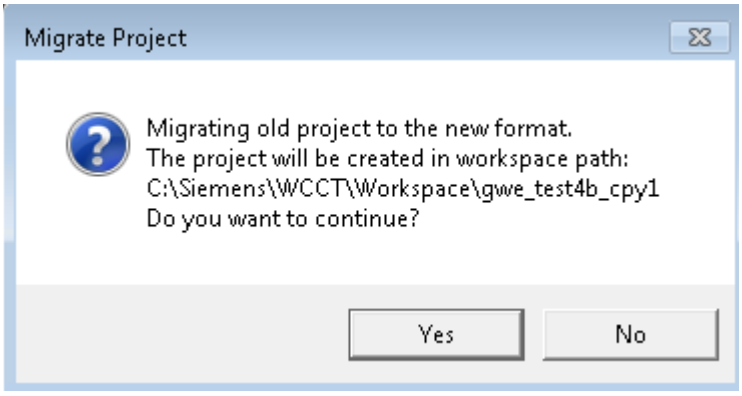


Figure E-4 Old Project Migration Message

If **No** is selected, the WCCT will close the window and nothing further will happen.

The old project will have been built with an older HWDESC MCF than the current one for the new WCCT, so, the WCCT will likely give a message like the following:

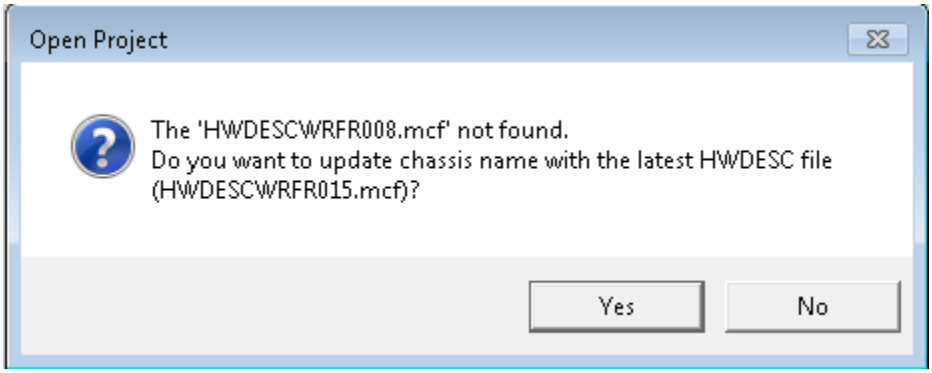


Figure E-5 Old HWDESC Not Found Message

If the HWDESC MCF has been installed the message will be slightly different:

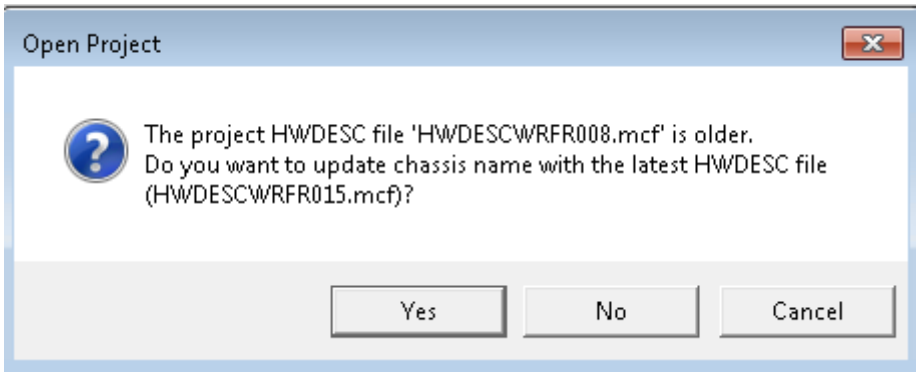


Figure E-6 Upgrade to new HWDESC MCF Message

It is recommended to upgrade the projects to the new HWDESC, select **Yes**.

When this is done, the WCCT will upgrade the chassis information so the new chassis name is used, e.g TwoTrackExtended. The new project will be saved in the new location, indicated in the bottom-right. The original old project is not modified in any way.

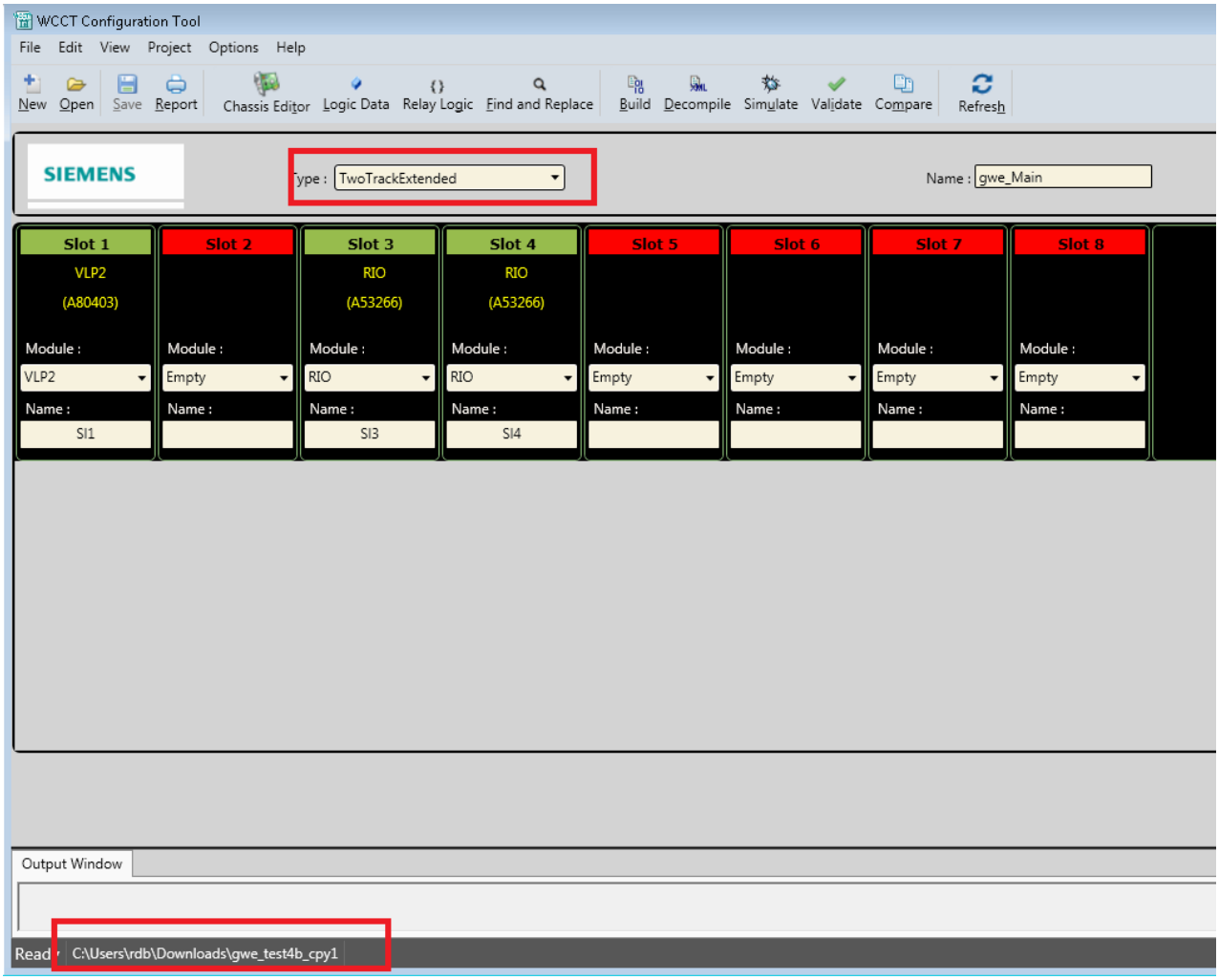


Figure E-7 Upgraded Project

If no is selected, the old chassis names are retained, e.g GEOTwoTrackExtended.

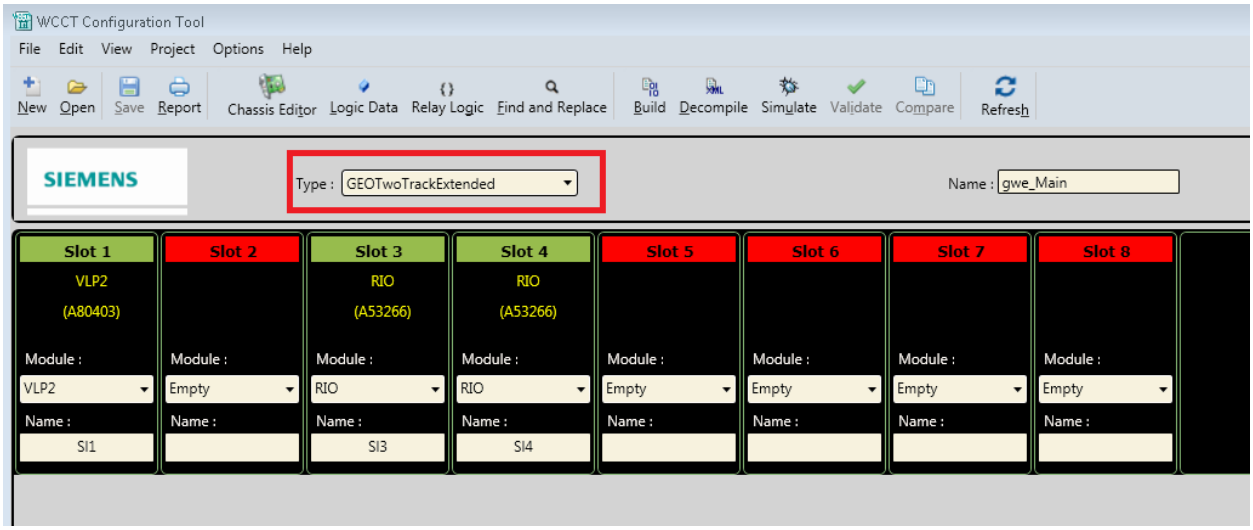


Figure E-8 Old Project

If a project of the same name already exists in the target workspace, the WCCT will not overwrite the existing one, but will append _cpy1, _cpy2 etc to the project name.

Once the project has been upgraded, select the **Project>Parse** to check for errors.

Some of the syntax rules for projects created with the older WCCT are not quite the same as the new WCCT. On opening the project, the WCCT will list any errors it finds in the output window, for example:

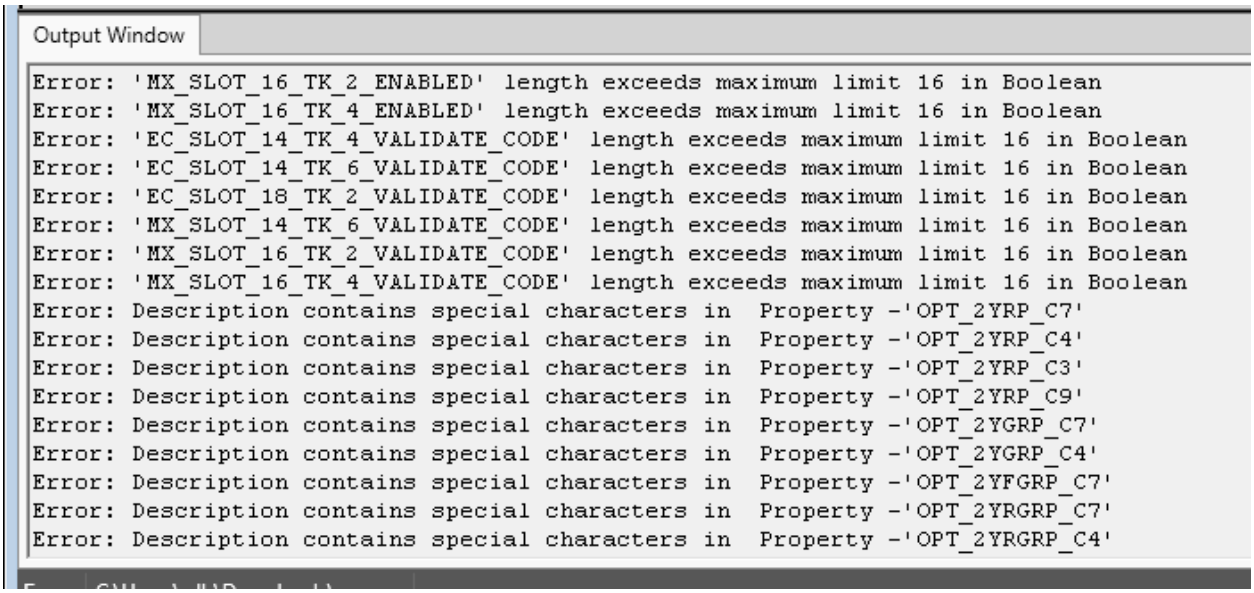


Figure E-9 Upgraded Project Error Messages

If variable names are too long, use **Find and Replace** to fix them.

It is recommended to go through each editor and do a **Save** to see where the consistency checks fail.