

# Trusted Toolset Suite

## Product Overview

This document describes the System Configuration Manager and the **IEC 61131 TOOLSET**, which is an enhanced version of the ISaGRAF Workstation. The toolset contains on-line documentation describing application design. This document includes the following:

- Installation
- System Configuration Manager
- Input / Output (I/O) Connection Editor
- Workstation Enhancements
- Intelligent Updates

The System Configuration Manager is used to manually create the System.INI file. This is a text file used to configure the operational parameters within a controller. The System Configuration Manager is an enhanced version of the System.INI File Transfer tool introduced with Version 1.2 of the Trusted® System.

The I/O Connection Editor is used establish a logical link between the I/O variables of the application and the physical channels of the boards existing on the target machine.

The operating parameters of the I/O modules which may be used with a Trusted System, as defined in Complex I/O definitions for the I/O Connection Editor, are defined in the appropriate I/O module or interface/bridge module PD.

## Features:

The Trusted Toolset Suite T8082 includes the following features:-

- IEC 61131 Trusted Toolset
- The System Configuration Manager
- The Trusted Application Validators
- The Regent+Plus I/O Module Drivers
- The Trusted I/O Module Drivers

The Trusted Application Validators can be installed from the CD. Details of which are contained within a separate product description; PD-T8015 - Application Validators.



Page intentionally left blank

## PREFACE

In no event will Rockwell Automation be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment. The examples given in this manual are included solely for illustrative purposes. Because of the many variables and requirements related to any particular installation, Rockwell Automation does not assume responsibility or reliability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, with respect to use of information, circuits, equipment, or software described in this manual.

All trademarks are acknowledged.

### DISCLAIMER

It is not intended that the information in this publication covers every possible detail about the construction, operation, or maintenance of a control system installation. You should also refer to your own local (or supplied) system safety manual, installation and operator/maintenance manuals.

### REVISION AND UPDATING POLICY

This document is based on information available at the time of its publication. The document contents are subject to change from time to time. The latest versions of the manuals are available at the Rockwell Automation Literature Library under "Product Information" information "Critical Process Control & Safety Systems".

### TRUSTED RELEASE

This technical manual applies to **Trusted Release: 3.6.1**

### LATEST PRODUCT INFORMATION

For the latest information about this product review the Product Notifications and Technical Notes issued by technical support. Product Notifications and product support are available at the Rockwell Automation Support Centre at <http://rockwellautomation.custhelp.com>

At the Search Knowledgebase tab select the option "By Product" then scroll down and select the Trusted product.

Some of the Answer ID's in the Knowledge Base require a TechConnect Support Contract. For more information about TechConnect Support Contract Access Level and Features please click on the following link:

[https://rockwellautomation.custhelp.com/app/answers/detail/a\\_id/50871](https://rockwellautomation.custhelp.com/app/answers/detail/a_id/50871)

This will get you to the login page where you must enter your login details.

---

**IMPORTANT** A login is required to access the link. If you do not have an account then you can create one using the "Sign Up" link at the top right of the web page.

---

**DOCUMENTATION FEEDBACK**

Your comments help us to write better user documentation. If you discover an error, or have a suggestion on how to make this publication better, send your comment to our technical support group at <http://rockwellautomation.custhelp.com>

## SCOPE

This manual specifies the maintenance requirements and describes the procedures to assist troubleshooting and maintenance of a Trusted system.

## WHO SHOULD USE THIS MANUAL

This manual is for plant maintenance personnel who are experienced in the operation and maintenance of electronic equipment and are trained to work with safety systems.

## SYMBOLS

In this manual we will use these notices to tell you about safety considerations.



**SHOCK HAZARD:** Identifies an electrical shock hazard. If a warning label is fitted, it can be on or inside the equipment.



**WARNING:** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which can cause injury or death, property damage or economic loss.



**ATTENTION:** Identifies information about practices or circumstances that can cause injury or death.



**CAUTION:** Identifies information about practices or circumstances that can cause property damage or economic loss.



**BURN HAZARD:** Identifies where a surface can reach dangerous temperatures. If a warning label is fitted, it can be on or inside the equipment.



This symbol identifies items which must be thought about and put in place when designing and assembling a Trusted controller for use in a Safety Instrumented Function (SIF). It appears extensively in the Trusted Safety Manual.

### IMPORTANT

Identifies information that is critical for successful application and understanding of the product.

### NOTE

Provides key information about the product or service.

### TIP

Tips give helpful information about using or setting up the equipment.

---

**WARNINGS AND CAUTIONS**

---

**WARNING: EXPLOSION RISK**

Do not connect or disconnect equipment while the circuit is live or unless the area is known to be free of ignitable concentrations or equivalent

---

**AVERTISSEMENT - RISQUE D'EXPLOSION**

Ne pas connecter ou déconnecter l'équipement alors qu'il est sous tension, sauf si l'environnement est exempt de concentrations inflammables ou équivalente

---

**MAINTENANCE**

Maintenance must be carried out only by qualified personnel. Failure to follow these instructions may result in personal injury.

---

**CAUTION: RADIO FREQUENCY INTERFERENCE**

Most electronic equipment is influenced by Radio Frequency Interference. Caution should be exercised with regard to the use of portable communications equipment around such equipment. Signs should be posted in the vicinity of the equipment cautioning against the use of portable communications equipment.

---

**CAUTION:**

The module PCBs contains static sensitive components. Static handling precautions must be observed. **DO NOT** touch exposed connector pins or attempt to dismantle a module.

---

**ISSUE RECORD**


<b>Issue</b>	<b>Date</b>	<b>Comments</b>
1.		
2.		
3.		
4.		
5.		
6.	Jul 04	
7.	Jan 06	Format/Release 3.5
8.	Aug 06	Corrections
9.	Sep 06	Template Editor
10.	Oct 06	Intelligent Updates
11.	Dec 06	No load threshold, password protections
12.	Mar 07	Detail of Intelligent online updates moved to AN-T80009
13.	Feb 08	Capabilities
14.	Feb 10	Note added about maximum expander chassis added
15.	Apr 16	Rebranded and updated to incorporate IEEE standards with correction of typographical errors.

Page intentionally left blank






---

# Table of Contents

<b>1.</b>	<b>Installation.....</b>	<b>10</b>
1.1.	Licensing .....	11
<b>2.</b>	<b>System Configuration Manager .....</b>	<b>14</b>
2.1.	Accessing the System Configuration Manager .....	15
2.2.	Menu Structure .....	16
2.2.1.	File Menu .....	16
2.2.2.	Template Menu.....	16
2.2.3.	Communications Menu.....	17
2.2.4.	Help Menu .....	17
<b>2.3.</b>	<b>Template Editor .....</b>	<b>18</b>
2.3.1.	Menu Structure .....	19
2.3.2.	New Templates .....	19
2.3.3.	Template Types.....	20
2.3.4.	Threshold Templates .....	22
2.3.5.	Threshold Template Editor.....	23
2.3.6.	LED Templates.....	24
2.3.7.	Module LED State Editor .....	26
2.3.8.	Channel LED State Editor .....	27
2.3.9.	Force Templates.....	28
2.3.10.	Module System Information Templates .....	29
2.3.11.	Module System Information Template Editor .....	31
2.3.12.	Module Flags Templates (Future Use Only).....	32
2.3.13.	Module Flags Template Editor .....	33
2.3.14.	Module Filter Templates (Future Use Only) .....	33
2.3.15.	Module Filter Template Editor.....	34
2.3.16.	Shutdown Templates .....	35
2.3.17.	Shutdown Template Editor .....	35
2.3.18.	Channel Type Templates.....	36
2.3.19.	Channel Type Template Editor.....	36
2.3.20.	De-energised Short Circuit Detection Templates .....	37
2.3.21.	De-energised Short Circuit Detection Template Editor .....	37

<b>2.4.</b>	<b>Configuring a SYSTEM .INI File .....</b>	<b>39</b>
<b>2.5.</b>	<b>Trusted TMR Processor .....</b>	<b>39</b>
<b>2.6.</b>	<b>Modules and Chassis .....</b>	<b>40</b>
<b>2.7.</b>	<b>Trusted TMR Communications Interface Module ( 8151 / 8151B ) .....</b>	<b>44</b>
<b>2.8.</b>	<b>Trusted I/O Modules.....</b>	<b>46</b>
2.8.1.	Scanning Controls: Pre Release 3.5.....	48
2.8.2.	Scanning Controls: From Release 3.5.....	48
<b>2.9.</b>	<b>Unused Slots .....</b>	<b>49</b>
<b>2.10.</b>	<b>Generating the SYSTEM.INI File .....</b>	<b>49</b>
<b>2.11.</b>	<b>Displaying the SYSTEM.INI File .....</b>	<b>50</b>
<b>2.12.</b>	<b>Example SYSTEM.INI File .....</b>	<b>52</b>
<b>2.13.</b>	<b>Generating the SYSTEM.INI File Diagram.....</b>	<b>56</b>
<b>2.14.</b>	<b>Downloading to the TMR Processor .....</b>	<b>57</b>
<b>2.15.</b>	<b>Uploading to the EWS .....</b>	<b>58</b>
<b>2.16.</b>	<b>Archiving .....</b>	<b>59</b>
2.16.1.	Save Configuration.....	59
2.16.2.	Delete Current Configuration .....	59
2.16.3.	Extract Configuration .....	59
2.16.4.	Archive Templates .....	59
2.16.5.	Restore Templates .....	59
<b>3.</b>	<b>The I/O Connection Editor .....</b>	<b>60</b>
<b>3.1.</b>	<b>Using the I/O Connection Editor .....</b>	<b>60</b>
<b>3.2.</b>	<b>Icons.....</b>	<b>61</b>
<b>3.3.</b>	<b>Defining I/O Boards .....</b>	<b>63</b>
3.3.1.	Selecting I/O board type  .....	63

---

3.3.2. Technical notes 	63
3.3.3. Removing a board 	63
3.3.4. Real boards and virtual boards 	63
<b>3.4. Setting Board Parameters</b>	<b>64</b>
<b>3.5. Connecting I/O Channels</b>	<b>64</b>
3.5.1. Connecting variables 	64
3.5.2. Removing connected variables 	65
3.5.3. Comments	65
<b>3.6. Directly Represented Variables</b>	<b>65</b>
<b>4. Trusted Board Definitions</b>	<b>67</b>
4.1. Process Historian	67
4.2. Sequence of Events	67
<b>5. Trusted workstation additions</b>	<b>68</b>
<b>6. ISaGRAF Workstation Enhancements</b>	<b>69</b>
6.1. Tooltips Enabled	69
6.2. Comment Variable Attachment	69
6.3. Power Flow Debugging	70
6.4. I/O Forcing	71
<b>6.5. Integer Interfacing Enhancements</b>	<b>72</b>
6.5.1. Write Protect	72
6.5.2. Registers as Unsigned Integer	72
6.5.3. Use Two Registers (32-bit)	72
6.5.4. Least Significant Word First	72

---

<b>6.6.</b>	<b>Printing</b> .....	<b>73</b>
<b>6.7.</b>	<b>Process Execution Timing Data</b> .....	<b>74</b>
<b>7.</b>	<b>Intelligent Online Updates</b> .....	<b>76</b>
<b>7.1.</b>	<b>Enabling Intelligent Online Updates</b> .....	<b>76</b>
<b>7.2.</b>	<b>Manual function block matching (from release 3.5.1)</b> .....	<b>78</b>
<b>7.3.</b>	<b>Operation</b> .....	<b>79</b>
<b>7.4.</b>	<b>Important Information</b> .....	<b>79</b>
<b>8.</b>	<b>ISA.INI settings</b> .....	<b>81</b>
<b>8.1.</b>	<b>Decimal format for Modbus addresses</b> .....	<b>81</b>
<b>8.2.</b>	<b>Configure an Internal variable or Constant with an Initial Value and the Retain attribute</b> .....	<b>81</b>
<b>8.3.</b>	<b>Change the documentation logo</b> .....	<b>82</b>
<b>9.</b>	<b>System Capabilities</b> .....	<b>83</b>
<b>10.</b>	<b>Password Permissions</b> .....	<b>86</b>

---

## Table of Figures

<b>Figure 1 Licensing Program.....</b>	<b>11</b>
<b>Figure 2 License Type.....</b>	<b>12</b>
<b>Figure 3 Email license request.....</b>	<b>12</b>
<b>Figure 4 System Configuration Manager Facilities.....</b>	<b>14</b>
<b>Figure 5 System Configuration Tool Opening Screen.....</b>	<b>15</b>
<b>Figure 6 Com Port Options Window.....</b>	<b>17</b>
<b>Figure 7 Template Editor Window.....</b>	<b>18</b>
<b>Figure 8 Template Editor Icon.....</b>	<b>18</b>
<b>Figure 9 Template Creation Window.....</b>	<b>19</b>
<b>Figure 10 Threshold Template Editor.....</b>	<b>23</b>
<b>Figure 11 Module LED State Editor.....</b>	<b>26</b>
<b>Figure 12 Channel LED State Editor.....</b>	<b>27</b>
<b>Figure 13 Force Template Editor.....</b>	<b>28</b>
<b>Figure 14 Module System Information Template Editor.....</b>	<b>31</b>
<b>Figure 15 Module Flags Template Editor.....</b>	<b>33</b>
<b>Figure 16 Module Filter Template Editor.....</b>	<b>34</b>
<b>Figure 17 Shutdown Template Editor (pre and post release 3.5).....</b>	<b>35</b>
<b>Figure 18 Channel Type Template Editor.....</b>	<b>37</b>
<b>Figure 19 De-energised Short Circuit Detection Template Editor.....</b>	<b>38</b>

---

<b>Figure 20 TMR Processor Configuration Window.....</b>	<b>39</b>
<b>Figure 21 'Replace Item in Slot' Window .....</b>	<b>40</b>
<b>Figure 22 'Insert New Chassis' Window .....</b>	<b>41</b>
<b>Figure 23 Configuration with one Expander Chassis.....</b>	<b>41</b>
<b>Figure 24 Chassis Connection Window .....</b>	<b>42</b>
<b>Figure 25 Expander Interface Parameters Window .....</b>	<b>42</b>
<b>Figure 26 Chassis Connection Window .....</b>	<b>43</b>
<b>Figure 27 Switch Configuration Window .....</b>	<b>43</b>
<b>Figure 28 Expander Interface Parameters Window .....</b>	<b>44</b>
<b>Figure 29 Communications Interface Parameters Window .....</b>	<b>45</b>
<b>Figure 30 'Replace Item in Slot' Window .....</b>	<b>46</b>
<b>Figure 31 Module Definition Window .....</b>	<b>47</b>
<b>Figure 32 Disable Scanning .....</b>	<b>49</b>
<b>Figure 33 INI Buffer Generation Icon .....</b>	<b>49</b>
<b>Figure 34 Version Control Information Editor Window .....</b>	<b>50</b>
<b>Figure 35 View INI Buffer Icon .....</b>	<b>50</b>
<b>Figure 36 INI Buffer Viewer .....</b>	<b>51</b>
<b>Figure 37 Regenerate Graphic from INI Buffer Icon.....</b>	<b>56</b>
<b>Figure 38 Updated Diagram .....</b>	<b>56</b>
<b>Figure 39 Download INI Buffer to MP Icon.....</b>	<b>57</b>
<b>Figure 40 Communications in Progress .....</b>	<b>57</b>

---

<b>Figure 41 Upload INI Buffer from MP Icon .....</b>	<b>58</b>
<b>Figure 42 Communication in Progress Window.....</b>	<b>58</b>
<b>Figure 43 I/O Connection Window.....</b>	<b>60</b>
<b>Figure 44 Connect I/O Channel Dialogue Box .....</b>	<b>64</b>
<b>Figure 45 Select Variable Dialogue .....</b>	<b>65</b>
<b>Figure 46 Enabling Tooltips.....</b>	<b>69</b>
<b>Figure 47 Referencing variable comments .....</b>	<b>69</b>
<b>Figure 48 Power Flow Debugging Off.....</b>	<b>70</b>
<b>Figure 49 Enable Power Flow Debugging .....</b>	<b>70</b>
<b>Figure 50 Power Flow Debugging On .....</b>	<b>70</b>
<b>Figure 51 Forced Data Point .....</b>	<b>71</b>
<b>Figure 52 Extended Attributes .....</b>	<b>72</b>
<b>Figure 53 Page Break Boundary .....</b>	<b>73</b>
<b>Figure 54 SYS_PE_TIMES time data ranges .....</b>	<b>75</b>
<b>Figure 55 Starting the Intelligent Update Manager.....</b>	<b>76</b>
<b>Figure 56 Intelligent Update Manager.....</b>	<b>77</b>
<b>Figure 57 Intelligent Update Options.....</b>	<b>78</b>

## Tables

<b>Table 1 Digital Input Threshold Diagram.....</b>	<b>22</b>
<b>Table 2 8403 Input Module LED Default Settings .....</b>	<b>24</b>
<b>Table 3 8451 Output Module LED Default Settings .....</b>	<b>25</b>
<b>Table 4 Analogue Input Module States .....</b>	<b>28</b>
<b>Table 5 OEM Parameters .....</b>	<b>67</b>

Page intentionally left blank

# 1. Installation

The Toolset Suite is validated for use on Windows NT and Windows XP service pack 2. It will not run on Windows Vista, 7, 8 or 10.

Please ensure that previous installations of this product are removed before attempting to load this product. If this product is being used to update an existing installation, all application files and registry entries will be maintained.

Place the CD in the CD ROM drive of the computer.

If auto-run has been enabled for the CD ROM Drive, the installation program will automatically start.

If auto-run has been disabled then run the program SETUP.EXE in the root folder of the CD.

The installation proceeds automatically apart from entry of user name.

Please note that the installation needs to be carried out with administrator privileges.

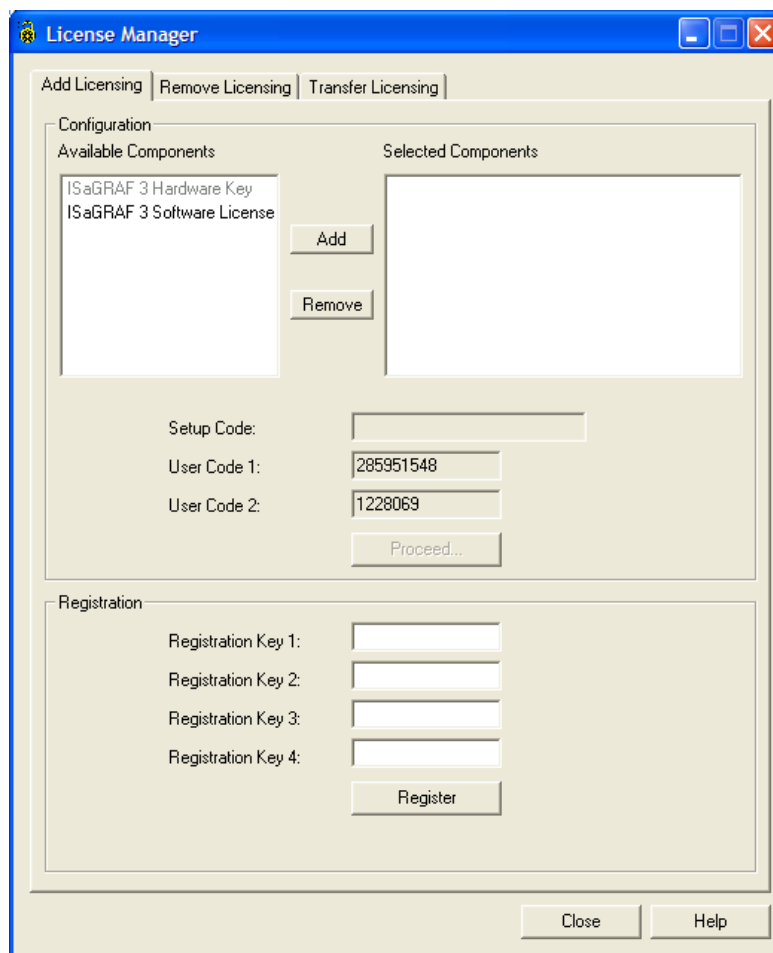
## 1.1. Licensing

Before release 3.5, the toolset requires a parallel port dongle to be fitted to the computer's parallel port. This must be present during the final stages of application compilation, when saving the I/O connection table and when downloading an application.

From release 3.5, there are three choices:

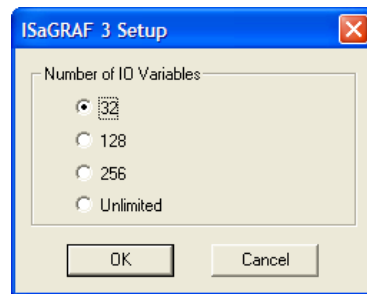
- An updated parallel port dongle, present at all times whilst the toolset is running (T8082P)
- A USB port dongle, present at all times whilst the toolset is running (T8082U)
- A software license, which will require the PC to have an Email client (T8082D)

A software license may be obtained using a separate program. From the Windows Start menu, select **All Programs | Trusted | Licensing**.



**Figure 1 Licensing Program**

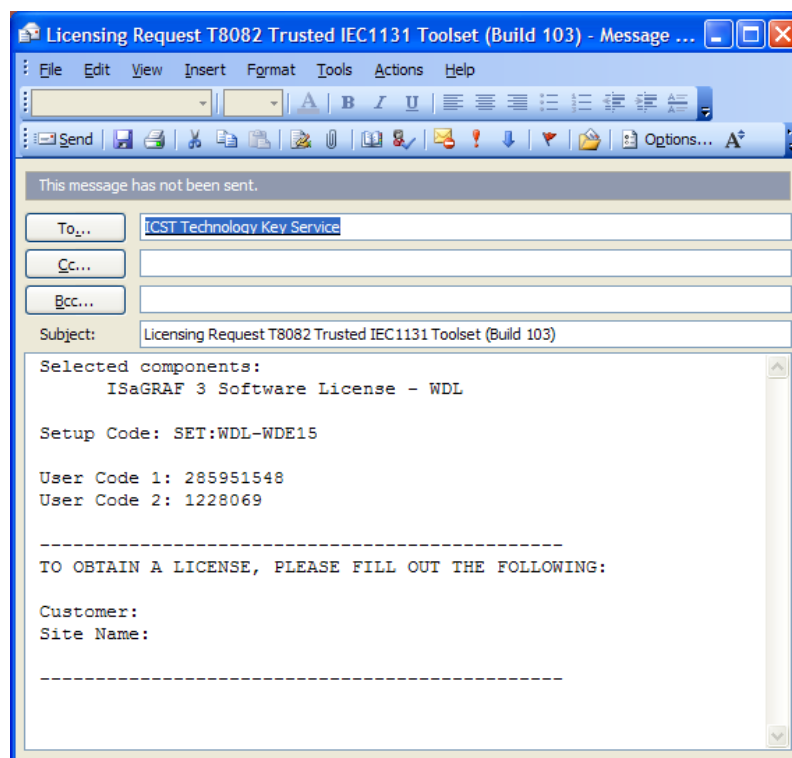
If a parallel or USB dongle is present, then 'ISaGRAF 3 Hardware Key' will not be grey. To request a license, click on 'ISaGRAF 3 Software License' and click **Add**.



**Figure 2 License Type**

Select the required license size. Most live systems will need an unlimited license. The count of I/O variables includes all those connected in the I/O Connection table. Click **OK**.

The appropriate setup code is now displayed. Click **Proceed** to request the license. The program requests permission to register by Email; if there is no Email client (e.g. Outlook) present on the computer, then either a hardware dongle is necessary, or the Email can be composed separately as shown below on another computer and sent to the Email address [keymaster@icstriplex.com](mailto:keymaster@icstriplex.com).



**Figure 3 Email license request**

Enter a company name and site name as shown. The Email should also include the serial number given on the Toolset packaging. Send the Email and close the licensing program. The request is processed manually.

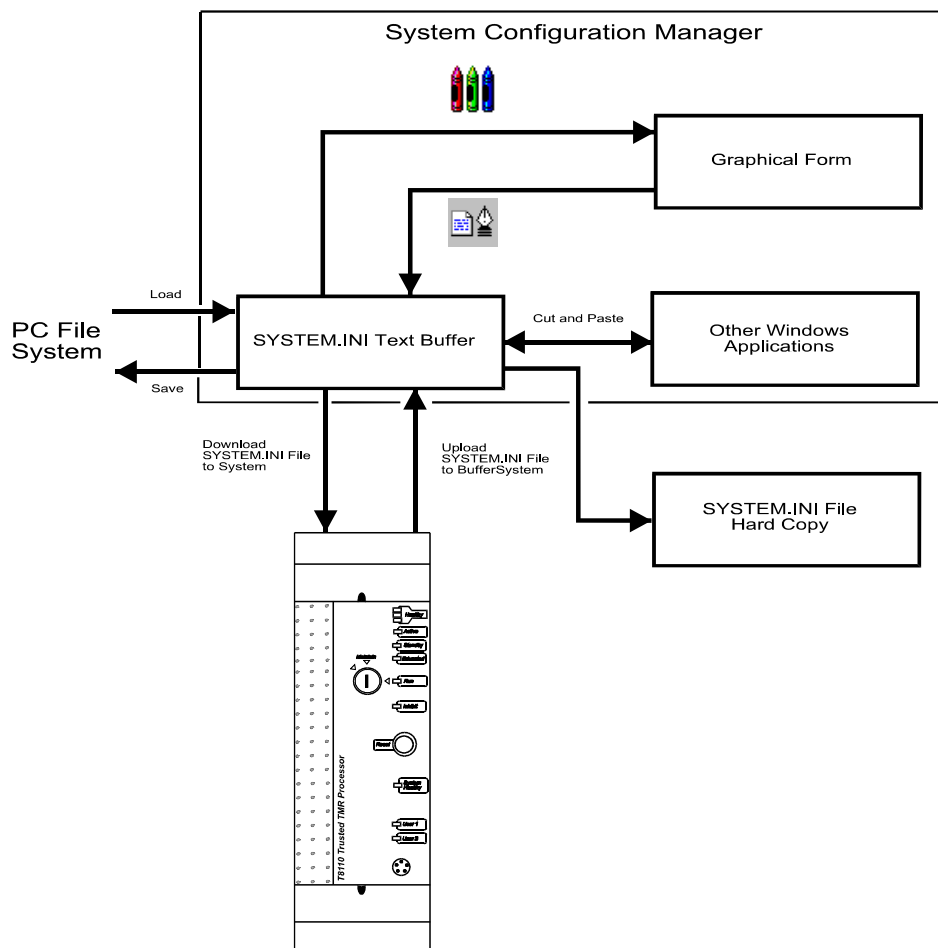
The reply will include two sets of four registration key numbers. Reopen the Licensing program. In the Registration area, enter the first four numbers and click Register. If successful, the number fields will clear. Enter the next four numbers and click Register. The license should now be successfully installed.

Other options in the Licensing program allow the surrendering of a license (**Remove Licensing**) through a similar procedure, or the transfer of a license to another computer; the online Help gives details on these processes.

**Note: If the Toolset is uninstalled and reinstalled, the license will be lost. Transfer the license to a removable disk before uninstalling.**

## 2. System Configuration Manager

The SYSTEM.INI file may be created manually, e.g. using 'Notepad', but it is best prepared with the aid of the System Configuration Manager. The facilities available via the System Configuration Manager are illustrated by Figure 4 below.

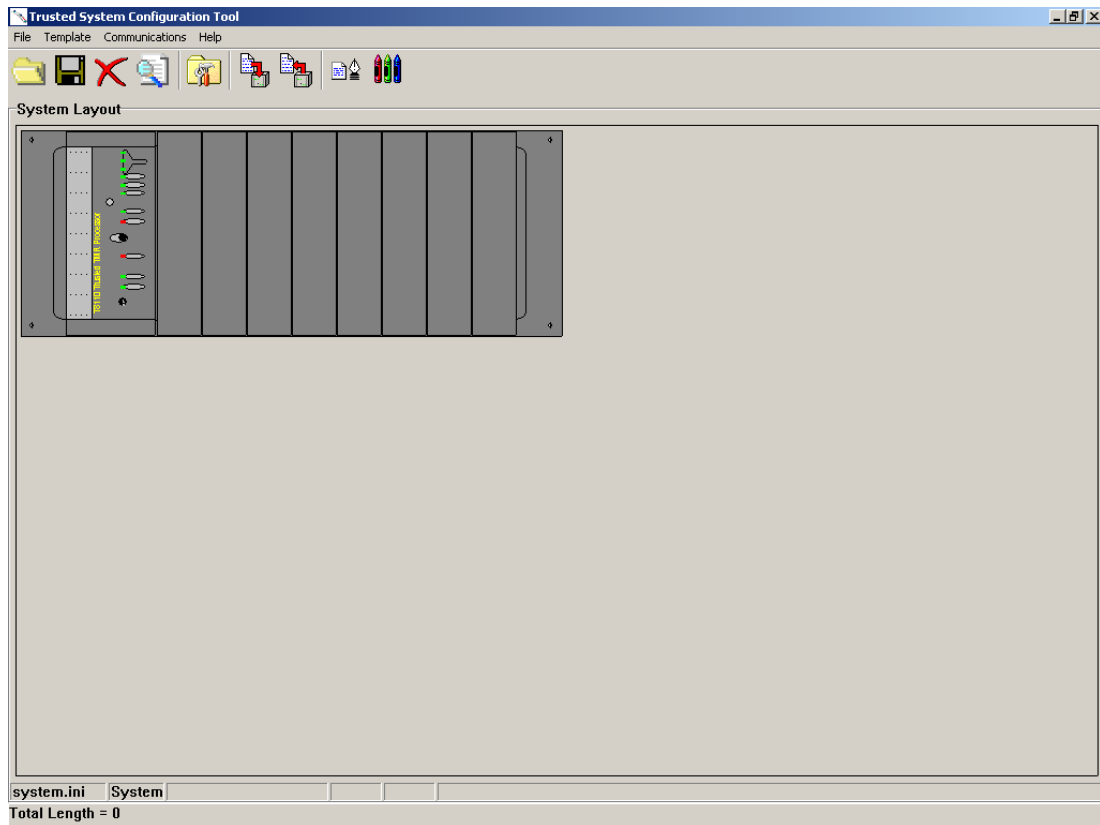


**Figure 4 System Configuration Manager Facilities**

SYSTEM.INI files may be uploaded to or downloaded from the Trusted TMR Processor using the System Configuration Manager. The Engineering Workstation (EWS) should be connected to the Trusted TMR Processor front panel diagnostic port using the Serial Maintenance Cable TC-304-01, or via Ethernet using an 8153 Communications Interface Adapter (on the back of the slot containing a Trusted Communications Interface), providing the Maintenance Enable keyswitch on the front panel of the Processor is set to the *Maintain* position.

## 2.1. Accessing the System Configuration Manager

The System Configuration Manager is used off-line to create/edit SYSTEM.INI files, and is accessed by selecting the **Tools / isa.mnu / System Config** in the Toolset program folder. The opening screen is shown in Figure 5 below.



**Figure 5 System Configuration Tool Opening Screen**

The opening screen shown above is used as an example only and depicts a single-chassis Trusted System housing a single TMR Processor. Only one Processor will be displayed even if two are fitted.

## 2.2. Menu Structure

### 2.2.1. File Menu

The **File** menu from the opening screen allows the user to implement the following:

1. Load the Configurator text buffer with data held within a file on the Workstation. Note that the buffer is separate from the diagram (graphical representation) of the system. The user is prompted to build the diagram when loading the buffer.
2. Save the text buffer to a file on the Workstation. Since the buffer and diagram are separate, the user is asked whether to save the existing buffer or rebuild it from the diagram.
3. Upload the System.INI file held within the Trusted System to the text buffer. The user is prompted to build the diagram when uploading the buffer.
4. Download the text buffer contents to the System.INI file held in the Trusted System. The user is prompted to specify communications settings (see section 2.2.3) if not already set, and also to verify that the buffer and diagram have been synchronised.

For example – to download the data held in a file called NEWSYS.INI to the System.INI, first load the NEWSYS.INI file into the text buffer and then download the text buffer to the Trusted System.

1. Erase the contents of the buffer.
2. Copy the contents of the buffer to the clipboard.
3. Print the contents of the buffer and displayed graphic diagram.
4. Import a configuration. This opens a Toolset application file ('appli.cnx', the connection table source code) and generates a graphical diagram of the Trusted System, representing modules for each module found in the application.

The user is prompted to search for the application folder, which will be similar to

..\Toolset\apl\

1. Note that this is only a first approximation to the configuration. Modules may be missing or incorrectly chosen, the modules will have no templates assigned and the diagram should be checked before continuing. This facility generates the diagram but does not create a buffer.

Exit from the System Configuration Manager.

### 2.2.2. Template Menu

**Templates** are segments of configuration. They can be viewed in text form, are referenced in the System.INI and can be assigned to more than one module. Each template performs a specific configuration of a Trusted module, such as trip thresholds and Light Emitting Diode (LED) states.

The **Template** menu contains one option - **Manage Templates**. This opens the Template Editor window. The Template Editor shows all templates stored in the 'Templates' folder in the Toolset directory. The 'Templates' folder is created automatically when the System Configuration Manager is first initiated.

The Template Editor is described in section 2.3 below.

### 2.2.3. Communications Menu

The **Communications** menu allows the user to configure the communications port on the Engineering Work Station (EWS) used to download the System.INI file to the TMR Processor. The **Communications** menu contains one option - **Configure Port**. This opens the Com Port Options window, shown in Figure 6 below.

#### Figure 6 Com Port Options Window

The EWS may be connected to the Trusted System either via a serial port to the front panel socket of the TMR processor (using Maintenance Cable TC-304-01) or via Ethernet to a Communications Module. Two tabs are provided to select these options.

On the 'Serial' tab, the Communications Ports' drop-down menu lists all available serial ports on the EWS. The remaining settings should be left at their defaults, unless the Trusted System communications configuration has been changed. Selecting a serial port defines that port as the communication channel, regardless of the settings on the 'Ethernet' tab.

If the Trusted System has previously been configured for Ethernet communications (using the System.INI configuration) then further communications may be made quicker by Ethernet. On the 'Ethernet' tab, enter the TCP/IP address assigned to the socket used on the communications module. Leave the port defined as 6000 unless changed in the System.INI configuration.

This options window is displayed if a System.INI file is downloaded without defining the communications method.

### 2.2.4. Help Menu

The **Help** menu provides the user with two options; **Introduction** and **About**.

Selecting **Introduction** provides the user with access to all the available online help screens. These are formatted as the normal HELP facilities found with any Windows based software package.

Selecting **About** provides the used with Version details of the System Configuration Manager.

## 2.3. Template Editor

Selecting the **Manage Templates** option in the **Template** menu displays the window shown in Figure 7.

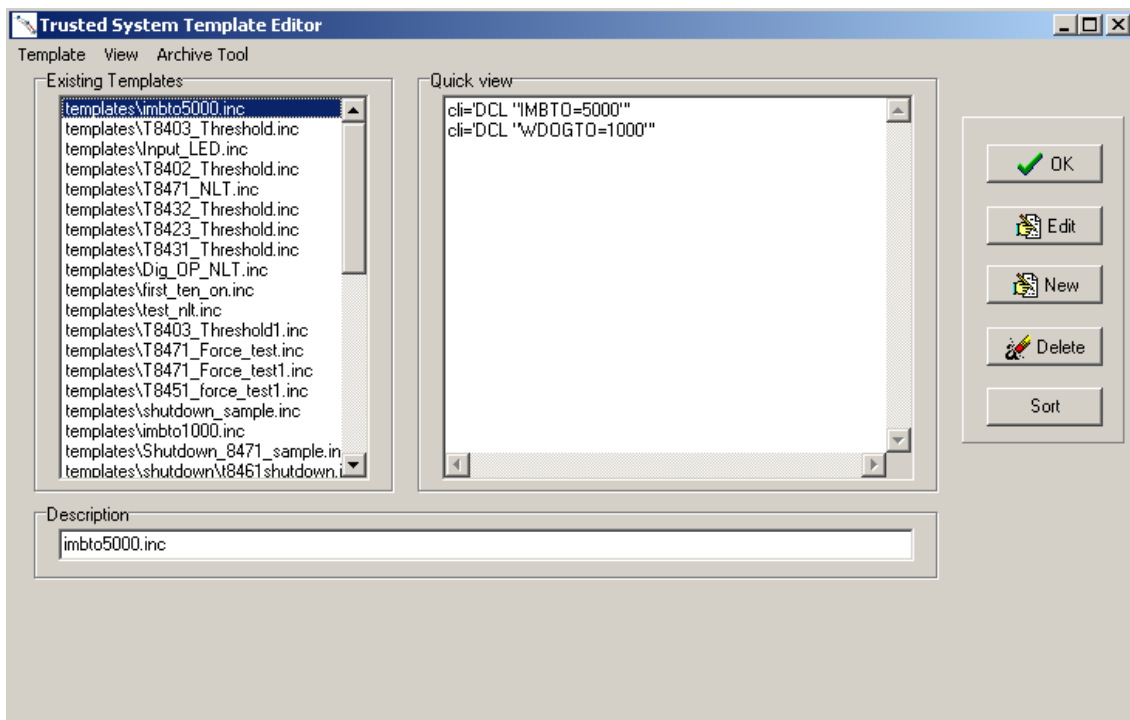


Figure 7 Template Editor Window

The **Manage Templates** option may also be initiated by selecting the command line button shown in Figure 8 below.



Figure 8 Template Editor Icon

### 2.3.1. Menu Structure

The **Template** menu allows the user to create new templates, edit existing templates and delete templates that are no longer required. These facilities are replicated by the buttons on the right-hand side of the display.

The **View** menu allows the user to sort existing templates in alphabetical order. This facility is replicated by a button on the right-hand side of the display.

The **Archive Tool** menu allows the user to save the template configuration, extract the template configuration, completely delete template configuration, archive templates and restore templates. These archives ensure that an entire System.INI configuration and associated templates are backed up correctly and available to transfer to another Engineering Workstation. Refer to section 2.16 for details. Note that from release 3.5, templates are extracted from the System.INI file itself, and so only the System.INI file needs to be archived. However, if the configuration needs to be transferred to a toolset issue before release 3.5, then a template archive is necessary.

All templates are listed in the **Existing Templates** window on the display. The contents of each template may be viewed by selecting the required template. This will cause the contents of the selected template to be displayed in the **Quick View** window of the display. Selecting **Edit** on the right hand side of the window will allow modifications to be made to this existing template.

### 2.3.2. New Templates

To create a new template from the Trusted System **Template Editor**, select the **New** button on the right-hand side of the display. The Template Creation window is displayed as shown in Figure 9 below.

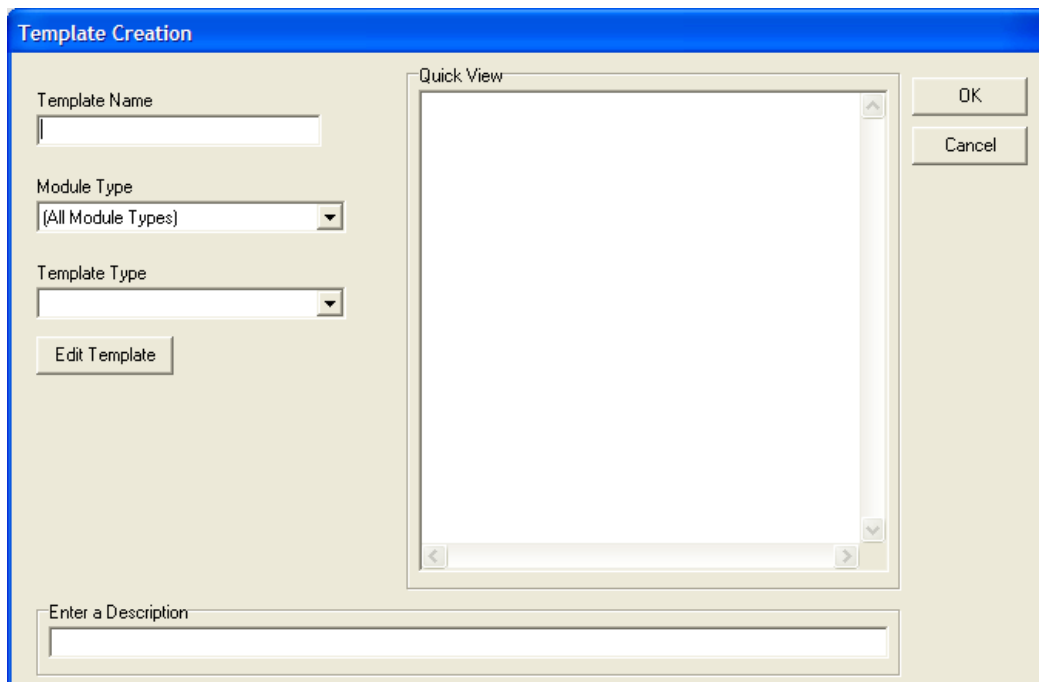


Figure 9 Template Creation Window

From the **Template Creation** window shown above, select the module type. Each template created can only be assigned to one module type. The available template types for that module are then available in the menu below. If '[All Module Types]' is selected, the templates available may be assigned to any module, but are limited to these portable types. Enter a name for the template and a description of its purpose or effect.

Select the **Edit Template** button on the window to design the template. This opens an editor appropriate to the template type.

When all values have been entered, select the **OK** button. This will return to the **Template Creation** window.

### 2.3.3. Template Types

There are ten types of template which may be created.

1. **Threshold** – defines the state of selected I/O module channels dependent upon the input signal to the channel.
2. **LED** – defines the colour and mode (flashing, steady or off) for channel LEDs for different channel states.
3. **Channel Forcing** – defines the state the unconnected channels must be forced into to provide a healthy (unalarmed) system state.
4. **Module System Information** – defines the specific timeout periods allowed for the module Watchdog and IMB (Inter-Modular-Bus).
5. **Module Flags** – defines the flag settings for the I/O module.
6. **Module Filter** – defines the internal filtering values for the I/O module.
7. **Additional** – allows the user to define additional Command Line Interface (CLI) entries that are external to the configuration data.
8. **Shutdown** – allows the user to define the output action required when the module enters the shutdown state (for example when the application is stopped).
9. **Channel Type** – allows the user to configure a channels operation (input to output) for the 8448 module.
10. **De-energised Short Circuit Detection** – allows the user to enable short circuit detection for de-energised outputs.

These are available for each module type as below.

	Threshold	LED	Force	System	Flags	Filter	Additional CLI	Shutdown	De-en SC	Special
8402	Yes		Yes	Yes	Yes	Yes	Yes			
8403	Yes	Yes	Yes	Yes	Yes	Yes	Yes			
8423	Yes	Yes	Yes	Yes	Yes	Yes	Yes			
8424	Yes	Yes	Yes	Yes	Yes	Yes	Yes			
8431	Yes	Yes	Yes	Yes	Yes	Yes	Yes			
8432	Yes		Yes	Yes	Yes	Yes	Yes			
8433	Yes	Yes	Yes	Yes	Yes	Yes	Yes			
8442		Yes		Yes	Yes		Yes			Speed monitor
8444		Yes	Yes	Yes	Yes		Yes	Yes		
8448	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Channel
8449	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	
8451		Yes	Yes	Yes	Yes		Yes	Yes	Yes	
8461		Yes	Yes	Yes	Yes		Yes	Yes	Yes	
8471		Yes	Yes	Yes	Yes		Yes	Yes	Yes	
8472		Yes	Yes	Yes	Yes		Yes	Yes	Yes	
8473		Yes	Yes	Yes	Yes		Yes	Yes	Yes	
8480		Yes	Yes	Yes	Yes		Yes	Yes		
All				Yes			Yes			

2.3.4. Threshold Templates

Input modules monitor and calculate the voltage level from the field at each channel to determine the appropriate state to report to the TMR Processor. After the module has calculated the input channel voltage, a state is then determined based on the channel threshold settings. There are 8 possible states (0 to 7). An example of these state thresholds is shown in Table 1 below.

Threshold	Description	State
Tmax	T <sub>max</sub>	The voltage is above the maximum for the module (set in manufacture). 6
	S/C	Short circuit. The voltage level indicates a short circuit field loop 5
T8		4 or 5
T7		4 or 5
	CC	Closed contact. 4
T6		3 or 4
T5		3 or 4
	IND	Indeterminate, this voltage indicates an error condition. 3
T4		2 or 3
T3		2 or 3
	OC	Open contact. 2
T2		1 or 2
T1		1 or 2
	O/C	Open circuit. The voltage level indicates an open circuit field loop condition. 1
Tmin	T <sub>min</sub>	The voltage is below the minimum for the module (set in manufacture). 0

**Table 1 Digital Input Threshold Diagram**

States 0 to 6 are based on the calculated voltage. One additional state (7) is reported when the module has completely failed. The description of the states shown above is in terms of a line monitored digital input switch configuration.

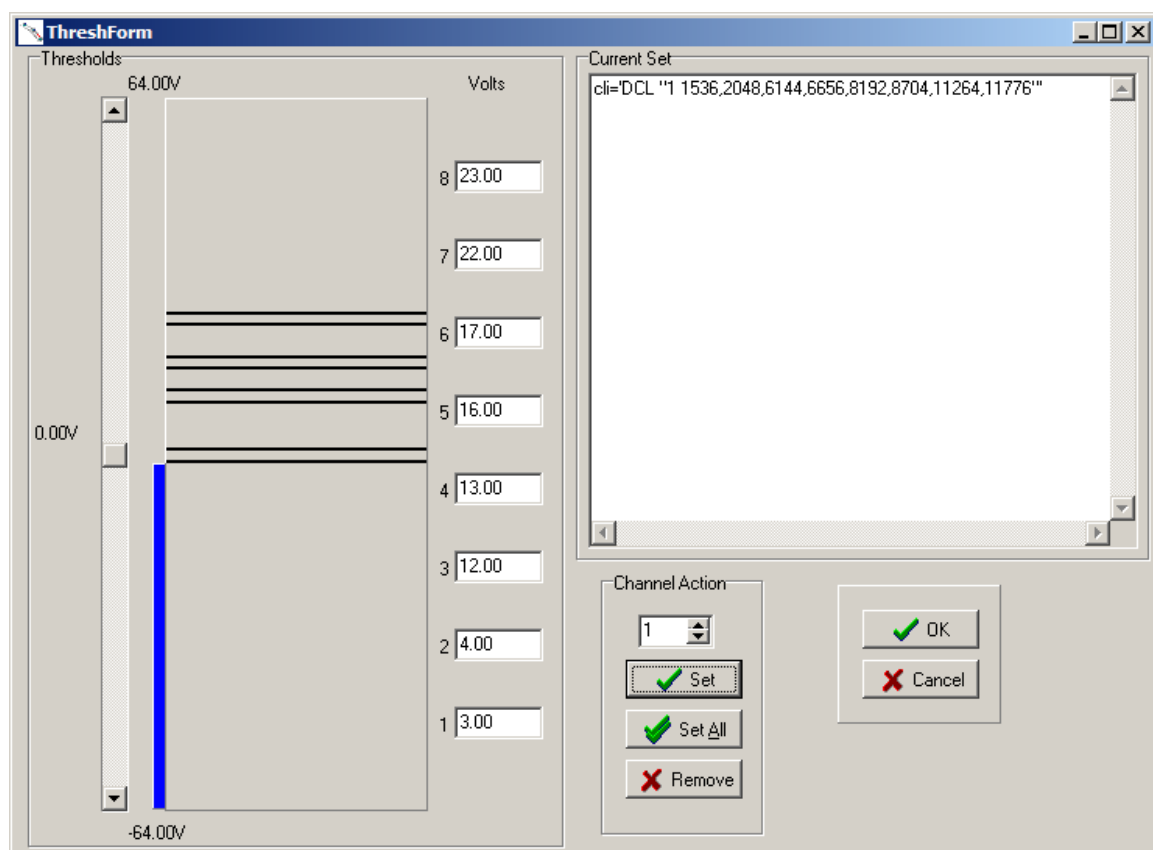
There are fixed minimum (Tmin) and maximum (Tmax) thresholds for the module, but each channel has eight configurable thresholds as illustrated above. Each state transition has a hysteresis shown by the grey areas in the diagram.

The rules for configuring thresholds are as follows:

- Each threshold must be greater than or equal to the previous threshold, i.e. T2 must be greater or equal to T1,  $T3 \geq T2$  etc...
- The state associated with each threshold is between the threshold settings, except that the lower threshold must be higher than Tmin (in the valid OC state range), and the upper threshold must be lower than Tmax (in the valid SC state range).
- Threshold values are defined as fractions of a volt. For example, the 8403 Digital Input Module, thresholds are represented as 512 counts per volt; e.g. a value of 8400 represents 16.406 volts (8400/512). The appropriate module Product Description defines the threshold fraction used.

### 2.3.5. Threshold Template Editor

The Threshold Template Editor window is shown in Figure 10 below.



**Figure 10 Threshold Template Editor**

There are two types of entries for this section: default settings that apply to all channels not individually configured, and individual channel settings.

The syntax for entering module default thresholds is:

**default = <T1>, <T2>, <T3>, <T4>, <T5>, <T6>, <T7>, <T8>**

The syntax for entering individual channel thresholds is:

**channel# = <default> | <T1>, <T2>, <T3>, <T4>, <T5>, <T6>, <T7>, <T8>**

If no entries are made to customize the thresholds, the module sets all channels to default settings. For example, in the 8403 Digital Input Module these are as follows:

**default = -2304,-1971,3046,3072,7168,8192,14848,15872**

The bar graph details the current voltage thresholds. These have been applied to Channel 1 of the I/O module in Figure 10 above by selecting the **Set** button (part of the **Channel Action** group of buttons). These values may be applied to all channels on the I/O module by selecting the **Set All** button. Alternatively, the user may select individual channels by selecting the channel number and then selecting the **Set** button.

The threshold values may be changed by dragging the horizontal lines on the bar graph. These values may also be changed by entering the voltage values in the windows of the **Volts** area. The vertical slider may be used to demonstrate the threshold settings.

Channels that have previously been **Set** but are no longer required may be deleted by selecting the **Remove** button.

### 2.3.6. LED Templates

## Input Modules

This template defines the mapping between the input state and the front panel indicators for the specific module. The mapping defines the required operation for all input channels. An LED mapping entry is made using the syntax shown below.

**state# = <OFF | RED | GREEN [+FLASH]>**

If no entries are made in this section, the module will operate with the default LED mapping. The example in Table 2 below shows defaults for the 8403 Digital Input Module, which are suitable for inputs without line monitoring devices installed.

0	= Red	– Out of range
1	= Off	– Open circuit
2	= Off	– Open field contact
3	= Off	– Indeterminate contact state
4	= Green	– Closed field contact
5	= Green	– Short circuit
6	= Green	– Overrange
7	= Off	–
8-	= Red	–

**Table 2 8403 Input Module LED Default Settings**

**Note:** States 8 through 15 all represent channel fault states.

LED templates define specific LED states. For example, if inputs are installed with line monitoring devices, the LED mapping entries can be made to flash red for the corresponding states that indicate a line fault condition, i.e. Open circuit, Indeterminate and Short circuit.

## Output Modules

This section defines the mapping between the output state and the front panel indicators for the specific module. The mapping defines the required operation for all output channels. An LED mapping entry is made using the syntax shown below.

**state# = <OFF | RED | GREEN [+FLASH]>**

If no entries are made in this section, the module will operate with the default LED mapping. The example in Table 3 below shows defaults for the 8451 Digital Output Module, which are suitable for outputs without line monitoring devices installed.

0	=	Off	–	Not used
1	=	Red	–	No field supply voltage
2	=	Off	–	Output de-energised (off)
3	=	Green	–	Line open circuit
4	=	Green	–	Output energised (on)
5	=	Red	–	Field short circuit
6	=	Off	–	Not used
7	=	Off	–	Not used
8-	=	Red	–	

**Table 3 8451 Output Module LED Default Settings**

**Note:** States 8 through 15 all represent channel fault states.

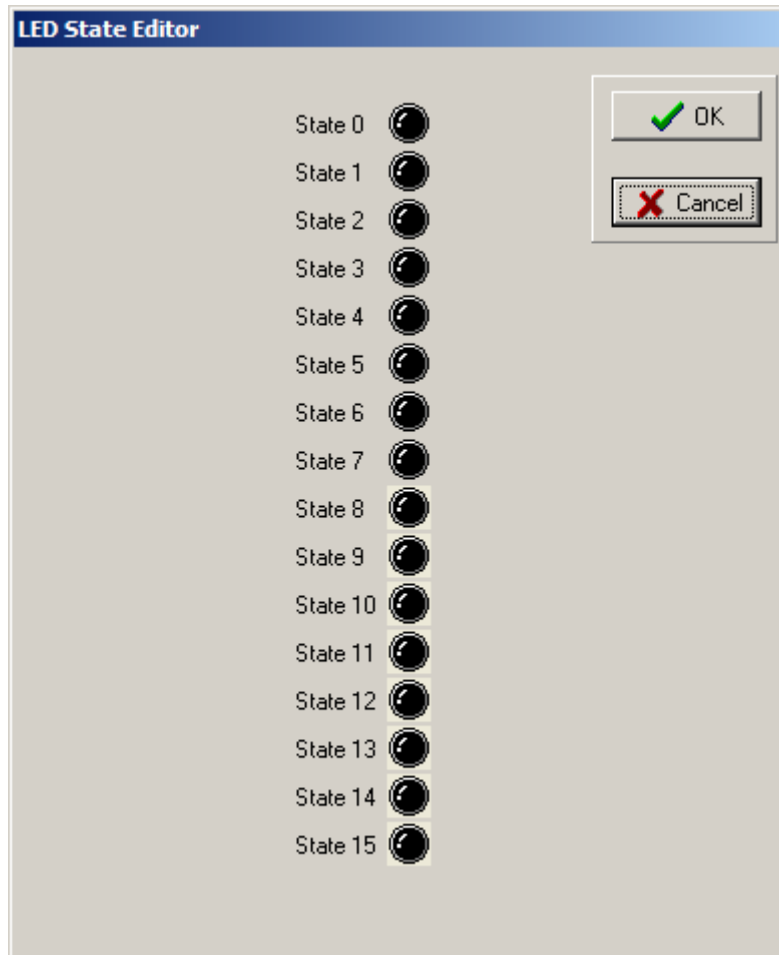
Input and Output Module LED templates are configured using the Module LED State Editor. On creating an LED template, select 'Module' in the LED Config section. The Module LED State Editor is described in section 2.3.7 below.

## Zone Interface Modules

Trusted 8448 Zone Interface Modules can have each channel defined as an input or an output using a Channel Template. In this case, one single LED definition for all channels is not appropriate. The LED states for this module are configured using the Channel LED State Editor. On creating an LED template, select 'Channel' in the LED Config section. The Channel LED State Editor is described in section 2.3.8 below.

## 2.3.7. Module LED State Editor

From the **Template Creation** window, select **LED Template** and select **Module** in the LED Config section. Select the **Edit** button. This will cause the Module **LED Editor** window to be displayed, as shown in Figure 11 below.



**Figure 11 Module LED State Editor**

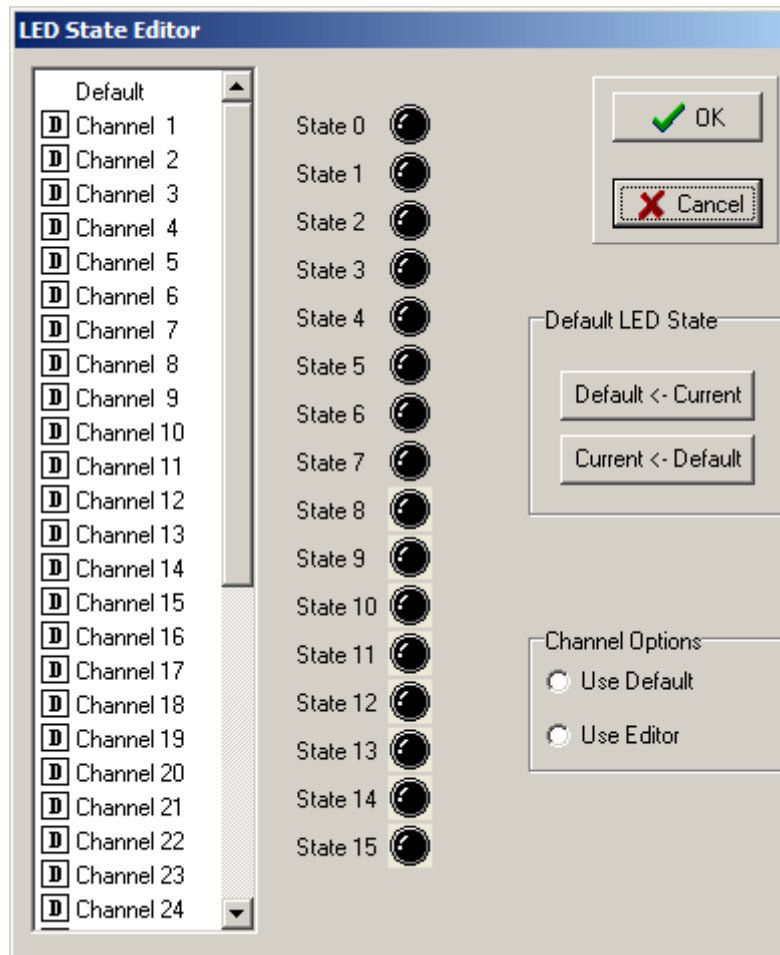
To change the colour and operating mode of the LED, point and click the left-hand mouse button on the required **State** LED icon. These changes will apply to all channel LEDs on the I/O modules that this template is applied to. The LED appearance for each state is then animated on the editor window.

When the correct LED states have been entered, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

Note that state 7 is a module fault. States 8 to 15 are not used for modules except the 8449 Valve Monitor Module. For this module, valve outputs and position inputs are grouped in pairs. States 0 to 7 are the output states 0 to 7 described in section 0 above. States 8 to 15 are the position input states 0 to 7 described in section 0 above.

### 2.3.8. Channel LED State Editor

From the **Template Creation** window, select **LED Template** and select **Channel** in the LED Config section. Select the **Edit** button. This will cause the Channel **LED Editor** window to be displayed, as shown in Figure 12 below.



**Figure 12 Channel LED State Editor**

Each channel can be individually configured with an LED state/colour map. To change the colour and operating mode of a channel LED, first select the channel to edit. Several channels may be selected at once using Ctrl-click or Shift-click. To change the LED appearance in a particular state, point and click the left-hand mouse button on the required **State** LED icon. The LED appearance for each state is then animated on the editor window. These changes will apply to the selected channel LEDs on the I/O modules that this template is applied to.

**D** The channels may also be assigned to the default LED state/colour map. Channels with a **D** symbol as shown left use the default map. To define whether a channel uses the default map or a specific map, select 'Use Default' or 'Use Editor' in the 'Channel Options' section. To edit the default map, select 'Default' at the top of the channel list and edit the LED colours.

A channel LED map may be assigned as the default by selecting the channel and selecting 'Default <- Current' in the 'Default LED State' section. A channel specific map may be changed to the current default by selecting the channel and selecting 'Current <- Default' in the 'Default LED State' section. The channel still has a specific LED map but it now matches the default.

When the correct LED states have been entered, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

### 2.3.9. Force Templates

Refer to the Trusted TMR System Safety Manual T8094; Publication [ICSTT-RM459-EN-P](#) (Legacy 555192).

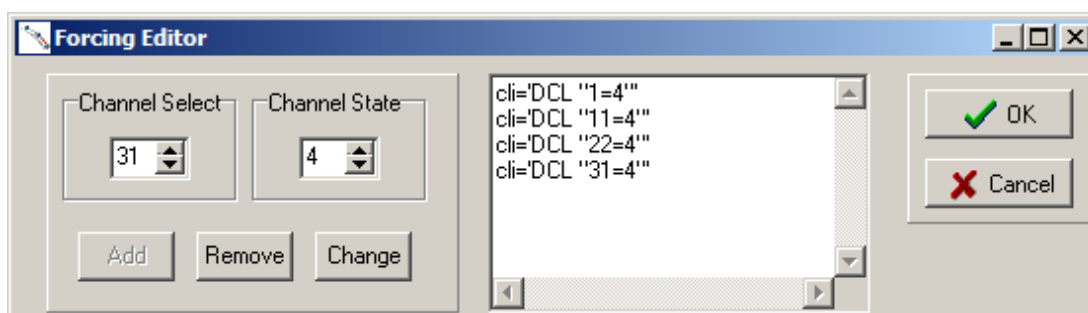
These templates allow forcing of the input or output channel states on-board the module to a specific value irrespective of the value that may be present. Channels may be forced to a specified state on a per-channel basis. The states that exist on a module are specific to that module. For example, channels on the 8431 Analogue Input Module may be any one of eight states detailed in Table 4 below.

0	-	Under range
1	-	Low-Low
2	-	Low
3	-	Normal
4	-	High
5	-	High-High
6	-	Over Range
7	-	Fail Safe

**Table 4 Analogue Input Module States**

### Force Template Editor

From the **Template Creation** window, select **Channel Forcing Template**, select the Module Group and Number of channels, then select the **Edit** button. This will cause the **Forcing Editor** window shown in Figure 13 below to be displayed.



**Figure 13 Force Template Editor**

The **Forcing Editor** window above shows Channels 1, 11, 22 and 31 forced to the 'High' state described in section 2.3.9 above. The LEDs displaying the channel status will be steady green (if set to the default colours) because they are indicating state 4.

To add a channel force to the template, the user must select the appropriate channel number in the **Channel Select** window, enter the required channel state in the **Channel State** window, then select the **Add** button.

To remove a channel from the template, the user must select the channel, then select the **Remove** button. A warning message is shown to ensure the correct force is removed.

To change the entered state for a channel, the user must select the appropriate channel, enter the new value in the **Channel State** window, then select the **Change** button.

When the correct channel and force states have been entered, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

### 2.3.10. Module System Information Templates

The typical entries for this template, include; NLthresh, IMBTO, WDOGTO, PWRFAILTO, and BYPASSTO. The syntax and function of each entry is defined below.

**nlthresh = < group1 >, < group2 >,< group3 >,< group4 > [,< group5 >]**

nlthresh ('no-load threshold') is a template that is used on output modules. It sets the current per slice (in milliamps) below which an output will signal state 3 (no load), and each of the parameters sets the threshold for an entire power group of eight digital outputs. Without this template, the defaults are 10mA, i.e. at least 30mA is required for the load to be recognised.

Outputs do not share current evenly across the three TMR slices and there may be a significant difference between slices. As an example, the 8451 output module has a minimum on state load current of 50 mA per channel. With a load of 51 mA, the load may be spread across the slices as follows:

Channel 1 Slice A 26 mA

Channel 1 Slice B 16 mA

Channel 1 Slice C 9 mA

Whilst the channel total is greater than the nominal minimum, slice C is below the default no-load threshold (10mA) and would therefore signal a no-load state whilst the other two slices will record an acceptable load. This causes a discrepancy alarm. In earlier firmware versions this may lead to a module shutdown.

It is recommended to group comparatively high and low power loads on different power groups. This prevents potential noise from a higher powered output affecting the lower powered signals. The following no-load threshold settings are recommended for each group:

25 to 80 mA	nlthresh = 5
80 to 150 mA	nlthresh = 10
150 to 300 mA	nlthresh = 15
300mA to limit	nlthresh = 20

As an example, if groups 1 and 2 (channels 1 to 16) are approximately 100 mA per channel and the rest are 50 mA per channel, the no-load thresholds should be set as:

**nlthresh = 10,10,5,5,5**

Output modules such as the 8471 (120Vdc) only have 4 output groups and the template is written, for example, as:

**nlthresh = 5, 5, 5, 5**

Whilst some output modules have a minimum on state load current of 20 or 25 mA per channel, it is still recommended to use the nlthresh template for these modules.

For the 8473, it is recommended to set nlthresh = 20, 20 because it will not accurately detect the no-load condition below this value.

**IMBTO = < value >**

The IMBTO (Inter-Modular Bus Timeout) is a timer run internally within each I/O module from start-up. If the TMR Processor does not scan the module within the programmed time, the module adopts the shutdown mode. In the majority of Trusted Systems, this entry may be omitted and the default used. In systems that are very large, or for test purposes, e.g. to allow the application program to be paused, it may be necessary to adjust the timeout period. The timeout period is defined in milliseconds; a value of zero disables the timeout.

**Note:** This value MUST NOT be set to zero for operational systems.

**WDOGTO = < value >**

The WDOGTO (System Watchdog Timeout) is a timer run internally within each I/O module from start-up. If the TMR Processor does not pet the module's watchdog within the programmed time, the module adopts the shutdown mode. However this parameter adjusts the system watchdog timeout interval. Again, it is not normally necessary to adjust this parameter, unless the system is very large or to allow certain tests to be performed. The timeout value is entered in milliseconds, and a value of zero disables the timeout.

**Note:** This value MUST NOT be set to zero for operational systems.

**PWRFAILTO = < value >**

The PWRFAILTO (Power Fail Timeout) adjusts the interval between the system detecting loss of both 24 Vdc power feeds to the module initiating its power-down sequence and fail-safe operation. Only in exceptional cases will it be necessary to adjust this setting. The timeout value is entered in milliseconds, and a value of zero disables the timeout.

**Note:** This value MUST NOT be set to zero for operational systems.

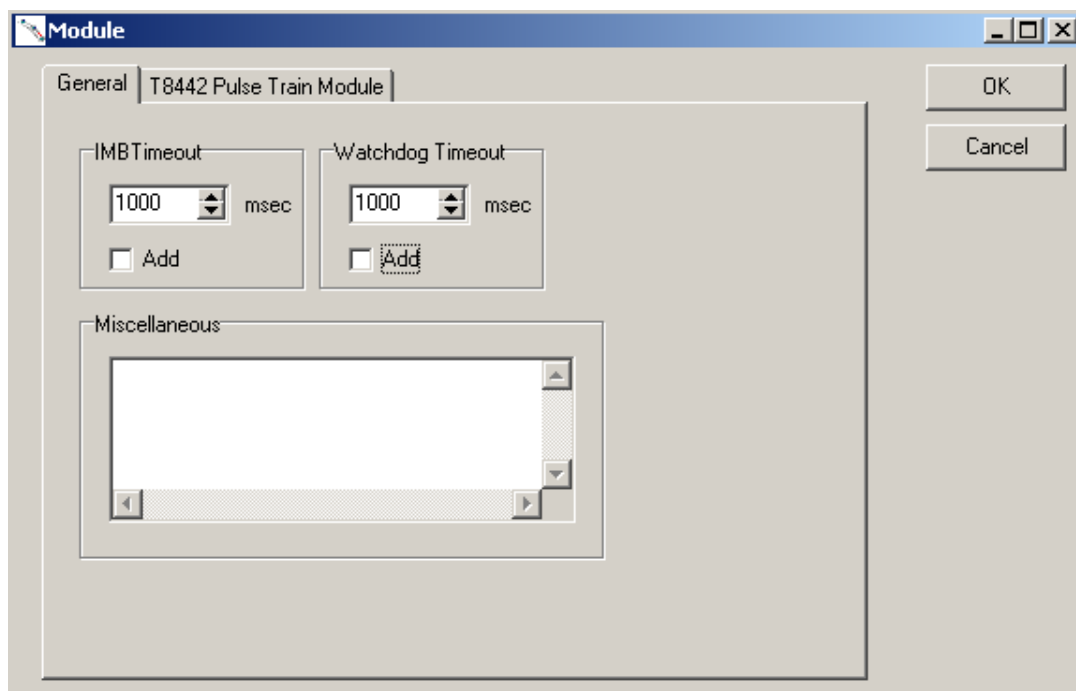
**BYPASSTO = < value >**

The BYPASSTO (Bypass Timeout) is used internally in the module to bypass the other timeouts during an Active/Standby changeover. Only in exceptional cases will it be necessary to adjust this setting. The timeout value is entered in milliseconds, and a value of zero disables the timeout.

**Note:** This value MUST NOT be set to zero for operational systems.

**2.3.11. Module System Information Template Editor**

From the **Template Creation** window, select **Module System Information**, then select the **Edit** button. This will cause the **Module** window shown in Figure 14 below to be displayed.



**Figure 14 Module System Information Template Editor**

The window above shows arbitrary values in the **IMB Timeout** and **Watchdog Timeout** windows. These may be changed and added by selecting **Add** as required.

It is recommended to set the above timeouts so that the module is polled at least four times at the normal scan time within the timeout, taking scheduled polling into account. For example, if the 'current' scan time is 200 milliseconds (as reported in the Toolset Debugger window), and the module is set for scheduled polling every 5<sup>th</sup> scan, then both timeouts (they operate in a similar way) need to be set to  $200 \times 5 \times 4 = 4000$  ms minimum. Ensure that the timeouts are still safe, i.e. less than half the process safety time.

The **Miscellaneous** window will normally be blank when initially editing this template. Timeout values for Power Fail and Bypass must be entered using this window, in the syntax shown in section 2.3.10.

When the correct timeout values have been entered, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

The Pulse Train Module is discontinued and its associated tab should not be used.

### 2.3.12. Module Flags Templates (Future Use Only)

#### Input Modules

There are two possible entries for this template; Currentsink and Ratiometric. The syntax and function of each entry is described below. These settings apply to the entire module.

##### **Currentsink = < TRUE | FALSE (default) >**

The input configuration option defines if the inputs are high or low side switching, i.e. switch to +Vdc or 0 Vdc (return). The default option is for high-side switch (to +Vdc). Setting this option to TRUE will select the switch operation to 0 Vdc (return).

##### **Ratiometric = < TRUE | FALSE (default) >**

This option allows the input thresholds to be defined either in terms of their absolute voltage or normalised to the nominal field supply voltage. The default operation is for the thresholds to be defined in absolute voltage.

When the ratiometric threshold operation is enabled the field supply must be connected to the monitoring terminals on the field termination unit. In ratiometric mode the range of voltages for the thresholds is 0 Vdc to 24 Vdc, the value will be re-scaled according to the current field supply voltage, e.g. a threshold value of 12 Vdc with an actual field supply of 30 Vdc will result in an actual threshold of 15 Vdc. If the field supply voltage is out of range, a line fault condition will be generated.

#### Output Modules

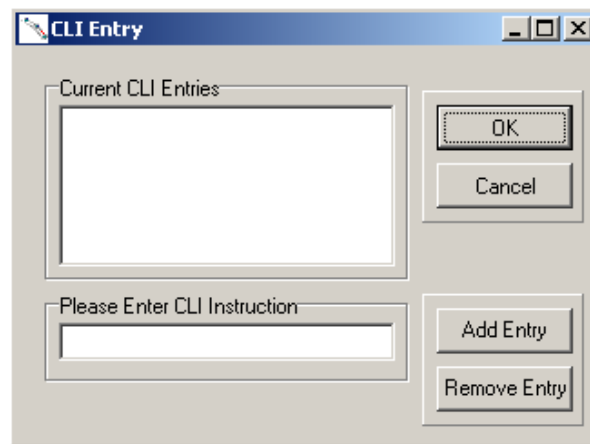
There is one possible entry for this template; LOGICAL. The syntax and function is described below. These settings apply to the entire module.

##### **LOGICAL = < TRUE | FALSE (default) >**

This configuration option defines how the module timestamps changes of state for the output channel. When set to True, the module will timestamp an output change when it receives a change of state to the logical output control signal. When set to False (default) the module will timestamp an output change of state when the measured voltage and current reflect an actual change of state in the output control circuit. Propagation delay between receiving the logic control signal and driving the output control signal is typically less than 1 millisecond. It is usually unnecessary to change this setting in most applications.

### 2.3.13. Module Flags Template Editor

From the **Template Creation** window, select **Module Flags**, then select the **Edit** button. This will cause the **Command Line Interface (CLI) Entry** window shown in Figure 15 below to be displayed.



**Figure 15 Module Flags Template Editor**

The **Current CLI Entries** area of the **CLI Entry** window will normally be blank when initially editing this template. If there are entries in this area, they may be deleted by highlighting them and selecting the **Remove Entry** button. Each entry must be deleted individually.

New instructions are entered in the **Please Enter CLI Instruction** area in the format shown above, then added to the template by selecting the **Add Entry** button.

When the correct instructions have been entered, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

### 2.3.14. Module Filter Templates (Future Use Only)

The Module Filter template is used to configure the filter type to use for each of the 40 channels on an input module. There are two types of entries for this template; default and channel settings.

The default entry sets the default filter type for all channels on the module. It has the following syntax.

**Default = < 0 | 1 | 2 | 3 >**

Input filtering for individual channels can be configured using the following

**channel# = < default > | < 0 | 1 | 2 | 3 >**

The module supports 4 input filter types, relating to the numbers in the syntax above.

0      4 ms filter

1      4 ms filter, same as 0

- 2 50 Hz notch filter
- 3 60 Hz notch filter

**Note:** 50 Hz and 60 Hz filters cannot be mixed on the same module.

#### 4 ms Filter

The samples are passed straight through with only a 4 ms signal averaging filter. The sampling frequency depends on whether another filter has been selected for any other channels. The default sampling frequency is 1.0 ms.

#### 50 Hz Filter (default)

This will filter out a 50 Hz signal in the sample data. The module sampling frequency is set to 1 ms, the filtering is performed by a moving boxcar average over 20 samples. The time is delayed for the channel by  $20 * 1.0 = 20$  ms to accommodate for the filter. This is the default line filter used.

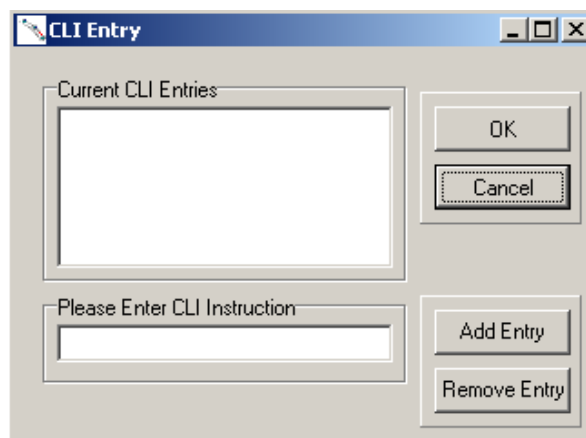
#### 60 Hz Filter

This will filter out a 60 Hz signal in the sample data. The module sampling frequency is set to 1.04 ms, the filtering is performed by a moving boxcar average over 16 samples. The time is delayed for the channel by  $16 * 1.04 = 16.64$  ms.

Since the propagation delays of the filters are a known constant, when a change of input state passes through the filter, the corresponding Sequence of Events (SOE) event time stamp is adjusted by the propagation delay.

#### 2.3.15. Module Filter Template Editor

From the **Template Creation** window, select **Module Filter**, then select the **Edit** button. This will cause the **CLI Entry** window shown in Figure 16 below to be displayed.



**Figure 16 Module Filter Template Editor**

The **Current CLI Entries** area of the **CLI Entry** window will normally be blank when initially editing this template. If there are entries in this area, they may be deleted by highlighting them and selecting the **Remove Entry** button. Each entry must be deleted individually.

New instructions are entered in the **Please Enter CLI Instruction** area in the format shown above, then added to the template by selecting the **Add Entry** button.

When the correct instructions have been entered, the user must select the OK button. This will return the user to the Template Creation window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

### 2.3.16. Shutdown Templates

This template is used to configure individual shutdown states for each channel of an output module. Output channels are driven to the default, fail-safe, state when the module enters the shutdown mode. Typically the module will enter the shutdown mode when any of the system timeouts occur, or when commanded by the TMR Processor. A channel may be configured to enter a specific shutdown state other than the default using this template.

A shutdown entry is made using the syntax shown below.

channel#=< ON | OFF | HOLD >

The three available shutdown modes are:

- ON: Energize the output
- OFF : De-energise the output
- HOLD: Hold, or maintain the output in the last commanded state

**Note:** All output channels used for safety critical applications must be configured with shutdown mode that places the process in the safe state.

### 2.3.17. Shutdown Template Editor

From the **Template Creation** window, select **Shutdown (Output Modules Only)**, then select the **Edit** button. This will cause the **Shutdown Editor** window shown in Figure 17 below to be displayed.

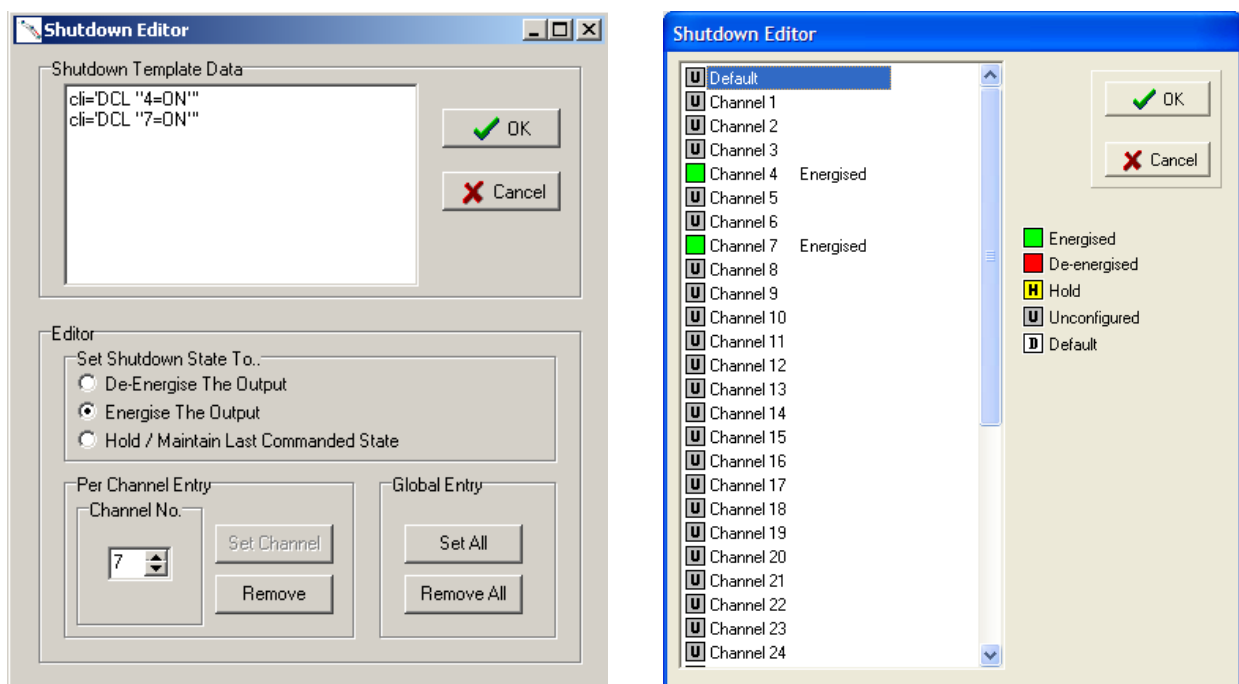


Figure 17 Shutdown Template Editor (pre and post release 3.5)

For earlier versions, the **Shutdown Template Data** area of the **Shutdown Editor** window will normally be blank when initially editing this template. If there are entries in this area, they may be deleted by highlighting them and selecting the **Remove** button to delete individual entries, or the **Remove All** button to globally delete entries. For later versions, all channels are unassigned by default.

The template is configured using the options and facilities available in the **Editor** area of the **Shutdown Editor** window (earlier versions) or by right clicking on each channel (later versions). The above displays show that channels 4 and 7 have been set to the **Energise The Output** option. All module outputs may be forced to a pre-determined state by selecting the **Set All** button in the **Global Entry** area.

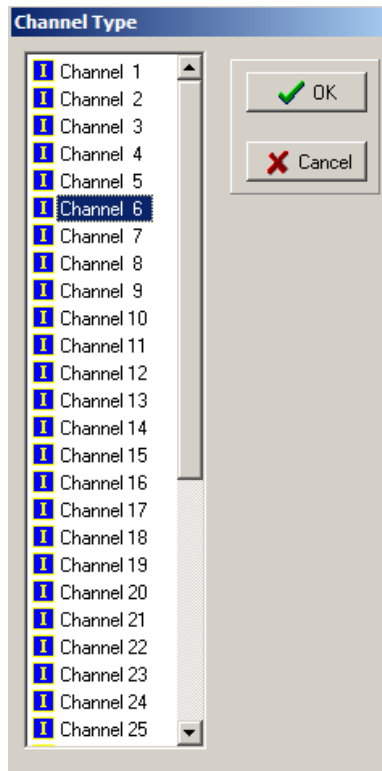
When the correct option and channels have been selected, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

#### 2.3.18. Channel Type Templates

This template defines the Trusted Zone Interface Module channel configuration. Each channel on a Zone Interface Module can either be an input or output. The default configuration is with all channels set as inputs.

#### 2.3.19. Channel Type Template Editor

From the **Template Creation** window, select **Channel Type (ZIM Only)**. Enter a description for the template, then select the **Edit** button. This will cause the **Channel Type Editor** window shown below to be displayed as shown in Figure 18 below.



**Figure 18 Channel Type Template Editor**

To change the channel type, point and click right-hand mouse button on the channel required to get the Type selection menu. Click on the desired channel type. The icon in the channel list will show I or O for input or output as requested.

When the correct channel Types have been entered, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

#### 2.3.20. De-energised Short Circuit Detection Templates

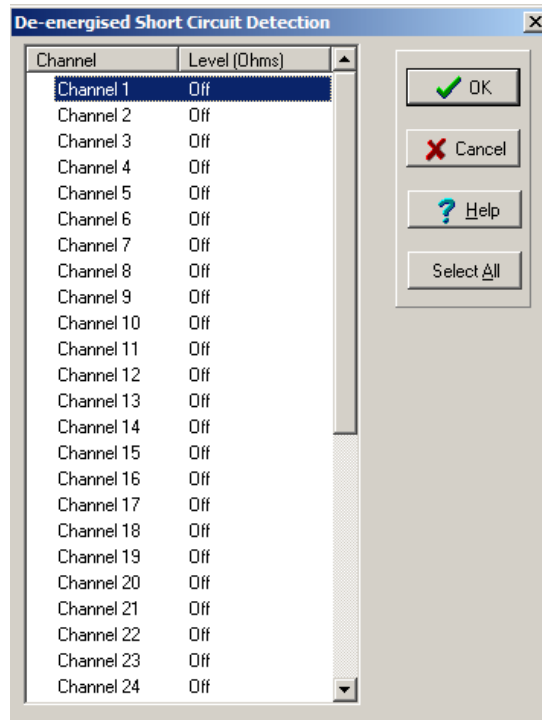
Output modules by default do not detect for short circuit faults when de-energised. Using this template, they may be configured to detect de-energised short circuit faults on some or all channels. The impedance below which a de-energised short circuit fault is reported is 40  $\Omega$ . Note that the cold resistance of small incandescent lamps may be less than this threshold.

Note that de-energised short circuit templates will not work on older 5V HIU hardware builds. An attempt to swap from a 3.3V module with this template, to a 5V module, will fail.

#### 2.3.21. De-energised Short Circuit Detection Template Editor

From the **Template Creation** window, select the **De-energised Short Circuit Detection** option, module voltage and number of channels. Enter a description for the template. Select the **Edit** button

on the display. This will cause the De-Energised Short Circuit Detection editor window to be displayed as shown in Figure 19 below.



**Figure 19 De-energised Short Circuit Detection Template Editor**

To change the channel line monitoring, point and click right-hand mouse button on the channel required to get the line monitor selection menu. Select either **40** or **off**. Selecting **Select All** (or Ctrl+A or the **Select All** icon) allows all channels to be set together.

When the correct channel selections have been entered, the user must select the **OK** button. This will return the user to the **Template Creation** window which permits the newly created template to be saved. The filename must not include any spaces. Underscores are recommended.

## 2.4. Configuring a SYSTEM .INI File

Once the user has configured all required templates for the Trusted System, the communications port on the EWS has been configured and communications with the Processor are open, the user may configure/edit the System.INI file as necessary. For the purpose of clarity, this Product Description will describe the full configuration of the System.INI file with the basic module tray as shown in the opening screen.

## 2.5. Trusted TMR Processor

Configuration of the Trusted TMR Processor is necessary at initial start-up of the Trusted System. This is achieved by clicking the left-hand mouse button on the Processor module shown on the opening screen. This will cause the dialogue box shown in Figure 20 below to appear.

Serial	Baud	Data Bits	Parity	Stop Bits	Line	Protocols
Serial 1	19200	8	Odd	1	rs232	
Serial 2	19200	8	Odd	1	rs485dmux	Modbus
Serial 3	19200	8	Odd	1	rs485dmux	

**Figure 20 TMR Processor Configuration Window**

The values shown in the display above are the defaults for the Trusted TMR Processor. The user must configure the Processor to suit the requirements of the Trusted System it is being used in. The configuration of the parameters in each of the above areas is described separately in PD-8110B.

## 2.6. Modules and Chassis

The next step in configuring the System.INI file requires the user to assign modules to empty slots and introduce Expander Chassis. The user must first assign Trusted Expander Interface and Communications Interface Modules to the appropriate slots currently shown as empty in the Controller Chassis. This is achieved by placing the mouse cursor on the slot then clicking the left-hand mouse button. This will cause the **Replace Item In Slot** window shown in Figure 21 below to be displayed.

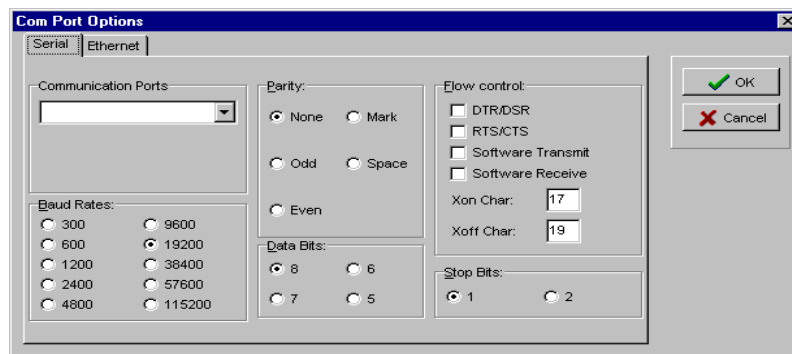


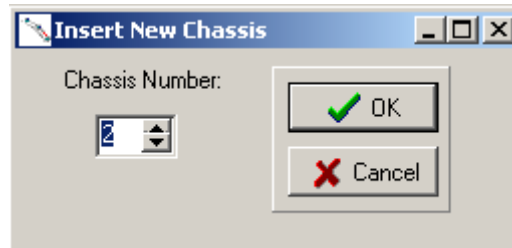
Figure 21 'Replace Item in Slot' Window

The window shown in Figure 21 above shows the Trusted modules which may be assigned to a Controller Chassis.

Modules are assigned to the empty slots by selecting the appropriate module type then selecting the **OK** button. This process must be repeated until all required modules have been assigned to the correct slots.

**Note:** Whilst Trusted Expander Interface Modules may be placed in any slot position within the Controller Chassis, it is recommended that they are assigned to Slots 1 and 2. If fitted elsewhere, the left-hand module must be in an odd numbered slot. Two Expander Interface Modules will need to be specified in the Primary and Secondary slots, to enable hot swap operation.

Expander Chassis are added to the System.INI file by placing the cursor on a blank area of the display and pressing the right-hand mouse button. This will initiate the **Insert New Chassis** window shown in Figure 22 below to be displayed.

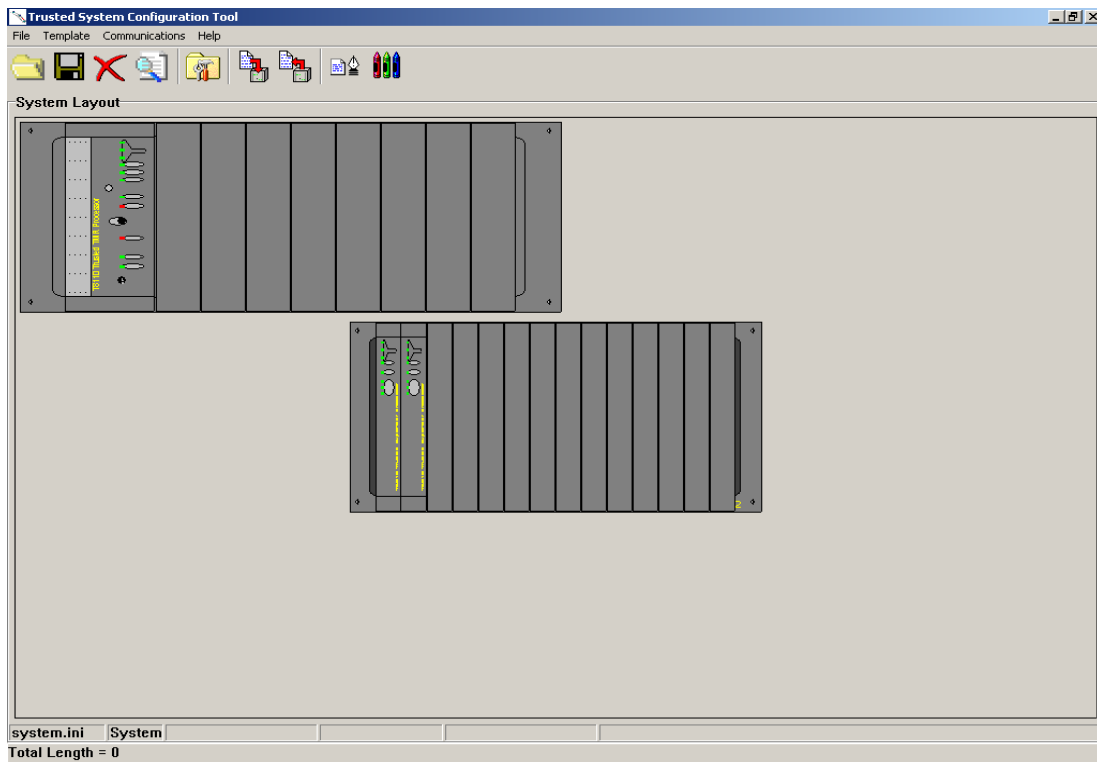


**Figure 22 'Insert New Chassis' Window**

The user must enter the number of the first Expander Chassis, e.g. 2 (but no more than 7), then select the **OK** button. This must be repeated until all required chassis have been added to the display. The number of chassis added must not exceed the maximum number detailed when the parameters of the Processor were entered previously. For the purpose of this exercise, the number entered was two which included the Controller Chassis.

**NOTE:** The maximum number of expander chassis that may be added is 27 making a total of 28 chassis in a configuration file (1 processor chassis and 27 expander chassis).

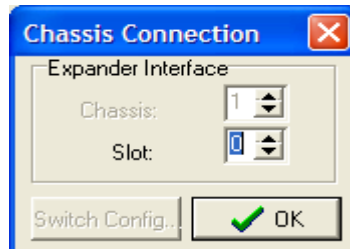
Once all required chassis have been added, the screen display will appear as shown in Figure 23 below.



**Figure 23 Configuration with one Expander Chassis**

At this point, it is necessary to ‘connect’ the Expander Chassis to the ‘master’ Trusted Expander Interface Module in the Controller Chassis. By default, this will be the module in Slot 1 (left-hand module). Connection is achieved by placing the cursor on the left-hand edge of the Expander

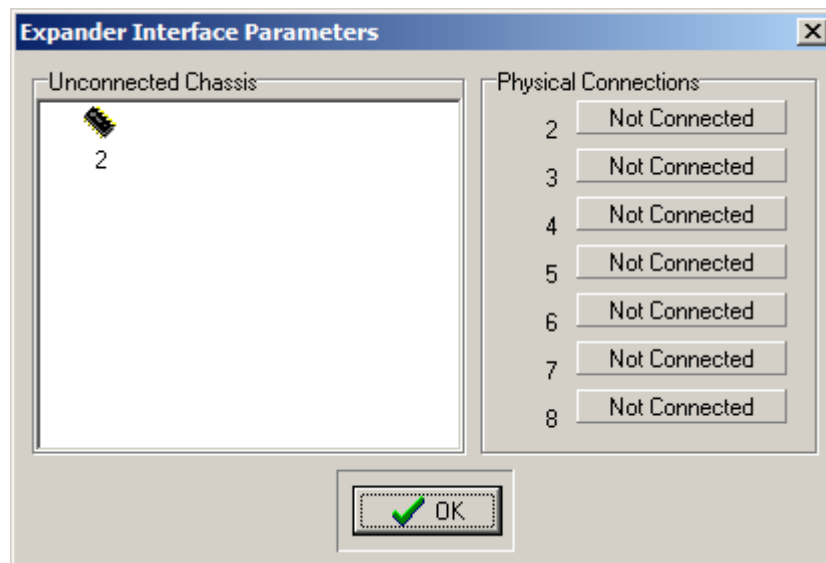
Chassis shown as Chassis 2, then pressing the left-hand mouse button. This will cause the **Chassis Connection** window shown in Figure 24 to be displayed.



**Figure 24 Chassis Connection Window**

The user should enter the slot number appropriate to the 'master' Trusted Expander Interface Module in the Controller Chassis, i.e. 1, then select the **OK** button. This connects the Expander Chassis to the 'master' Trusted Expander Interface Module. If no Expander Interface Module has been defined at this point, one will be created automatically at the referenced position.

It is also possible to connect expander chassis to a Trusted Expander Interface Module by selecting the Expander Interface Module with the left mouse button. The window shown in Figure 25 below is displayed.



**Figure 25 Expander Interface Parameters Window**

To connect an expander chassis to a channel on the Expander Interface, left-click and drag the 'chip' icon in the **Unconnected Chassis** window to the appropriate channel in the **Physical Connections** area. The channel will then show the number of the connected chassis. Repeat for the other unconnected chassis. To disconnect a chassis, left click on the channel and drag the icon back to the **Unconnected Chassis** window. Click **OK** when complete.

Note that the connected Expander Chassis are now marked by a light blue 'lightning flash'. This indicates that the Expander Chassis is connected.

The **Chassis** Connection window also shows the switch configuration necessary for each expander chassis. Place the cursor on the left-hand edge of Expander Chassis again, then press the left-hand button to cause the **Chassis Connection** window shown in Figure 26 below to be displayed again.

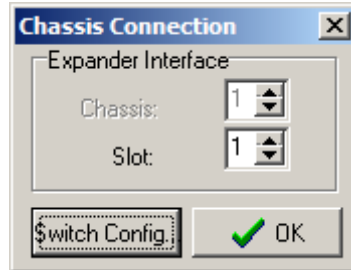


Figure 26 Chassis Connection Window

Once a Trusted Expander Chassis has been connected, the **Switch Config** button on the **Chassis Connection** window becomes active. Selecting this button will cause the **Switch Configuration** window shown in Figure 27 below to be displayed.

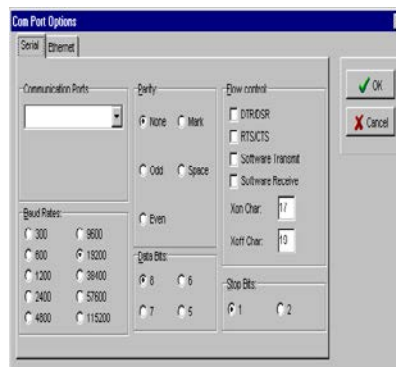


Figure 27 Switch Configuration Window

This window shows the positions that the three DIP switches on the rear of the Expander Chassis should be set in for Chassis 2. The first expansion chassis is always ID 2, the second ID 3 and so on. If the switches are not set in the correct position, the Expander comms link will not function. Please refer to product description PD-8300 Trusted TMR Expander Chassis for DIP switch details.

All Expander Chassis must be 'connected' to the 'master' Trusted Expander Interface Module as described above. Confirmation that all chassis are connected may be obtained by placing the cursor on the module in Slot 1 of the Controller Chassis and pressing the left-hand mouse button. This will initiate the display shown in Figure 28 below.

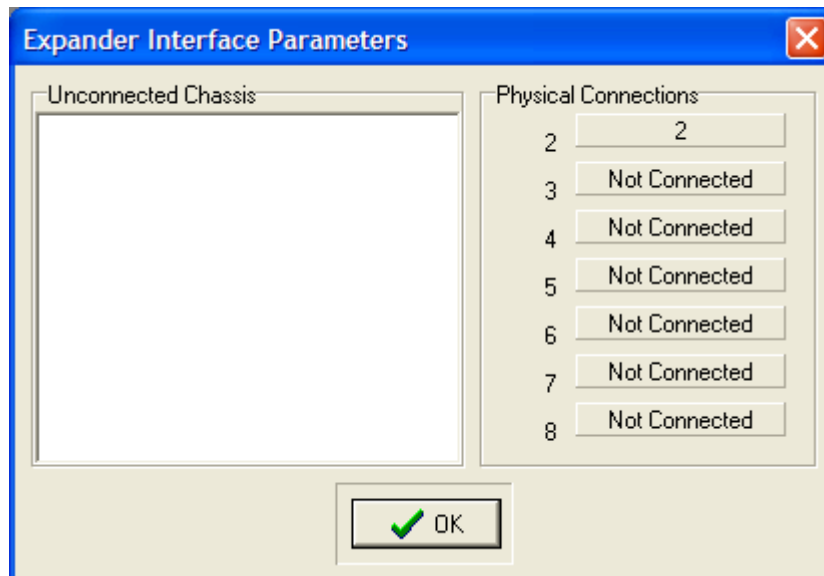
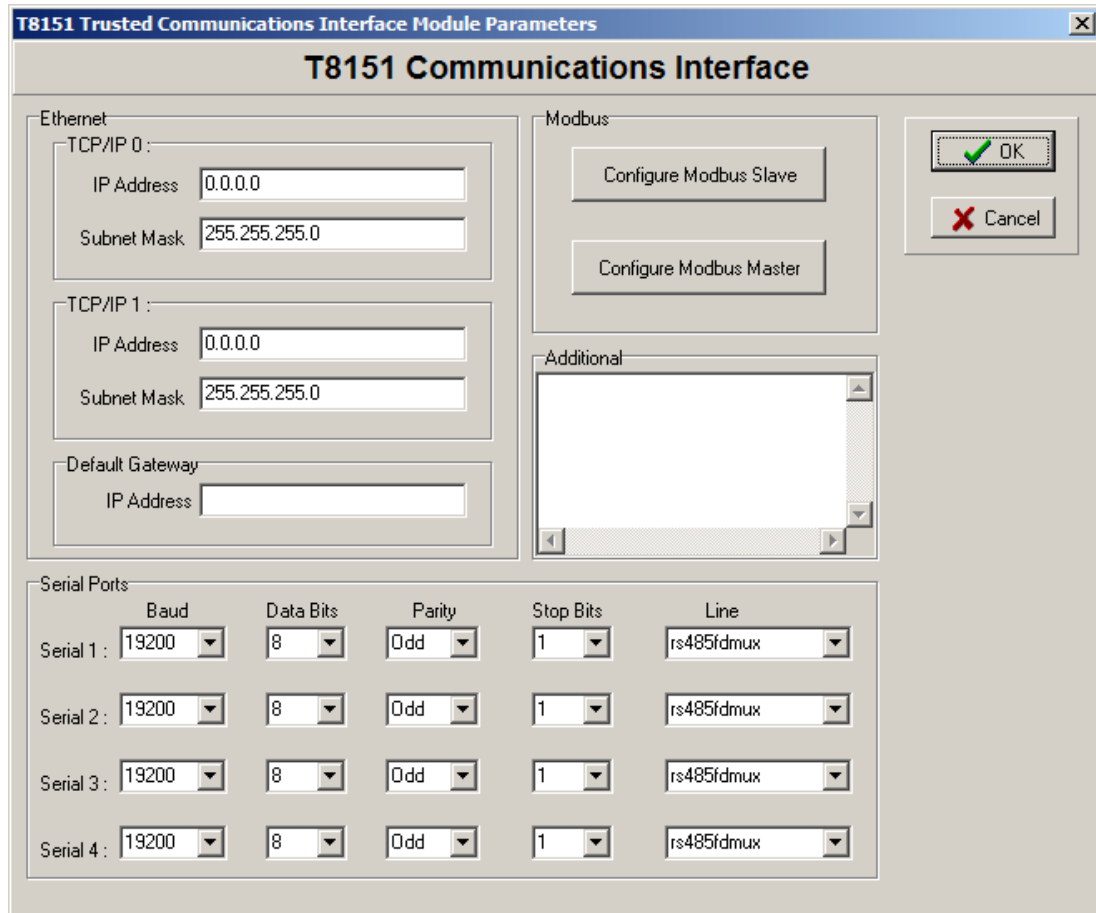


Figure 28 Expander Interface Parameters Window

## 2.7. Trusted TMR Communications Interface Module ( 8151 / 8151B )

The Trusted Communications Interface Modules are all created using the same template. Left-click on the required module slot and select **8151: Communications Interface**. A new Communications module appears in the slot.

Select the module to be configured by clicking the left-hand mouse button on the Communications Module on the opening screen. This will cause the Trusted **Communications Interface Module Parameters** window shown in Figure 29 below to be displayed.



**Figure 29 Communications Interface Parameters Window**

For further information on setting up the port configuration(s) and any associated Modbus addresses, please refer to Product Description PD-8151B, Trusted Communication Interface. This includes extensive details on the Modbus setup and protocol.

The user must select the **OK** button to store the configuration data in the diagram and return the screen display to the **System Configuration Manager** window.

## 2.8. Trusted I/O Modules

Having completed the configuration of the TMR Processor, Expanders and Communication Interface modules the user may assign Trusted I/O modules to the pre-defined slots in the both the Trusted Controller and Expander Chassis.

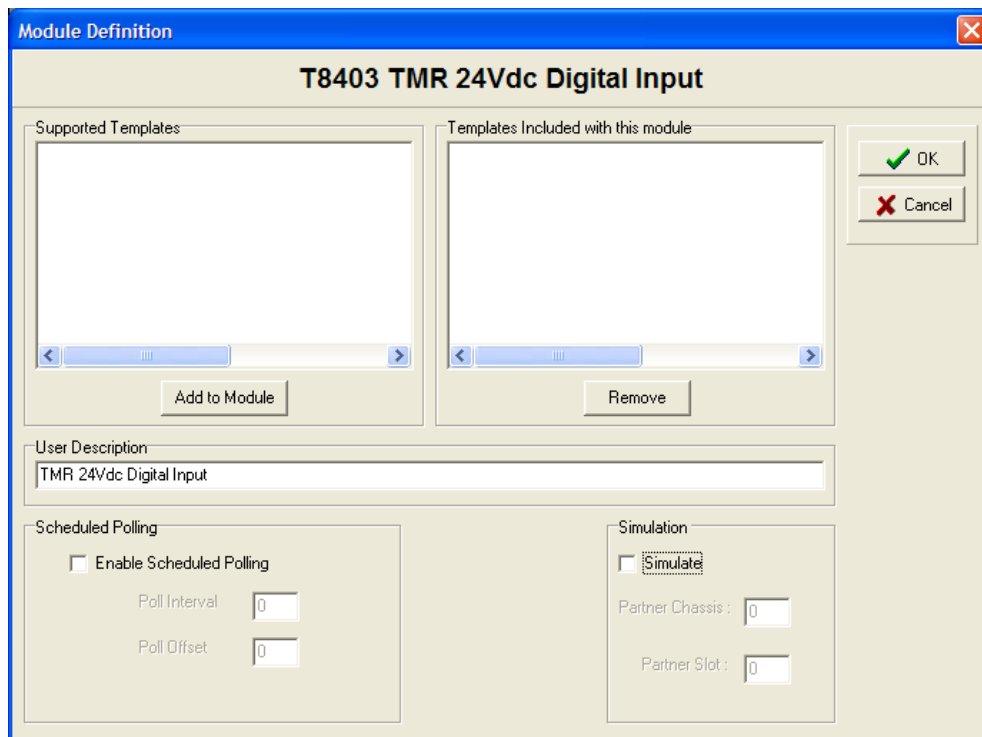
This is achieved by placing the mouse cursor on the slot then clicking the right-hand mouse button. This will cause the **Replace Item In Slot** window shown in Figure 30 below to be displayed.

### Figure 30 'Replace Item in Slot' Window

The **Replace Item In Slot** window shows the Trusted modules which may be assigned to a Controller Chassis.

Modules are assigned to the empty slots by selecting the appropriate module type then selecting the **OK** button. This process must be repeated until all required modules have been assigned to the correct slots.

Having completed the assignment of modules to their correct slots, the user may configure each module as required. This is achieved by pointing the mouse at the appropriate module in the Controller or Expander Chassis and pressing the left-hand button. This will cause the **Module Definition** window shown in Figure 31 below to be displayed, in this example for an 8403 Digital Input Module.



**Figure 31 Module Definition Window**

The **Supported Templates** area of the **Module Definition** windows displays a list of templates contained in the Template Library generated for the Trusted System. These templates have been filtered to only show the categories appropriate for the module. Templates are assigned to the selected module by highlighting the required template in the **Supported Templates** area, then selecting the **Add to Module** button. Templates are deleted by highlighting the template in the **Templates Included with this module** area, then selecting the **Remove** button.

The **Simulation** area allows module simulation to be enabled or disabled (default). **Partner Chassis** and **Partner Slot** become active when the **Simulate** box is enabled allowing the user to identify the position for a unique Companion Slot position for a standby module. **Partner Chassis** and **Partner Slot** should not be specified for a SmartSlot system, because more than one module may use the same smart slot. If the same partner chassis/slot is specified for different modules, the MP will remove the partnering. Therefore, SmartSlot systems should have **Partner Chassis** and **Partner Slot** set to 0.

For Companion Slot modules, enter a module in the primary slot. Tick 'Simulate' and enter the partner (secondary) chassis and slot location. Do NOT enter a module in the secondary slot

Enabling **Simulate** allows the system to start up without the primary module installed. A software simulation of the module is invoked if the module is not present. Simulation provides a set of default values for the field inputs corresponding to the default safe-state (logic '0'). If the module is present the system will use the actual module instead of the simulation model. This feature is useful during system integration as it allows the normal applications to be loaded and tested without a full complement of I/O modules.

The **Partner Chassis** and **Partner Slot** must be specified if the System must be able to start without the primary module fitted. The secondary module position must always be the adjacent slot to the right of that occupied by the primary module when using the Companion Slot configuration. When starting up a system with a secondary module, the simulation model is first opened (inputs signals all report default safe-state) and the system performs an Active/Standby changeover between the simulated model and the actual secondary module.

#### 2.8.1. Scanning Controls: Pre Release 3.5

The **Scanning** area allows the minimum scan rate for the module to be defined. An interval of zero is the default and instructs the system to scan the module as frequently as possible. Where the signals associated with the module may have a slower rate of change, higher priority is required for other signals, or greater time is required to be allocated to the application program execution the scan interval may be set to a slower rate.

Before release 3.5, the only mode selection available for use is time and the scan interval is entered in milliseconds.

The rate at which the input module information is passed to the application program may be defined within the **Check For Change Rates** area. If no update rates are defined, all the input information will be passed to the application program at the rate defined by the scan interval except **Housekeeping** information which is updated once per second. Setting slower rates for individual rack information allows the system performance to be optimised to meet the application specific needs, e.g. voltage information may be used for reporting only and require only relatively slow updates, whereas the Boolean Input status information may be required at the maximum possible rate.

For most applications, it is not necessary to adjust these parameters. If adjustment becomes necessary, the user must enable the function by clicking the mouse in the adjacent box then entering the required data.

**Note:** All input information that is used for safety-critical functions must have an update rate less than one-half the Process Safety Time.

#### 2.8.2. Scanning Controls: From Release 3.5

From release 3.5, the module may be configured to be polled only when required. Check 'Enable Scheduled Polling' to set the module as polled.

'Poll Interval' is how often the module will be scanned, for example 5 means that the module will only be scanned once every 5 application scans.

'Poll Offset' sets which actual application scan will scan the module. This enables module scans to be evenly distributed. For example, if two modules had the Poll interval set to 5, and one have the poll offset set to 3 and the other to 4, the first module would be polled in the third scan and the other would be polled in the fourth scan, so that they do not get polled in the same application scan.

**Note:** All input information that is used for safety-critical functions must have an update rate less than one-half the Process Safety Time.

The **User Description** area may be used to enter any descriptive information about the input module if required. This information is saved in the System.INI file for documentation commentary only and is not used for any operational function of the input module.

Output modules are configured in the same manner as described above.

## 2.9. Unused Slots

Slots which have no modules configured and will never be used as companion slots or hot swap slots may be taken out of the scanning process to shorten the processing times. This action will provide significant gains where there are many empty slots.

To do this, move the cursor over the slot and click the left mouse button.

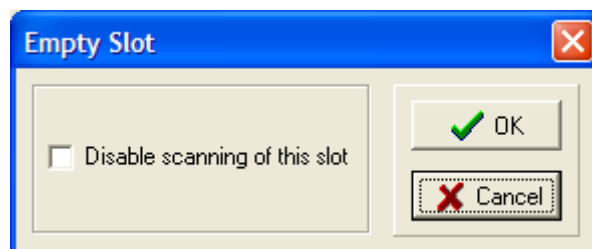


Figure 32 Disable Scanning

To remove the slot from the scanning process, check 'Disable scanning of this slot'. Note that from release 3.5, this action will totally disable all communications to that slot, and this is indicated by a red 'X' over the slot. It is now impossible to hot-swap a module to that slot, or to perform diagnostics on a module moved to that slot. For this reason, do not disable any companion slots or smart slots, and leave one slot active as a diagnostic slot.

## 2.10. Generating the SYSTEM.INI File

At this point, the configuration only exists as a graphical representation in the diagram. When data has been entered/edited into the diagram, the .INI file can be generated from the diagram by selecting the **Generate the INI Buffer** button shown in Figure 33 below.

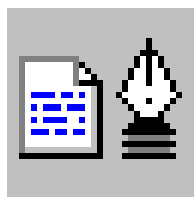
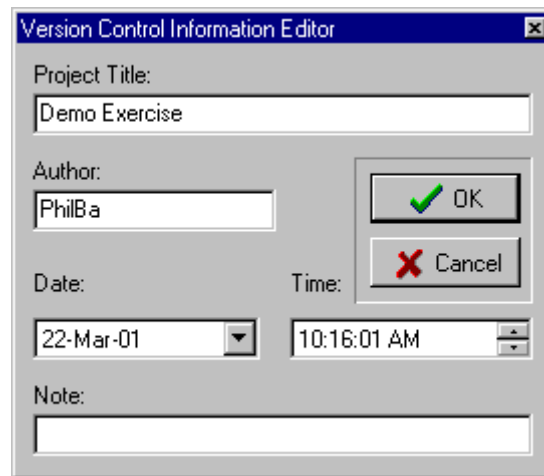


Figure 33 INI Buffer Generation Icon

Selecting the button above initiates the display shown in Figure 34 below.



**Figure 34 Version Control Information Editor Window**

The user must provide a name for the System.INI file in the **Project Title** area of the display above and enter initials/name in the **Author** area. A revision or reference comment can be entered in the **Note:** area. Selecting the **OK** button will then generate the System.INI file from the diagram. The .INI Buffer now matches the diagram.

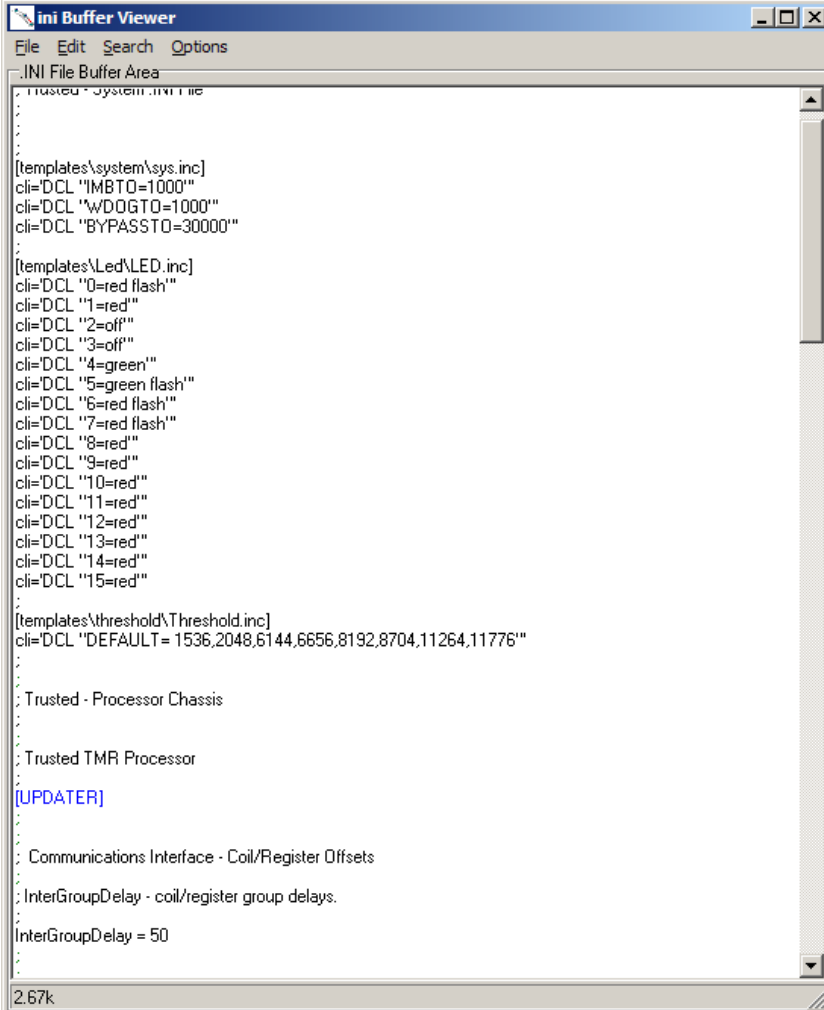
## 2.11. Displaying the SYSTEM.INI File

The current contents of the .INI file in the buffer may be displayed by selecting the **View INI Buffer** button shown Figure 35 below.



**Figure 35 View INI Buffer Icon**

The configured .INI file settings for the system will appear on the screen as shown in Figure 36 below:



```
ini Buffer Viewer
File Edit Search Options
.INI File Buffer Area
Trusted - System.INI file
.
.
[templates\system\sys.inc]
cli=DCL "IMBTO=1000"
cli=DCL "w/DOGT0=1000"
cli=DCL "BYPASST0=30000"
.
[templates\Led\LED.inc]
cli=DCL "0=red flash"
cli=DCL "1=red"
cli=DCL "2=off"
cli=DCL "3=off"
cli=DCL "4=green"
cli=DCL "5=green flash"
cli=DCL "6=red flash"
cli=DCL "7=red flash"
cli=DCL "8=red"
cli=DCL "9=red"
cli=DCL "10=red"
cli=DCL "11=red"
cli=DCL "12=red"
cli=DCL "13=red"
cli=DCL "14=red"
cli=DCL "15=red"
.
[templates\threshold\Threshold.inc]
cli=DCL "DEFAULT= 1536,2048,6144,6656,8192,8704,11264,11776"
.
.
Trusted - Processor Chassis
.
Trusted TMR Processor
[UPDATER]
.
.
Communications Interface - Coil/Register Offsets
.
InterGroupDelay - coil/register group delays.
InterGroupDelay = 50
.
.
2.67k
```

**Figure 36 INI Buffer Viewer**

The above display shows only a part of the System.INI file created in this Product Description. The complete file is shown later.

The **File** menu from the display shown in Figure 36 above provides the user with two options; load an existing .INI file and save the displayed .INI file.

The **Edit** menu enables the user to implement normal Windows 'copy', 'delete', and 'select' functions. This menu also includes an option allowing the user to select specified fonts for the display.

The **Search** menu provides the user with the normal Windows 'find', 'replace', and repeat 'search' functions.

The **Options** menu enables the user to select colour preferences for the display and also to associate all .INI files with the chosen preferences.

## 2.12. Example SYSTEM.INI File

```
;
; Trusted - System .INI File
;
;
;
[templates\system\sys.inc]
cli='DCL "IMBTO=1000"'
cli='DCL "WDOGTO=1000"'
cli='DCL "BYPASSTO=30000"'
;
[templates\Led\LED.inc]
cli='DCL "0=red flash"'
cli='DCL "1=red"'
cli='DCL "2=off"'
cli='DCL "3=off"'
cli='DCL "4=green"'
cli='DCL "5=green flash"'
cli='DCL "6=red flash"'
cli='DCL "7=red flash"'
cli='DCL "8=red"'
cli='DCL "9=red"'
cli='DCL "10=red"'
cli='DCL "11=red"'
cli='DCL "12=red"'
cli='DCL "13=red"'
cli='DCL "14=red"'
cli='DCL "15=red"'
;
[templates\threshold\Threshold.inc]
cli='DCL "DEFAULT= 1536,2048,6144,6656,8192,8704,11264,11776"'
;
;
; Trusted - Processor Chassis
;
;
; Trusted TMR Processor
;
[UPDATER]
;
;
; Communications Interface - Coil/Register Offsets
;
; InterGroupDelay - coil/register group delays.
```

```
;
InterGroupDelay = 50
;
;
; Block Write Protection Of Modbus Variables.
;
AutoProtect = 0
;
[ISAGRAF_CONFIG]
;
ISA_SLEEP_PERIOD = 32
MAX_SCAN_TIME = 1000
;
[IMB]
SC=2,2,NIO
SC=2,1,NIO
SC=2,13,XPM
SC=2,14,XPM
CC=2,2,1
SC=1,8,PIM
SC=1,7,PIM
SC=1,2,XIM
SC=1,1,XIM
SC=1,15,MP
;
;
; ICS 2000 Configuration
;
[I2K_SYSTEM]
a=a
b=a
c=a
;
;
[COMMS]
serial1=19200,8,Odd,1,rs232
serial2=19200,8,Odd,1,rs485fdmux
serial3=19200,8,Odd,1,rs485fdmux
;
[MODBUS]
;
[IRIG]
Enabled=0
Mode=1
```

```
LEDMonitoring=0
;
[CHASSIS]
max_racks =2
;
;
[CHASSIS_01_SLOT_01]
module_type = 8311
;
[CHASSIS_01_SLOT_02]
module_type = 8311
;
[CHASSIS_01_SLOT_07]
module_type=8151
serial1=19200,8,Odd,1,rs485fdmux
serial2=19200,8,Odd,1,rs485fdmux
serial3=19200,8,Odd,1,rs485fdmux
serial4=19200,8,Odd,1,rs485fdmux
tcp_ip0=10.10.10.2,255.255.255.0,11.10.9.8
tcp_ip1=9.9.9.2,255.255.255.0,11.10.9.8
;
; Modbus slave settings
;
mb_port_timeout=300
;
[CHASSIS_01_SLOT_08]
module_type=8151
serial1=19200,8,Odd,1,rs485fdmux
serial2=19200,8,Odd,1,rs485fdmux
serial3=19200,8,Odd,1,rs485fdmux
serial4=19200,8,Odd,1,rs485fdmux
tcp_ip0=10.10.10.1,255.255.255.0,11.10.9.8
tcp_ip1=9.9.9.1,255.255.255.0,11.10.9.8
;
; Modbus slave settings
;
mb_port_timeout=300
;
; Trusted - Expander Chassis
;
; Chassis Number 2
;
;
[CHASSIS_02_SLOT_01]
module_type = 8403
```

```
description=TMR 24Vdc Digital Input
simulate= 0
scan_interval = 0
;
cli='DCB'
cli='DCL "[SYSTEM]'"
${templates\system\sys.inc}
cli='DCL "[LED]'"
${templates\Led\LED.inc}
cli='DCL "[THRESHOLD]'"
${templates\threshold\Threshold.inc}
cli='DCE'
;
[CHASSIS_02_SLOT_02]
module_type = 8451
description=TMR 24Vdc Digital Output
simulate= 0
scan_interval = 0
;
cli='DCB'
cli='DCL "[SYSTEM]'"
${templates\system\sys.inc}
cli='DCL "[LED]'"
${templates\Led\LED.inc}
cli='DCE'
;
[VCI]
;
;VERSION CONTROL INFORMATION
;The <CRC> variable is calculated on everything
;excluding the CRC line
;
Author = NickO
ProjectTitle = Demonstration File
Date = 18/06/2004
Time = 12:32:01
Note =
```

## 2.13. Generating the SYSTEM.INI File Diagram

An additional feature of the System Configuration Manager enables the user to generate a diagram of the Trusted system from the System.INI file. This is achieved by selecting the **Regenerate Graphic from INI Buffer** button on the tool bar of the opening screen, shown in Figure 37 below.

Figure 37 Regenerate Graphic from INI Buffer Icon

Selecting this button updates the display of the System diagram as shown in Figure 38 below.

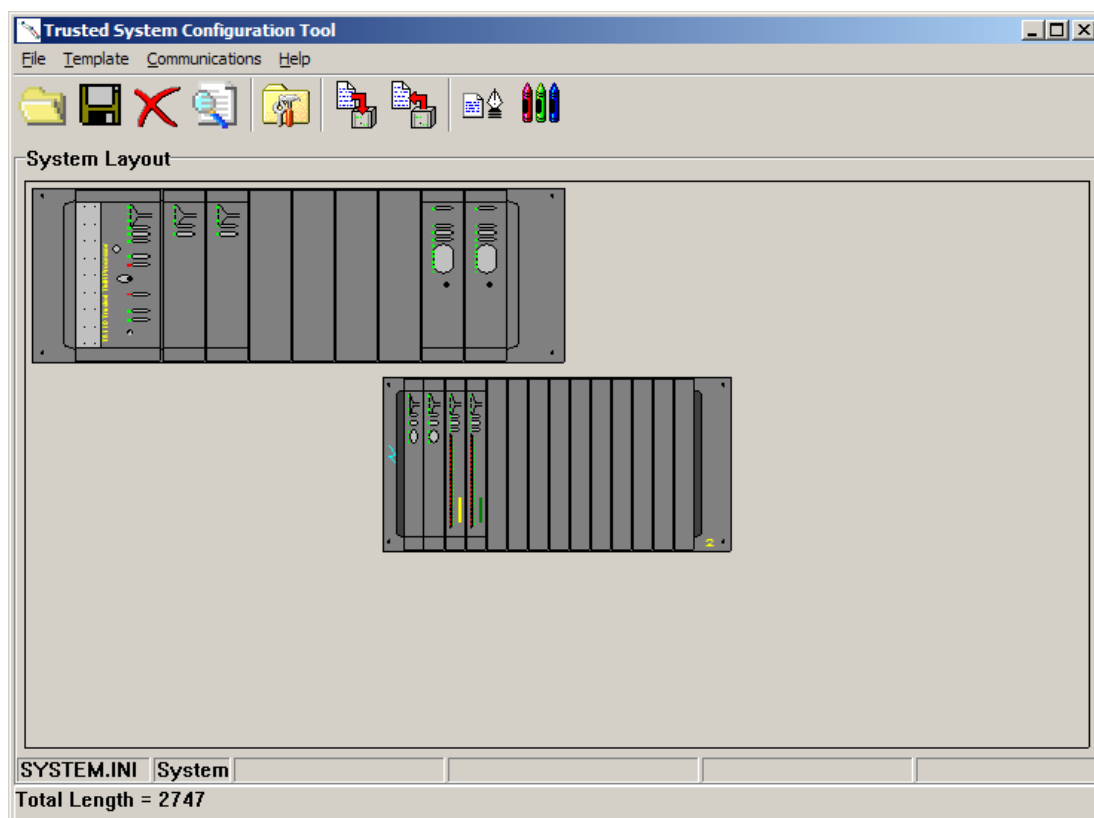


Figure 38 Updated Diagram

## 2.14. Downloading to the TMR Processor

The System.INI file may be downloaded to the Trusted TMR Processor by selecting the **Download .INI Buffer to MP** button on the tool bar of the **System Configuration Manager** screen, shown in Figure 39 below.

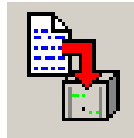


Figure 39 Download INI Buffer to MP Icon

Selection of the **Download .INI Buffer to MP** button starts the process of downloading the System.INI file to the Processor, given that the communications is set up and the Trusted System TMR Processor keyswitch is set in the Maintain position. Whilst this taking place, the **Communications in progress** window shown below is displayed.

Figure 40 Communications in Progress

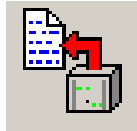
The System.INI file is transmitted in packets of data. The **TX** LED in the **Communications in progress** window flashes as each packet of data is being downloaded. The **RX** flashes to indicate that a packet has been successfully downloaded.

The **Cancel** button changes to **OK** when all data has been successfully downloaded. When this has occurred, the user must select **OK** to return to the **System Configuration Manager** screen.

Note: Once downloading of the edited/new SYSTEM.INI file has been completed, the user can either reboot the Trusted TMR Processor to implement the file, or stop then start the application.

## 2.15. Uploading to the EWS

The System.INI file may be uploaded from the Trusted TMR Processor by selecting the **Upload the .INI Buffer From MP** button on the tool bar of the **System Configuration Manager** screen, shown in Figure 41 below.



**Figure 41 Upload INI Buffer from MP Icon**

Selection of the **Upload the .INI Buffer From MP** button starts the process of uploading the System.INI file to the EWS. Whilst this is taking place, the **Communications in progress** window shown below is displayed as shown in Figure 42 below.

**Figure 42 Communication in Progress Window**

The System.INI file is transmitted in packets of data. The **TX LED** in the **Communications in progress** window flashes as each packet of data is being uploaded. The **RX** flashes to indicate that a packet has been successfully uploaded.

The **Cancel** button changes to **OK** when all data has been successfully uploaded. When this has occurred, the user must select **OK** to return to **System Configuration Manager** screen.

## 2.16. Archiving

There is a facility on the System Configuration Tool to archive files. From release 3.5, the templates are extracted from the System.INI file itself, and so there is no need to archive templates. However, if the configuration is to be transferred to a toolset before release 3.5, then a template archive is necessary.

Within the Template Editor (**Manage Templates** area) select **Archive Tool**.

### 2.16.1. Save Configuration

This option will allow the user to save the manage template and .INI file. This option is not present from release 3.5.

On selecting **Save Configuration** the user is prompted to select the INI file to be added to the Archive. This can be selected from the directory structure appearing on the left hand part of the window. Once selected the user will again be prompted to select an Archive file to create. A name should be written into the **File name:** box. Selecting **ok** will save the archive. This generates a .ZIP archive of the .INI file and templates.

### 2.16.2. Delete Current Configuration

This option will delete all existing templates. This option is not present from release 3.5.

### 2.16.3. Extract Configuration

This option will allow the user to extract an archive template directory and possible included .INI file. This option is not present from release 3.5.

On selecting **Extract Configuration** the user will be prompted that any existing templates will be deleted, **Ok** will invoke this instruction. The archive files are saved into the working folder, and the templates appear in the Template Editor window. The extracted .INI file should be available in the working folder.

### 2.16.4. Archive Templates

A selection of templates can be saved to a .ZIP archive. Select the templates to archive and specify a filename for the archive.

### 2.16.5. Restore Templates

Templates in an archive can be restored using this command. Select the archive to open. All the templates in the archive will be added to the Template Editor window.

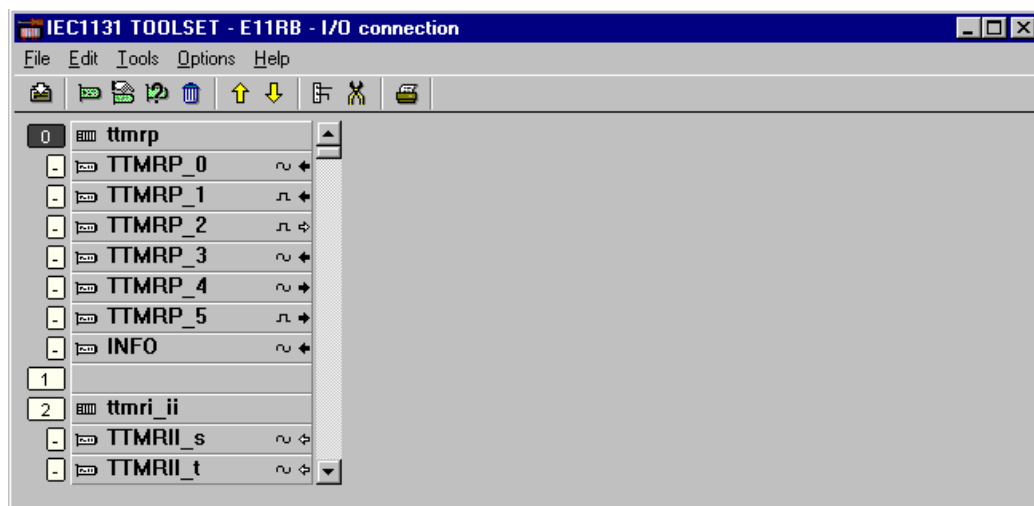
## 3. The I/O Connection Editor

### 3.1. Using the I/O Connection Editor



The aim of the I/O connection operation is to establish a logical link between the I/O variables of the application and the physical channels of the boards existing on the target machine. To make this link the user has to identify and setup all the boards of the target machine, and place I/O variables on corresponding I/O channels.

The I/O connection window is shown in Figure 43 below: -










**Figure 43 I/O Connection Window**




In the window above, the column on the far left shows the connection table (**rack**) of the target machine, with **board slots**. A slot may be free, or used by one I/O board or complex equipment. Each slot is identified by an **order number**. The column on the right of the rack shows the boards allocated to the rack. The area on the right of these boards (blank in the example above) shows the board's parameters and the variables connected on the selected board. A board may have up to **128** I/O channels.

## 3.2. Icons




The icons displayed on each board slot indicate the type and attributes of variables that may be connected to the board channels. The **IEC 61131 TOOLSET** system does not allow the connection of variables of different types on the same board. Icons for board types are as shown below.

-  .....Boolean type
-  .....integer/real type (both types of variables may be connected)
-  .....message type
-  .....inputs - no channel connected
-  .....outputs - no channel connected
-  .....inputs - at least one channel connected
-  .....outputs - at least one channel connected



The icons shown below are used to show the type of I/O device installed on a slot.

-  .....complex I/O equipment
-  .....real I/O board
-  .....virtual I/O board

The icons shown below are used to draw a parameter or a channel.

-  .....board parameter
-  .....free channel
-  .....connected channel

Use the buttons shown below in the toolbar, or **Edit / Move board up/down** menu commands, to move the selected I/O board one line up or down in the main list.

  .....Moving boards in list

The **Edit/Insert slot** command inserts an empty slot at the current position.

### 3.3. Defining I/O Boards

The **Edit** menu contains the commands to define the selected board (setup its parameters) and to connect I/O variables to its channels.

#### 3.3.1. Selecting I/O board type

Before connecting I/O variables to a board, the board identification must be entered. A library of pre-defined boards is available on the **IEC 61131 TOOLSET** workbench. This library may have been compiled by one or more I/O device suppliers. The **Edit/Set Board/Equipment** command is used to setup board identification. This command can be used to select either a single board or complex I/O equipment, from the **IEC 61131 TOOLSET** library. It is also possible to double click on a slot to set the corresponding board or equipment.

All the channels of a single board have the same type (Boolean, integer/real or message) and direction (input or output). Real and integer variables are not distinguished during I/O connection. A complex I/O equipment represents an I/O device with channels of different types or directions. A complex I/O equipment is represented as a list of single I/O boards. It uses only one slot in the rack list.

#### 3.3.2. Technical notes


The **Edit/Technical note** command displays the on-line user's guide of the selected board or complex equipment. The technical note and associated product description contain all the information required to enable the user to configure the board correctly.

#### 3.3.3. Removing a board

The **Edit/Clear slot** command is used to remove the currently selected board or I/O equipment. If variables are already connected to the corresponding channels, they are automatically disconnected when clearing the slot.

#### 3.3.4. Real boards and virtual boards

The **Edit/Real/virtual board** command sets the validity of the selected board or complex I/O equipment. The following icons are displayed in the rack list to show the validity of a board:

.....real I/O board

.....virtual I/O board

In **Real Mode**, I/O variables are directly linked to the corresponding I/O devices. Input or output operations in the application program tie directly to corresponding input or output conditions of the actual field I/O devices. In **Virtual Mode**, I/O variables are processed exactly as internal variables. They can be read or updated by the debugger, so that the user can simulate the I/O processing, but no real world connection is made.

### 3.4. Setting Board Parameters

To set the value of a board parameter, double click on its name in the list on the right. It is also possible to select (highlight) it and choose the **Set channel/parameter** command of the **Edit** menu. Parameters are shown at the beginning of the list. The following icon is used to represent them in the list:

.....board parameter

The meaning and input format of the parameter are designed by the supplier of the corresponding I/O board or equipment. Use the **Tools/Technical note** command or refer to the hardware manual for more information regarding board parameters.

### 3.5. Connecting I/O Channels

When a board is defined, the user has to connect I/O variables to its channels. The following icons are used to represent channels in the list:



.....free channel

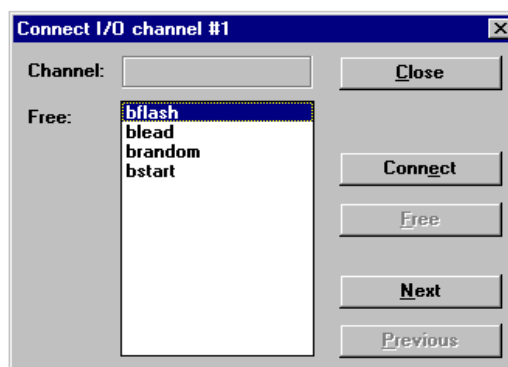


.....connected channel

#### 3.5.1. Connecting variables

To set the connection of a channel, double click on its location in the list on the right. It is also possible to select (highlight) it and run the **Edit/Set channel/parameter** command.

The dialogue box shown in Figure 44 below appears.



**Figure 44 Connect I/O Channel Dialogue Box**

The list contains all the variables which match the selected board type and direction. Only variables which are not yet connected are listed here. The **Connect** button connects the variable selected in the list to the selected channel. The **Free** button removes (disconnects) the variable from the selected channel. **Next** and **Previous** buttons are used to select another channel of the board. The location of the selected channel is always displayed in the title of the dialogue box.

### 3.5.2. Removing connected variables

The **Tools/Free board channels** command disconnects all the I/O variables already connected on the selected board.

### 3.5.3. Comments

The comment text of a declared I/O variable is displayed with its name in the board list. As **IEC 61131 TOOLSET** also enables the use of directly represented variables (% notation), comments may also be attached to free channels. Use the **Tools/Set channel comment** command to enter a comment for the free channel currently selected in the board list. This command cannot be used to change the comment of an I/O variable declared in the project dictionary. However, these comments appear in the program editors in the **Select Variable** dialogue, against the appropriate directly represented variable.

## 3.6. Directly Represented Variables

Free channels are the ones which are not linked to a declared I/O variable. **IEC 61131 TOOLSET** enables the use of **directly represented variables** in the source of the programs to represent a free channel. The identifier of a directly represented variable always begins with % character.

Within a toolset program, the directly represented variables are listed on opening the variable dialogue to type in a variable name, as shown in Figure 45 below.

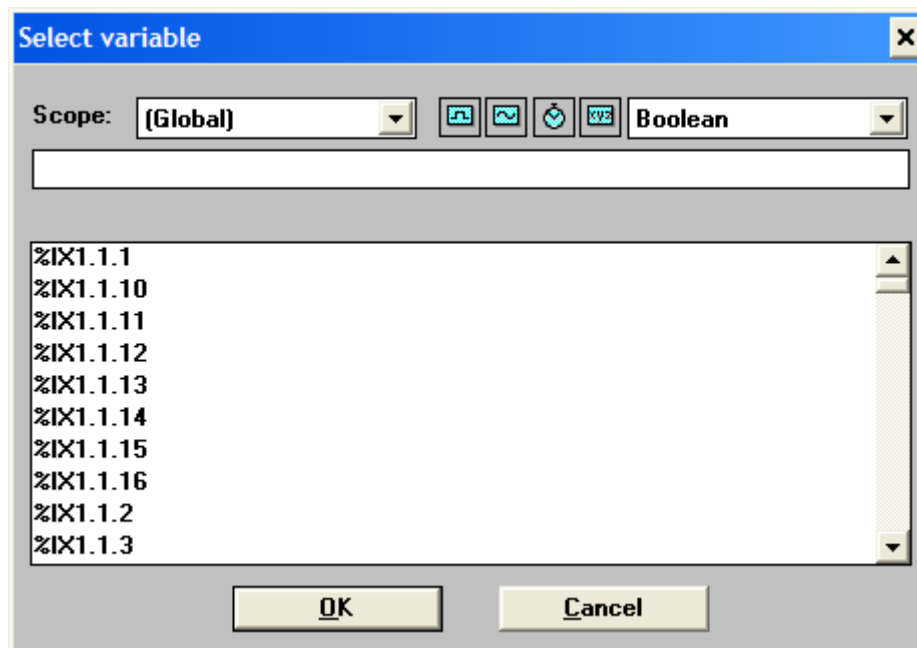


Figure 45 Select Variable Dialogue

The naming conventions of a directly represented variable for a channel of a single board are listed below. *s* is the slot number of the board. *c* is the number of the channel.

- `%IXs.c`.....free channel of a Boolean input board
- `%IDs.c`.....free channel of an integer input board
- `%ISS.c`.....free channel of a message input board
- `%QXs.c`.....free channel of a Boolean output board
- `%QDs.c`.....free channel of an integer output board
- `%QSs.c`.....free channel of a message output board

## 4. Trusted Board Definitions

Details of complex I/O equipment definitions are given in the appropriate I/O module or interface/bridge module PD. These define the data exchanged between the system and application in relation to a particular module. Each module in the system needs an entry in the I/O Connection Editor, as described in section 3 above. The definitions given in this section are for single board definitions, used for non-module related tasks.

### 4.1. Process Historian

Description: Process Historian with selectable Delta capture

OEM PARAMETERS:

OEM parameter	Valid numbers	Description
DELTA	1 – any	The change in the variable that is needed from the last scan to log the variable to the process historian collector.
BANK	1 – 32	There are 32 banks. These all contain a maximum of 100 analogue variables

Table 5 OEM Parameters

PARAMETERS

BOARD_ID	7000
DELTA	100
BANK	1

### 4.2. Sequence of Events

Description: Sequence of Events (SOE). Any variable connected to a channel of this board will be stored in the SOE buffer on the Trusted Communications Interface. The board has the facility for up to 100 variables.

OEM PARAMETERS:

None

## 5. Trusted workstation additions

Two Trusted Workstation Additions, listed and briefly described below, are supplied as part of the T8082 software package. The files are identified by their file names.

vxa.dll	Enables points within the application program to be nominated for SOE reporting. Defines additional parameters for Modbus interface
isdk_com.dll	Provides the ICS communications protocol to the Trusted TMR Communications Interface Module.

## 6. ISaGRAF Workstation Enhancements

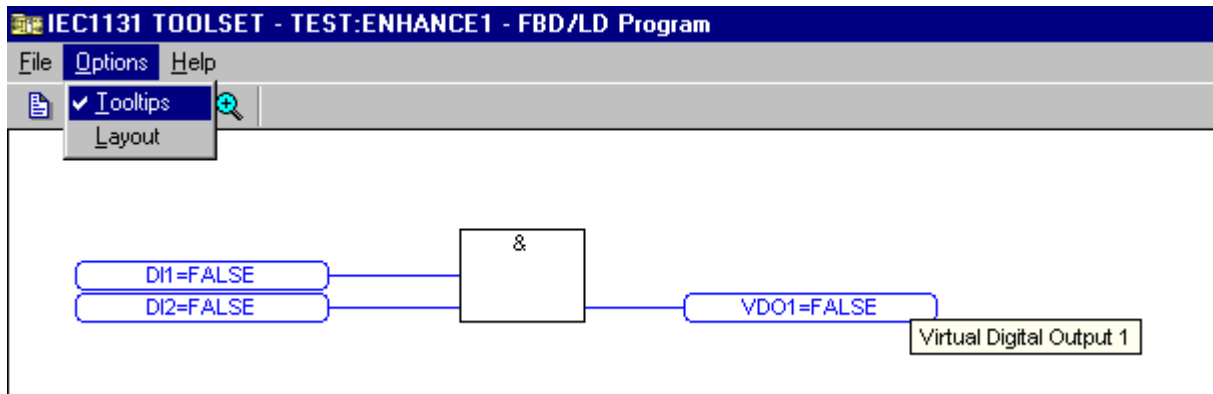


Figure 46 Enabling Tooltips

### 6.1. Tooltips Enabled

Tooltips can be selected when the workstation is monitoring the program on-line via the drop down menu **Options Tooltips** as shown in Figure 46 above. Enabling this feature will allow the dictionary comment field to be displayed by floating the mouse pointer over the tag name.

### 6.2. Comment Variable Attachment

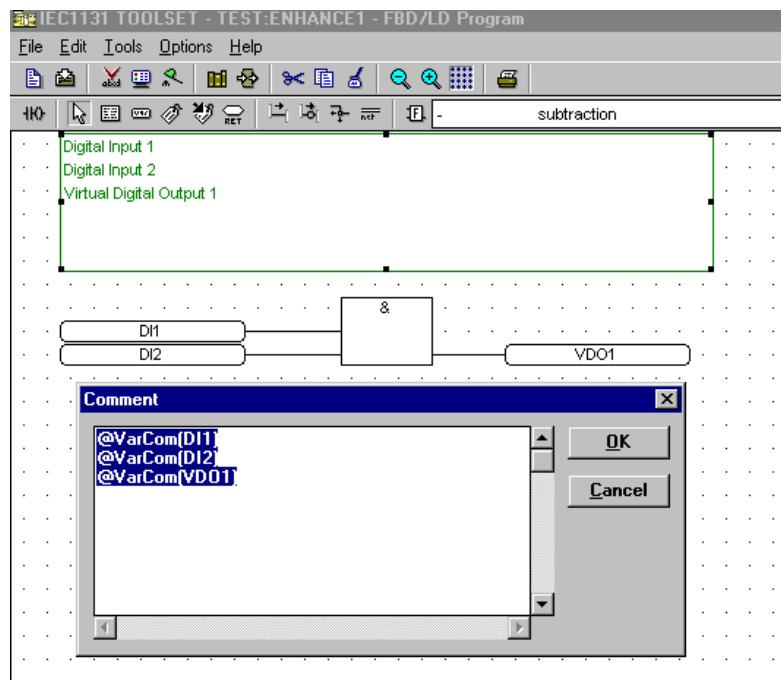


Figure 47 Referencing variable comments

Dictionary tag comments can be assigned by inserting a comment box into the program and typing **@VarCom(dictionary tag)** . This will lift the comment field from the dictionary and display it in the comment box as illustrated in Figure 47 above.

### 6.3. Power Flow Debugging



Figure 48 Power Flow Debugging Off

Power Flow Debugging is an option that makes complex logic resolution more efficient and easier to read. Interconnections between gates are animated blue or red according to their state without having to connect them to internal tags Figure 48 above shows Power Flow Debugging turned off. Note that the connection between the & (AND) and =1 (exclusive OR) gates is black.

To enable this function from the program editing window, select **Option** then **Enable power flow debugging** as shown in Figure 49 below.

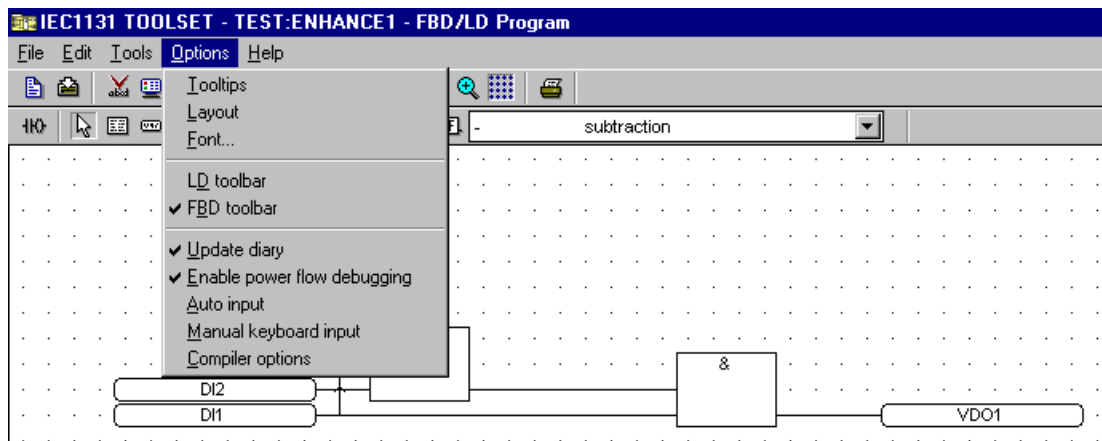


Figure 49 Enable Power Flow Debugging

When the application is compiled and downloaded, the Power Flow Debugging facility will be enabled. Compare Figure 48 above with Figure 50 below, where the connection between the gates now reflects the logic state.

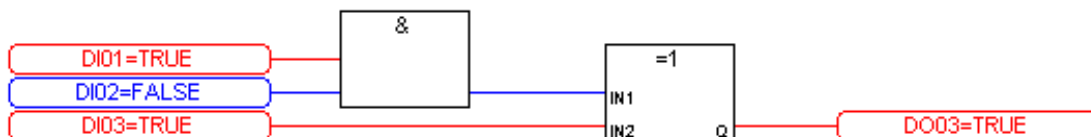
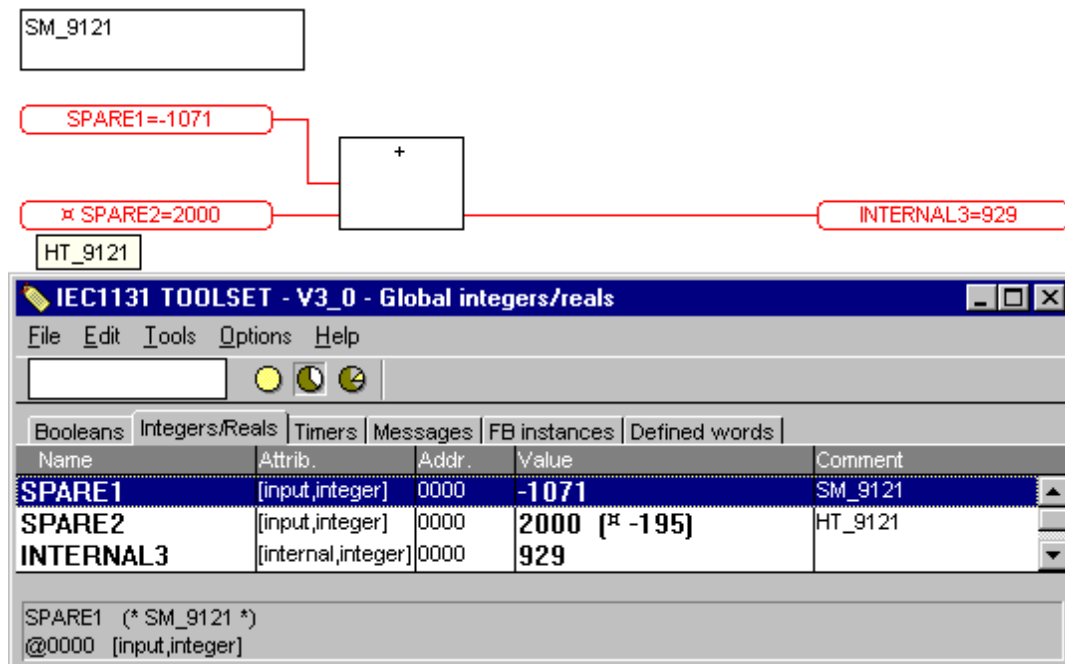


Figure 50 Power Flow Debugging On

## 6.4. I/O Forcing

As part of the operation and maintenance of a safety system, there are situations that arise requiring the forcing (locking) of data points. It is always an advantage to be aware of the fact that data points on the system have been forced. The toolset now provides extensive monitoring facilities of forced data, as shown in Figure 51 below.



**Figure 51 Forced Data Point**

When a data point is forced, a symbol is displayed in the logic and dictionary to reflect that the point has been forced. In the dictionary, both the forced value and the true field value are displayed as illustrated in Figure 51 above. The actual field value on an I/O point is contained within brackets. In the example above, the field value is -195 and the forced value is 2000.

The ISaGRAF Users' Guide section A.26.2 describes the way forcing operates. Forcing removes the connection between the application variable and the physical module. Forcing is not allowed on internal variables.

For an input, the variable is disconnected from the module. The forced value is applied to the variable, so that the application sees the forced value. The real input is still visible in the dictionary, shown in brackets above.

For an output, the variable is also disconnected from the module. In this case, the application is unable to change the real output. However, forced values are applied to both the application variable and to the real output. Normally the application will immediately overwrite this value, but if the output is read before it is written, the forced value will be read until the point in the code where the output is written. Therefore it is not recommended to read from an output variable before it is written, e.g. in creating a latching ladder rung.

SOE is disabled for forced I/O points.

## 6.5. Integer Interfacing Enhancements

There are four options for the **Extended** Attributes of analogue data. These control the manner in which data is controlled and interpreted over Modbus communications. The options dialogue for Extended Attributes is shown in Figure 52 below.

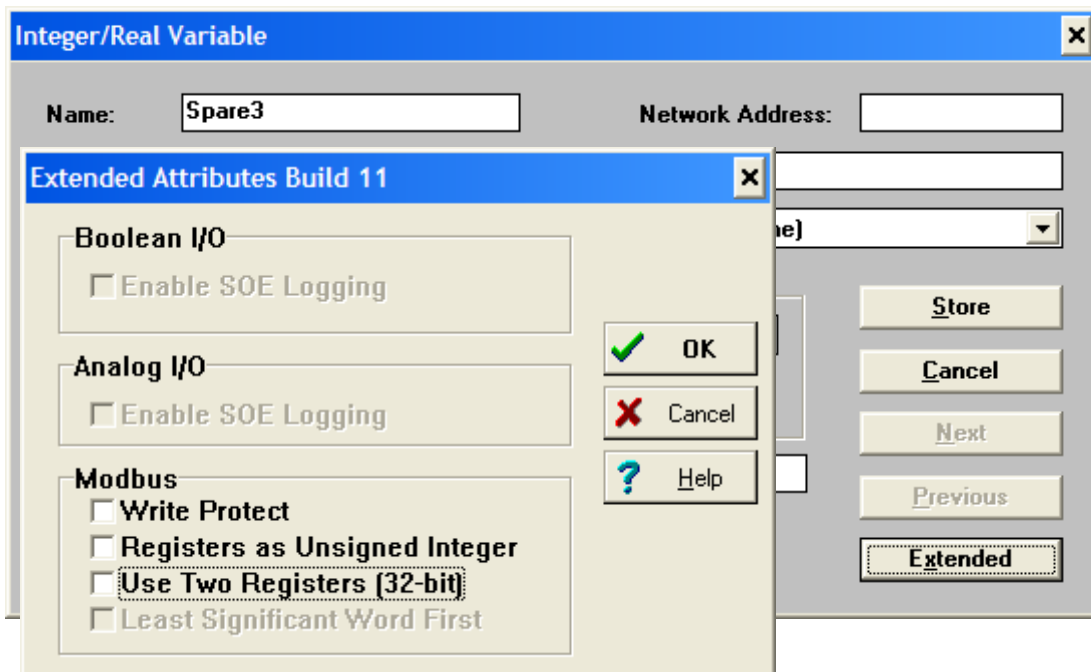


Figure 52 Extended Attributes

### 6.5.1. Write Protect

Checking this option inhibits external communications from overwriting a data point. This applies to both Integer/Real and Boolean variables.

### 6.5.2. Registers as Unsigned Integer

Integer data is normally treated as 32-bit signed within the application (for signed data, the most significant bit is used to distinguish positive and negative numbers). This option reads and writes Modbus data as unsigned data.


### 6.5.3. Use Two Registers (32-bit)


By default, only the lower 16 bits of an analogue data point are transferred to and from the Modbus communications. This option uses two contiguous Modbus registers to transfer all 32 bits.

### 6.5.4. Least Significant Word First

If 'Use Two Registers' is selected, by default the most significant 16 bit word is addressed first on Modbus communications. This option reverses the words on the Modbus addresses.

## 6.6. Printing

The Program editor can show page breaks on the diagram. To display page breaks, open the Document Generator window by selecting menu option **File** then **Print**. Close the Document Generator window without printing. Show the grid on the diagram by selecting icon  or by selecting menu option **Options – Layout – Display Grid**. In addition to the grid, the page breaks are now visible as shown in Figure 53 below. This allows the logic to be arranged to avoid crossing a page break.

Note that page breaks may be inserted into the diagram manually. These are called **Network Breaks** and may be inserted using the icon .

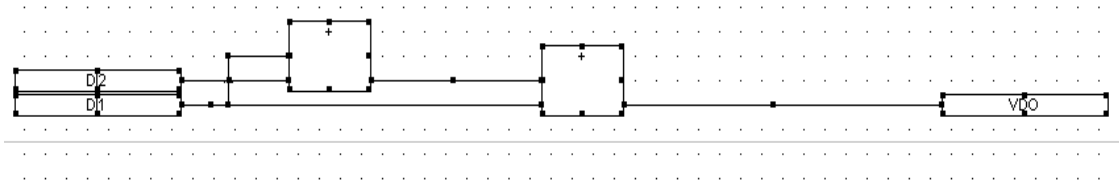


Figure 53 Page Break Boundary

## 6.7. Process Execution Timing Data

Detailed information on the timed operations carried out on each scan are available via the SYSTEM function block. This function block is available in the FBD program menu and has an application note available from the 'Note' button on the SYSTEM function block 'Info' window. This application note gives detailed information on configuration in FBD and structured text programs.

The function block allows access to a variety of information by passing a category keyword to the Mode input and an argument number to the Arg input. The category keyword SYS\_PE\_TIMES (value 128) is an addition which provides the following data:

Argument	Arg input value	
PE_INIT	0	Returns 0 on the first scan of the application (i.e. each time the application is started), or 1 otherwise. When it returns 0, none of the elapsed times are valid.
PE_ALL	1	Returns the elapsed time from the start of one scan to the start of the next scan.
PE_CYC	2	Returns the elapsed time from the last start of the APPLICATION processing to this current start of the APPLICATION processing.
PE_IO	3	Returns the elapsed time from the start to the IO scan to the end of the IO scan. Includes all IMB transactions such as NIO output processing, NIO input processing, TIO processing, REGENT processing, COMMUNICATION module processing, STANDBY MP processing, etc.
PE_INP	4	returns the elapsed time from the start of INPUT processing to the end of INPUT processing. Includes updating OEM boards, lock counts, i2k, nvm updates, and conversions.
PE_APP	5	returns the elapsed time from the start of the APPLICATION processing to the end of the APPLICATION processing.
PE_COM	6	returns the elapsed time from the start of the COMMS processing to the end of the COMMS processing. Includes all OSE signals directed to the ISaGRAF kernel such as .INI changes, ACT/STDBY signals, communication requests from the ISaGRAF comms task, etc.
PE_OUT	7	returns the elapsed time from the start of the OUTPUT processing to the end of the OUTPUT processing. Includes updates from OEM boards (i.e. generating the FOB), i2k, nvm, SOE, PH

Notes:

All calls to SYS\_PE\_TIMES other than PE\_INIT return elapsed time in milliseconds.

All calls to SYS\_PE\_TIMES other than PE\_INIT are invalid on the initial scan.

All time data calls to SYS\_PE\_TIMES return the elapsed times from the PREVIOUS scan, except PE\_CYC which is valid for the current scan (except the initial scan).

Figure 54 below shows a schematic of the timing data described above.

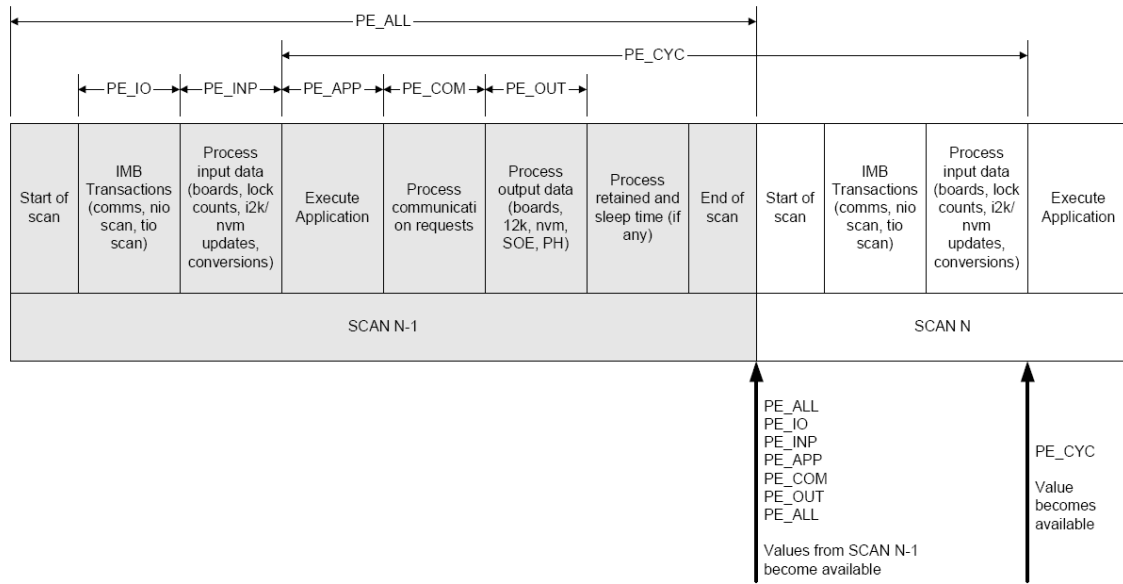


Figure 54 SYS\_PE\_TIMES time data ranges

## 7. Intelligent Online Updates

An on-line update involves changing an application on the Toolset which is currently running on a system, loading those changes into the system, then having the system switch to the changed application without interruption to the process that the application is controlling.

The Trusted system supports two types of on-line updates: 'Normal' and 'Intelligent'. Normal updates are available in all released versions of the Trusted system. Intelligent Updates are only supported in release 3.4 and above. Both on-line update features enable the user to modify the application while the process is running. While both types of on-line updates perform essentially the same function, Intelligent Updates allow the application to be modified in a number of ways that Normal Updates would not allow.

'Normal' updates allow changes to an application as long as the data memory space doesn't change between the applications (the 'CRC' for the application must not change). This means that there can be no added or deleted variables, or added or deleted function blocks called within programs, among other changes which affect the data space (see the Toolset User Guide section A.15.6). This means that spare variables, function blocks, programs etc. must be declared in advance, and 'MNTV' settings in the Toolset ISA.INI file must be set to specify the number of spare variables of each type (details are given in the Toolset User Guide as above).

'Intelligent' updates allow the data memory space to change, allowing more changes to be made. The toolset has an extension called the Intelligent Online Update Manager. This compares the running and new applications and attempts to match up data points and values between the two compilations. This allows updates in more cases but cannot deal with all possible changes.

Intelligent updates are not enabled by default. It is important to enable them before commissioning a system. The following section explains how to enable and use them.

### 7.1. Enabling Intelligent Online Updates

The Intelligent Update Manager is accessed from the Programs window. Select Tools – Isa.mnu – Online Update Manager as shown in Figure 55 below.

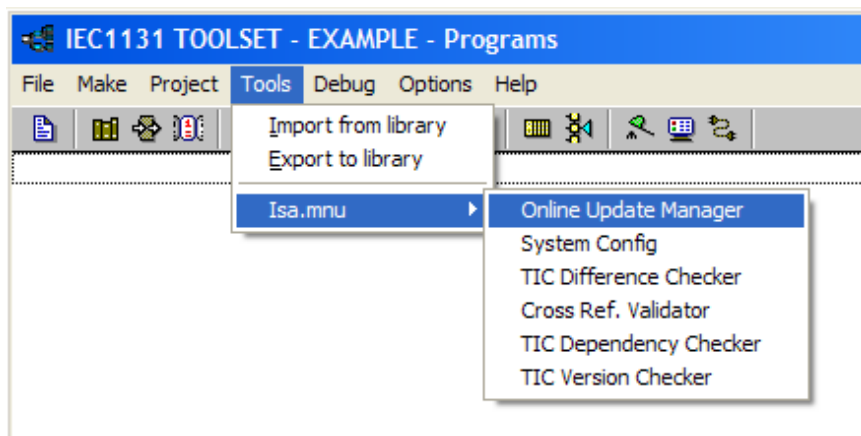
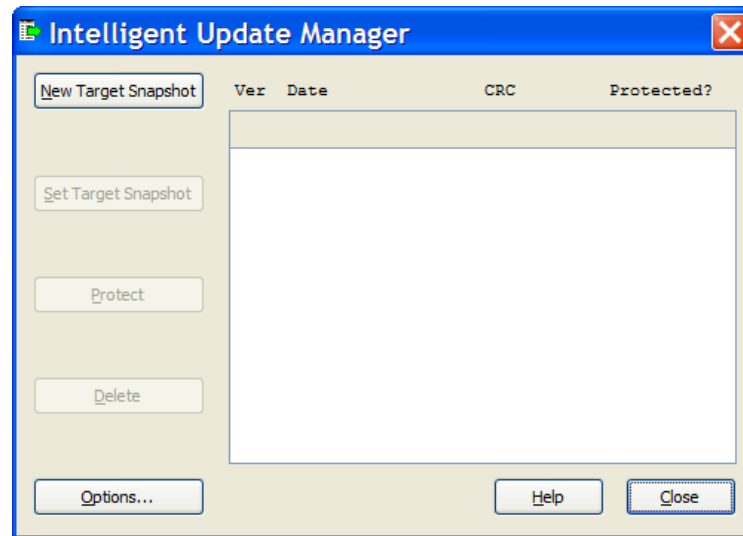


Figure 55 Starting the Intelligent Update Manager

The Intelligent Update Manager window is shown in Figure 56 below. The controls are explained in application note AN-T80009.



**Figure 56 Intelligent Update Manager**

Intelligent Updates are disabled for a new project. To enable Intelligent Updates for a project, click on the 'Options...' button to open the dialogue in Figure 57 below, then select 'Enable Intelligent On-line Updates'. It is recommended to select all five information options; these provide important information at compilation and download which should be read with care.

'Allow function block matching' adds the ability to confirm the matching of function blocks between the existing and new compilations. Its sub option enables the recording of previous matching decisions. These options are available from release 3.5.1 and it is recommended to select these two options.

The options are explained in Application Note AN-T80009.

Click 'OK'.

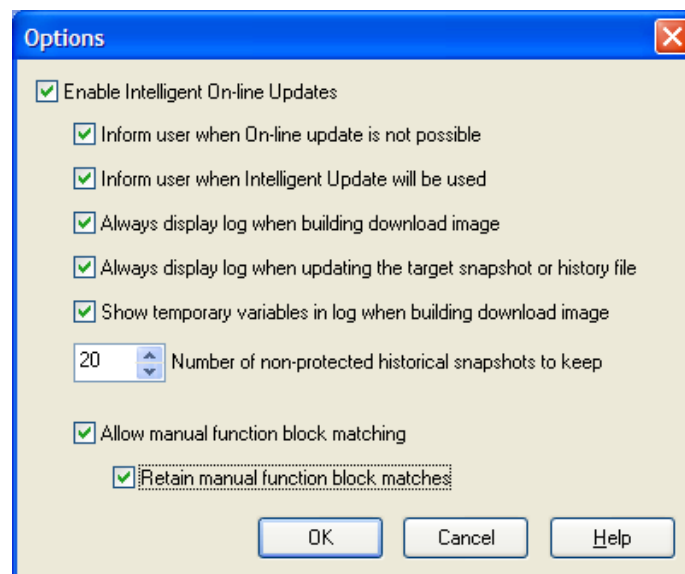
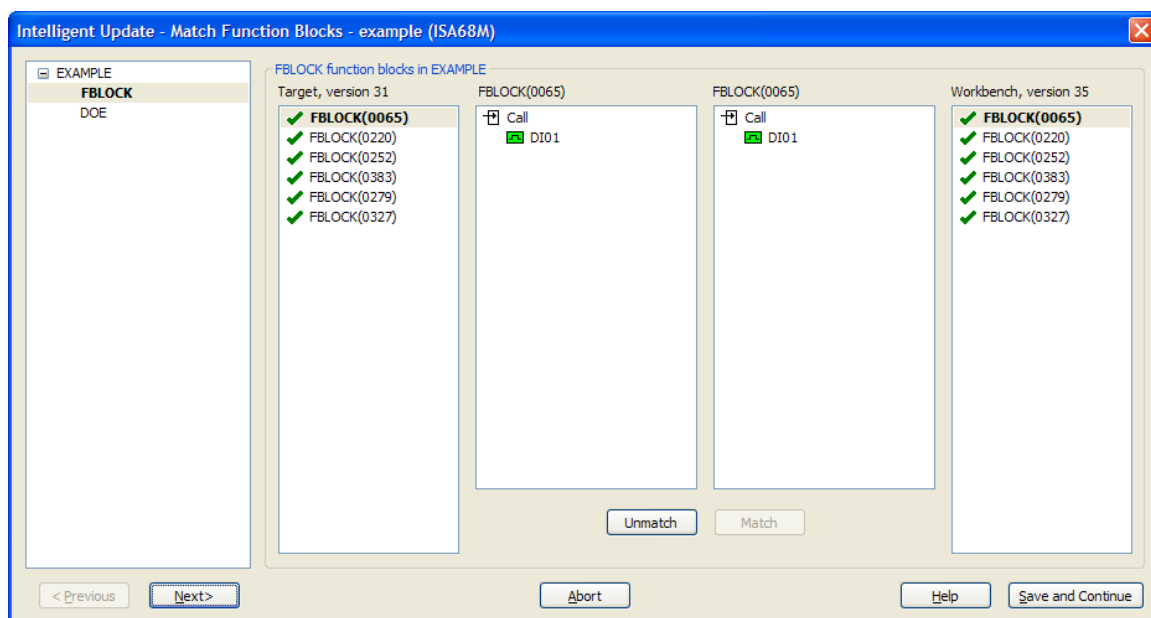


Figure 57 Intelligent Update Options

## 7.2. Manual function block matching (from release 3.5.1)

Function blocks are identified by the Online Update manager using the names of the variables wired to their inputs. Where the inputs match, the online update manager automatically creates a recipe to transfer the values of local variables inside the function block to the new version for bumpless transfer. However, function blocks with inputs wired from other blocks or from corner connections cannot be uniquely identified, because these wires are given temporary names which may change on each compilation. These are reported as 'DELETED' and 'NEW' before release 3.5.1 and their local variables reset to initial values.

At release 3.5.1, a manual matching window is presented on compilation. This presents all function blocks which need the user to confirm a match.

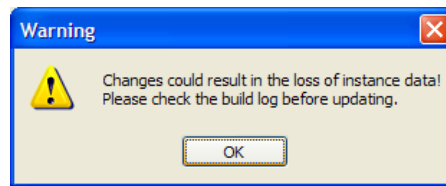


This window shows each type of function block in each program. 'Previous' and 'Next' page through each matchable group of function blocks. Each function block which needs manual matching is shown, with its 'instance number' (e.g. 0065 above) and its calling parameters, traced back to named variables. The instance numbers do not usually change, but they will change if any blocks are copied/cut/pasted or if standard blocks are inserted (e.g. AND, OR etc.). More information is given in Application Note AN-T80009.

Select the old function block on the list on the left for the existing running application, select its equivalent in the new application, and click 'Match' to inform the update manager that these are the same function block.

When all pre-existing blocks are matched, click 'Save and Continue'.

If there are any function blocks which are not matched (either automatically or manually), the following message appears:



For these function blocks, review the report that follows to check that these function blocks are genuinely deleted or new. This warning also appears at the end of the report.

### 7.3. Operation

When enabled, the Intelligent Update Manager runs each time an application is compiled and downloaded. Intelligent Updates work by comparing the most recently compiled application with a copy ('snapshot') of the application currently running in the controller. A list of variable values to be transferred ('recipe') is generated that the controller loads when realizing an application update. When an application is downloaded (either as the initial download or as an on-line update), the Intelligent Update Manager utility saves a snapshot of that application in a database for use the next time the application is compiled. This snapshot is necessary for the next Intelligent Update to work.

### 7.4. Important Information

The following information must be read and understood before working on a safety certified system.

Refer to sections 3.11 & 3.12 of the Trusted TMR System Safety Manual T8094; Publication [ICSTT-RM459-EN-P](#) (Legacy 555192). These sections describe the standards and procedures to be applied when programming and updating a safety system. They cover workbench configuration, language selection, formal testing, program design, communications interaction and online modification.

Refer to the Toolset User Guide section A.15.6. This section describes the preparation and application of updates where only the normal method is available, but the section on 'Operations' is relevant to both forms of update.

On line variable forcing and power flow monitoring of a running application (on a live system) must be carried out using the same version of the application as that running in the system.

Some changes to logic will cause function block internal variables to be reset to default values. This may cause plant shutdowns. From release 3.5.1, the user is able to match function blocks to avoid losing data. Before release 3.5.1, care must be taken with logic changes. This issue, and other online update issues, are explained in section 1.9 and 1.10 of the Application Note AN-T80009. Read these issues before performing changes.

The capabilities of intelligent online updates are described in section 1.7 and 1.8 of the Application Note AN-T80009. Pay particular attention to cases where values will not be preserved.

Note that SFC programs are **NOT** recommended for safety systems, and should not be used for systems needing Intelligent Updates.

For the on line update to function, the application size must be less than about 800K bytes to ensure that the application, the online update recipe and other data fit into the 1MB available. A rough guide to the application size is the size of the compiled appli.x6m file in the application folder, after completely compiling the code for the Motorola option.

## 8. ISA.INI settings

The Toolset EXE folder contains a system initialisation file called ISA.INI. This controls the operation of the Toolset programs. Many of the entries are simply recording the settings assigned in the Options menus in each Toolset sub-program. There are however some options which may prove useful to the programmer.

### 8.1. Decimal format for Modbus addresses

The following setting in the ISA.INI file allows the user to edit and display Modbus addresses in the Network Address field as decimal numbers rather than the default hexadecimal number format. Place the following setting in the section marked [EDIT] in the ISA.INI file.

[EDIT]

NwAddrDecimal=1

With this setting:

- a) When editing an individual variable definition in the Dictionary, you can enter normal decimal numbers for the Modbus address. For example, enter 40001 rather than the hexadecimal address 9c41.
- b) In the Dictionary display list of variables, the address column shows the values in decimal format.
- c) When you use the Export/Import tools in the Dictionary, the Network Address is imported and exported in decimal format, i.e. 40001 rather than 16#9c41.
- d) The source files for dictionary variables, APPLI.DCO, APPLI.DLO, still contain the Modbus Address in hexadecimal, so other software that uses these files (e.g. the OPC Server) remain unaffected.

### 8.2. Configure an Internal variable or Constant with an Initial Value and the Retain attribute

The following setting in the ISA.INI file allows the user to configure variables with an initial value, as well as the Retain attribute. Without this setting, the dictionary only allows a variable to have one or the other attributes, but not both. Place the following setting in the section marked [DEBUG] in the ISA.INI file.

[DEBUG]

AllowRetainInit=1

With this setting:

a) An "Internal" variable configured with an initial value and Retain attribute can have an initial value the first time you ever download and run the application. Any change to the value on-line will then be retained in the target if you ever reboot the target, or stop and start the application. However, if you ever start from scratch with a new processor, the original initial value defined in the dictionary will be used. Changes to the value of the "Internal" variable are retained in the processor, but are not saved back to the Toolset.

b) A variable can be defined as a "Constant" rather than an "Internal". When a Constant is configured with an Initial Value, and the Retain attribute, the Constant has an initial value the first time you ever download and run the application. When you change the value on-line, the value is changed in the Processor, and then the Toolset prompts you to save this change back to the Toolset. Saving this change back to the Toolset modifies the original Initial value defined in the dictionary. The changed value in the processor is retained if you ever reboot the processor, or stop and start the application. Since the change is also saved to the Toolset, if you ever start from scratch with a new processor, the last change you saved to the Toolset is used as the initial value when the application is downloaded and run.

### 8.3. Change the documentation logo

The logo printed on the toolset application documentation is stored in the toolset file `exe/printlogo.bmp`. This may be replaced with another appropriate monochrome bitmap. Review the following setting in the section marked [WSDG1EDT] in the ISA.INI file.

```
[WSDG1EDT]
```

```
Logo=C:\Trusted\Toolset\exe\printlogo.bmp
```

Changing this entry will redirect the toolset to print a different logo on the documentation.

## 9. System Capabilities

Maximum number of chassis 28

One processor chassis and four sets of seven expansion chassis.

Maximum number of modules 138

Not including the main processor or expander processor slots, i.e. 8 in the processor chassis and 12 per expander chassis. The number includes all active module slots, i.e. those not disabled in the system configuration.

Note that if more than 11 chassis are fitted (128 slots) then some slots must be disabled in the system configuration to reduce the total below 138. Disabling slots will also slightly reduce the scan time, although there should be spare slots available for diagnostic purposes. Do not disable slots used as smart slots.

Maximum variables in any one list 16,000 at release 3.5

This refers to the number of variables displayed in any one tab of the dictionary. The Extended Attributes compiler (VXA.DLL) has a limit of 16,000 variables per variable type. This means that it will accept no more than 16,000 Booleans, 16,000 analogues etc. Beyond this limit, the Extended Attributes compiler will fail to prepare the extended attributes data, including SOE collection, Modbus data presentation and Modbus write protection. This will also prevent intelligent online updates from working, although the application will still function but without the extended attribute functions.

Maximum system.INI size 64 kB

Maximum application download 850 kB at release 3.5 processor firmware  
960 kB at release 3.5.1 processor firmware

The application download size approximates to the size of the file appli.x6m in the application folder after compilation. This includes the object code and online update recipe. A 1 MB area (1024 kB) is available for the application and .INI downloads.

Maximum scan time Less than a quarter of the watchdog times

There is no fixed limit to scan times. Scan times of 500 ms are possible and are operational in the field. The watchdog times (maximum scan time, IMB timeout and Watchdog timeout) must be set to allow for delays during swaps and updates, as long as they do not exceed half of the effective process safety time as detailed in the Trusted TMR System Safety Manual T8094; Publication [ICSTT-RM459-EN-P](#) (Legacy 555192). Note that scheduled polling on an I/O module will affect the watchdogs on 8480 modules and the watchdog times on these modules should be adjusted. (see TN20057).

Minimum scan time 32 ms

This is set to allow room for communications traffic and background tasks. It is especially important on some hybrid systems where legacy I/O scanning is asynchronous to main processor scanning.

Scan time estimations are given in PD-8110B. Note that these tend to be pessimistic and are only estimates.

Maximum total variables of any type 65,535

This is described in TA20004 and a method for determining the limit is given in AN-80011. This covers all variables of any one type (Boolean, analogue, timer etc.) across the whole application, including internal temporary variables to reference wires and connection corners etc. (see AN-80009).

Maximum instance number 9999

Each FBD program has a numbering scheme in its source code (the application file *program.lsf*) for variables, functions/function blocks and wires. The numbers are allocated sequentially from the highest, and lower numbers are not reused. When the numbers reach 10000 the number rolls over and the online update manager cannot recognise them; it will always apply an online update and some function block local variables will be reset. AN-80009 describes a method to reset these numbers, essentially by cutting and pasting the logic.

Maximum FBD program size Approximately 75 % of diagram area

The maximum FBD program complexity (as measured by the size of the application file *program.lsf*) and the proportion of space used by the program on the diagram, do not have clear limits. At large sizes, a program will appear to compile properly but it is possible that not all wires on the diagram will work. It is recommended to keep the program to less than 75 % of the available diagram area, and to turn off power flow debugging on programs larger than 50 % of the area (because power flow debugging adds many temporary variables).

Maximum application or program name 8 characters

Maximum variable name 32 characters (at TUV 3.5 toolset)

16 characters (before TUV 3.5 toolset)

Maximum conversion table points 32 per table

Maximum conversion tables 127

Maximum programs in an application 255

Maximum levels of program hierarchy 20 (depth of subprogram nesting)

Maximum FBD diagram width 200 columns (see above for height)

Maximum I/O connection boards 256

Maximum function (block) parameters 32 (total of inputs and outputs)

The ISaGRAF users' guide lists other limits but those detailed above will be the first to be encountered.

## 10. Password Permissions

Many different functions may be protected by a password in the Toolset. The ISaGRAF User's Guide describes how to set these protections. The following is a list of some of the functions that may be protected.

**Global protection:** When opening project, this is the first protection met by users preventing the project from opening. It should be set to a lower priority user password to allow all with passwords to access the project e.g. protection level 15.

**Overwrite with archive:** Possibility to overwrite the existing copy of a project in workbench with archived one from the Tools/archive/projects menu.

**Backup on archive:** Possibility to create a backup copy of the project.

**Project Descriptor:** Possibility to change the project description in Project | Project descriptor.

**History of modifications:** Possibility to view (Read) or alter (Full) the history under Project | History of modifications.

**I/O connection:** Possibility to view (Read) or alter (Full) the I/O connection table from the menu Project | I/O connection.

**Global variables:** In the dictionary, possibility to view (Read) or alter (Full) the global variables.

**Global and common defined words:** In the dictionary, possibility to view (Read) or alter (Full) the defined words (both global and common).

**Create new program:** Possibility to add programs in the File | New menu.

**Move program in hierarchy:** Change the program's position with File | Arrange programs menu.

**Section: Begin:** Add/remove programs in that section.

**Section: Sequential:** Add/remove programs in that section.

**Section: End:** Add/remove programs in that section.

**Verify program:** Verification of programs from menu Make | Verify.

**Make application code:** Make a project from menu Make | Make application.

**Touch (delete generated code):** Delete the compiled code from menu Make | Touch.

**Conversion tables:** Possibility to view (Read) or change, add or delete (Full) conversion tables from menu Tools | Conversion tables in the dictionary.

**Application run time parameters:** Possibility to view (Read) or alter (Full) the runtime settings from menu Make | Application run time options.

Compiler options: Possibility to view (Read) or alter (Full) the compiler options from menu Make | Compiler options.

Resource definition: Possibility to view (Read) or alter (Full) the resource definition from menu Make | Resources. (This is not necessary for Trusted systems).

Debug application: Possibility to go into debug mode (Debug | Debug).

Simulate application: Possibility to go into simulation mode (Debug | Simulate).

Download/start/stop application: In the debugger, the ability to Download, Start or Stop the application currently running from the File menu.

Update application: Possibility to update the current application (online change) from the File menu.

Communication parameters: Possibility to view (Read) or alter (Full) communications parameters from menu Options | Parameters in the debugger window. (This is not necessary for Trusted systems).

Set cycle timing: Possibility to set the project's cycle time from menu Control | Change cycle timing in the debugger window. (This is not recommended for live Trusted systems).

Set execution mode: Possibility to set the "real time", "cycle to cycle" and "execute one cycle" modes in debugger window. This is in the Control menu. (Live Trusted systems should be left in "real time").

Change variable state: Possibility to modify a variable's value in Simulation mode.

Lock/unlock variable: Possibility to lock or unlock variables in debug mode and Control | Unlock All I/O Variables in the debugger window.

Control SFC program: Possibility to Start, Kill, Freeze or Restart SFC programs from the File menu in debug mode.

Control timer variable: Possibility to start and stop a timer variable in debug mode.

Set IL breakpoints: Possibility to add or remove breakpoints in debug mode from the Edit menu for an Instruction List program (if these menu items are enabled in the program's options).

Set SFC breakpoints: Possibility to add or remove breakpoints in debug mode from the Control menu for a Sequential Function Chart program.

Create graphics: Possibility to view (Read) or alter (Full) Spotlight graphics from the Tools | Spotlight menu in debug mode.

Spy lists: Possibility to view (Read) or alter (Full) Spy lists from the Tools | Spy lists menu in debug mode.

Print project document: Possibility to print a document from the Project | Print menu.

Customize project document: Possibility to change the items that are printed from the document generator.

Program <name>: Possibility to view (Read) or alter (Full) the given program.