

# Allen Bradley SLC-500

Level 2 – Advanced Concepts

**EthernetSupport.com**

*“Customized Automation Training”*



Copyright (c) 1999 Ricky Bryce.

Permission is granted to copy, distribute verbatim copies of this document, commercially or non-commercially with no front cover texts, or back cover texts. Changing this document is not allowed.

Disclaimer:

This document is written in the hope that you can utilize for your own education to gain knowledge of PLC systems (should you decide to utilize this document).

Although I believe the information in this document to be accurate, it is YOUR responsibility to verify this information before implementing it in any way, especially when damage to personnel or equipment could result.

By continuing to read this document, you agree to hold no one who writes, modifies, or distributes this document liable in any way (even negligence).

Due to the wide variety of plant applications, some of the examples in this document may be prohibited at your location, or could cause damage to equipment, or harm personnel.

About the Author:

I'm Ricky Bryce, and I'm employed as an instructor of Automation Training, Inc (ATI) since April, 2004. This document is a collection of texts and graphics I've put together over the past few years, and has been distributed under the GFDL since 1999.

I hope you get much use out of it, and I would like your feedback as to how this document can be improved.

As a supplement to this document, I would like to invite you to my website at <http://www.LearnAutomation.com>. I'm in the process of uploading documentation and videos that will further help you with problems or questions you have with Allen Bradley processors.

You are also invited to visit my employers site at <http://www.atifortraining.com>.

If you are interested in on site training, please contact my employer at **(866) 573-9849**

"Human Knowledge Belongs to Everyone"



## Table of Contents

Questionnaire for Allen Bradley.....	9
PLC/SLC Level 2.....	11
Glossary.....	15
Numbering Systems.....	19
Hardware .....	20
Discrete Input Modules.....	20
Discrete Output Modules.....	21
Analog Modules.....	22
The Chassis.....	24
The Power Supply.....	26
The Processor.....	27
The Scanner Module.....	30
Addressing in the SLC.....	32
Configuring the DF1 Driver.....	33
To go on line, you must go to RSLogix at this point.....	35
RSLinx.....	36
Utilizing the BootP/DHCP Server.....	36
Setting up the Ethernet Driver.....	38
Backup Restore Utility.....	41
Creating a new project.....	43
Remote I/O Concepts for the SLC.....	49
terms.....	50
Physical Topology.....	51
Rack Number.....	52
Starting Module Group.....	53
Baud Rate.....	54
Image Size.....	55
1- Slot Addressing.....	56
2-Slot Addressing.....	56
½- Slot Addressing.....	56
Addressing.....	63
Adding Ladder Logic .....	64
Finding Errors.....	69
Branching.....	71
Basic Instructions.....	74
Timers.....	77
Timer On Delay (TON).....	77
Timer Off Delay (TOF).....	78
Retentative On Delay Timer (RTO).....	79
Counters.....	80
Working With Analog.....	82

Discrete vs. Analog.....	82
Compare instructions.....	85
.....	86
PID Demonstration—SLC 500.....	86
Proportional Gain .....	91
Integral .....	95
Derivative.....	100
.....	103
Documentation.....	104
Symbols.....	104
Descriptions.....	110
Rung Comments.....	113
Page Titles.....	114
Advanced Diagnostics.....	116
Importing and Exporting the Documentation Database.....	119
Changing the View Settings.....	123
.....	129
On line Editing.....	130
Start Rung Edits.....	131
Make Changes.....	132
Accept Edits.....	133
Test Edits.....	134
Assemble Edits.....	135
Forcing I/O.....	136
Using the Histogram.....	139
Custom Data Monitor.....	143
Trend Chart.....	148
Custom Help Screen.....	153
Hide and Unhide Program and Data Files.....	159
Project Compare Utility.....	160
Printing Reports.....	165
GNU Free Documentation License.....	175
GNU Free Documentation License.....	175
How to use this License for your documents.....	179

Name \_\_\_\_\_ Date \_\_\_\_\_

***Questionnaire for Allen Bradley  
Automation Systems***

- 1)What is the primary purpose you are attending this class?
  
- 2)Are you interested in programming, troubleshooting, or both?
  
- 3)What do you find most difficult about Allen Bradley PLC's
  
- 4)How often do you access the Allen Bradley PLC? (once a day, once a week, once a month, etc?)
  
- 5)After taking this class, will you be putting your knowledge to use right away in the plant?
  
- 6)What type of equipment do you generally work with?
  
- 7)What types of networks are you using with your PLC system? Ie... Data Highway plus, controlnet, devicenet, Ethernet, etc?

8)What is your company's policy on forcing?

9)Do you generally have access to the Internet as you work?

10)Can you bring a copy of some plant programs into the classroom tomorrow?

11)Will you ever be installing new systems, or checking new systems once they have been installed?

12)Will you ever be modifying the I/O structure of existing systems?

13)Do you have any common system failures that are related to the Allen Bradley PLC? If so, what are these failures

14)Are you interested in learning features of RSLogix that are not currently in use by your plant, but, if used could reduce downtime?

Name \_\_\_\_\_ Date \_\_\_\_\_ Score \_\_\_\_\_

***PLC/SLC Level 2  
PreTest 20 Minutes***

- 1) What is the definition of a **BIT** of memory?
  
- 2) How many bits are in a **WORD** of memory?
  
- 3) What are some examples of what a TON (Timer On Delay) instruction might be used for?
  
- 4) In your plant, where are some examples you might use a CTU (Count Up Instruction)?
  
- 5) How many bits of memory are generally needed to store an **ANALOG** value?
  
- 6) What is the purpose of **Data Files**?
  
- 7) What are **Program Files** used for?
  
- 8) What is the purpose of RSLinx software?
  
- 9) What is the purpose of RSLogix software?

- 10) Give an example of a system where a PID loop might be used?
- 11) The 'P' in PID is for Proportional. This proportional influence is determined by what component of your loop?
- 12) What is Integral influence based on in a PID loop?
- 13) What determines how much Derivative influence the PID loop will have on the output?
- 14) If the processor fault light is blinking, what should you do before clearing the fault, and taking the processor back to run mode?
- 15) What instruction must be placed in ladder 2 so each of the subroutines will execute?
- 16) If you have a motor that will not run when the operator pushes a start button, you may need to go on line with the processor and locate the output in the PLC program that energizes the motor. This allows you to see what other conditions must be met before the processor will energize the output. What features of RSLogix might you use to locate the motor's output in the PLC project?
- 17) In #16, you find that the reason for the failure is a bad thermal switch on the motor. Since you do not have any more thermal switches in stock, you decide to force the motor's output instruction on. What are some dangers with this, and what are some better options that you could have chosen.

- 18)The processor is in Remote Run mode. You perform an on line edit and inadvertently jump to a routine that does not exist. What will happen the instant your edits are tested?
- 19)You suspect an intermittent problem with a limit switch on a safety gate. The intermittent problem is periodically shutting a system down. What feature of RSLogix software might you use to confirm or rule out what you suspect?
- 20)An operator informs you of variations in a product. You suspect the variations in product are due to a fluctuation in temperature. What feature of RSLogix software would allow you to graph a temperature over time?
- 21)What will happen to your SLC chassis if you place more modules in the chassis than the power supply can handle?
- 22)If a fuse blows on a power supply, what does this usually indicate?
- 23)What use would an MMI device have on PLC controlled equipment?
- 24)What cable would you use if you need to hook your laptop to channel 0 of the SLC processor (assuming channel 0 is at it's default protocol).
- 25)What are the advantages of using Ethernet to communicate with your SLC vs DF1 (RS232)?.



# Glossary

## Addressing:

**Bit:** Smallest unit of information the PLC can process– ON or OFF

**Word:** Memory location consisting of 16 Bits of information

## Hardware:

**Input module:** Reads the STATUS of field devices

**Output module:** CONTROLS field devices

**Discrete module:** Reads or controls devices which only have 2 states:  
on or off

**Analog Module:** Reads or controls devices which have a range  
such as 0 to 10 volts or 4 to 20 milliamps

**Power Supply:** Provides control power to modules on the backplane

**Chassis:** The physical device that modules are plugged into.

**Local Chassis:** The chassis where the processor resides.

**Processor:** The 'Brain' of the PLC which contains the machine program.

## **Troubleshooting Tools:**

### **Cross Referencing:**

Allows the troubleshooter to quickly navigate through the program by listing all locations in ladder logic where a particular address is located.

Usually the troubleshooter will cross reference a false condition on a rung of logic to find the output that will turn the referenced bit on.

You can Access Cross Referencing by right clicking a particular address. (Note: You must be on the address, not the instruction)

### **Custom Data Monitor Utility:**

Allows the troubleshooter to gather data from various memory locations onto one screen for easy troubleshooting. For example: One can create a custom data monitor for the failure of a particular motor. Next time the motor fails, the troubleshooter can simply look down the list of conditions that must be met, and see in real time which condition is causing the failure.

### **Force:**

Simulates real world jumpers. Use care while performing a force. You must understand fully how a force is going to affect your system. In most cases, only addresses starting with an I: or an O: can be forced.

To force an input or output, you can right click on the address in logic, then choose force on or force off. After the force is installed, forces can then be enabled from the on line tool bar.

### **Trending:**

Trending acts somewhat like a 'software chart recorder', and allows you to track an analog signal over time. Trending can be found toward the bottom of the project tree.

### **Communication terminology:**

**RSLinx:** This is the communication server. If RSLinx is not set up properly, RSLogix will not communicate to the processor.

**Driver:** Allows RSLinx to communicate with a particular hardware device. The most common drivers are the DF1 driver to communicate with Channel 0, the PCMK driver to communicate with channel 1. Configure drivers by clicking communication on the menu bar.

**RSWho:** A graphical screen which will display what devices RSLinx has established communication with. Access RSWho by clicking communication on the menu bar. Then click on the name of the driver you wish to use for communication. The right hand side of the screen will reveal devices the driver has communication with.

**RSLogix:** The software which allows you to troubleshoot or program a processor.

**On line:** Actively communicating with the processor (ladder spinning)

**Download:** If a program was changed off line, it must be downloaded (or sent to) the processor. When downloading the processor must be in program or remote program mode. A good way to download once RSLinx is properly set up is to click COMMS on the menu bar, and then go to Who Active. Click the driver name, highlight your processor, then click DOWNLOAD.

**Memory Layout:**

**Data Files:** A section of the processor's memory that stores information. You can also think of Data Files as variables.

**Program Files:** A section of the processor's memory which holds programs. Programs hold Routines. Logic in Program Files is used to manipulate Data Files.

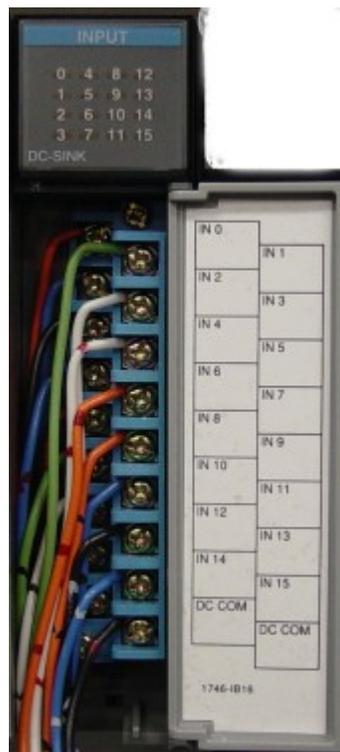


## ***Hardware -- Discrete Input Modules***

The purpose of the discrete input module is to read the status of field devices. When a voltage is detected on the terminal of an input module with respect to common, the corresponding status light is energized, and during the processor scan, the value of 1 is placed into the input data table. Examples of input devices include switches, pushbuttons, or auxiliary contacts on a motor starter.

Please answer the following questions:

- 1) What is the catalog number of your DC Input module?
- 2) Name at least three field devices that can be connected to the DC Input module?
- 3) What do the status lights indicate on the front of the DC Input Module?
- 4) What slot number is your DC Input module located in?

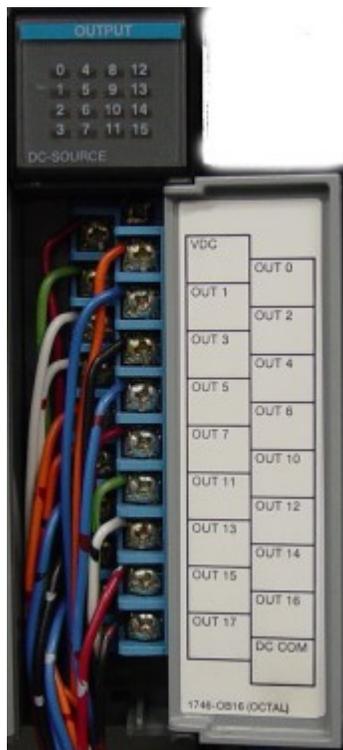


## *Discrete Output Modules*

The purpose of the discrete output module is to control field devices. The discrete input module requires power from an external source. When a 1 is placed into the output table of the PLC (in run mode), a status light is energized on the module, and a connection is made between the source, and the corresponding output terminal. Examples of output devices include: lights, solenoids, motor starter coils, and contactors. If you have an inductive load as the output, be sure to use the proper surge suppression.

Please answer the following questions:

- 1) What is the catalog number of your DC Output module?
- 2) Name at least three field devices that can be connect to the DC Output module:
- 3) What do the status lights indicate on the DC Output module?
- 4) What slot number is your DC Output module located in?
- 5) If the load on the DC output card is inductive, what should be done across the load to minimize the effects of inductive kick?



## *Analog Modules*

Analog modules are used to control and read the status of analog devices. Analog devices have a range of states instead of just on/off states like discrete devices.

Some analog modules have switches which determine whether the input channels are to be set up for voltage or current. Some analog modules are configured through software.

Examples of analog inputs include: Potentiometer, Pressure Transducers, Variable speed drives, and with a thermal couple module, temperature can be read into the processor's memory.

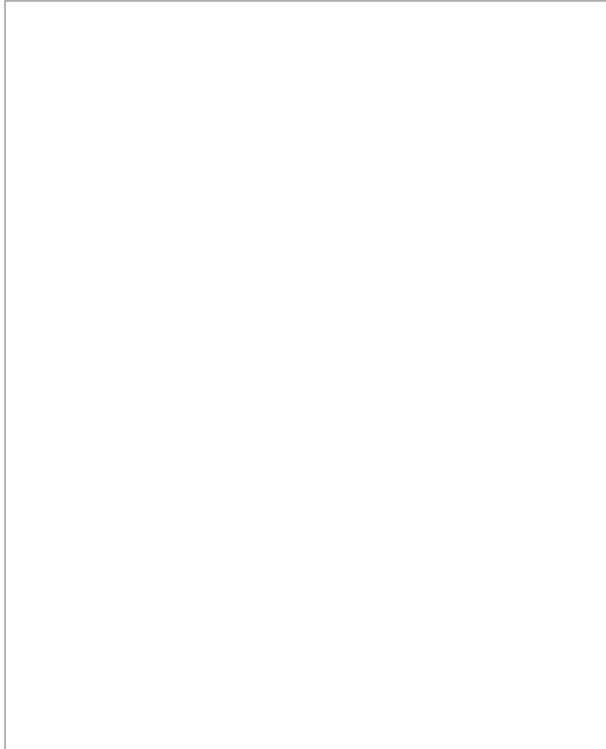
Examples of analog outputs include: Meters, Variable Speed Drives, Valve Positioners, and chart recorders.

An analog signal cannot be expressed with a single bit, and therefore analog values will consume a word of memory.

Please answer the following questions:

- 1) What is the catalog number of your analog module?
- 2) How many channels of Input, and how many channels of Output are available on this module?
- 3) How do you set up the input channels to accept either a current or a voltage input?
- 4) What range voltage or current will the Input channels accept on your module? What range of voltage will the output channels accept?
- 5) Name at least three devices that are analog inputs:
- 6) Name at least three field devices that are analog outputs:
- 7) What slot number is your analog output module located in?

## **An Analog Module – 1747-NIO4V**



## *The Chassis*

The chassis is the device which holds modules. Allen Bradley makes the SLC chassis available in 4, 7, 10, or 13 slots, and the PLC-5 chassis in 4, 8, 12, or 16 slots. For the SLC, the processor is included in the slot count. For the PLC-5 chassis, the processor is not included in the slot count.

Both the SLC and the PLC support extended local I/O. This is more common in the SLC, however. For the SLC system, a maximum of 3 chassis can be connected together using extended local I/O, but not to exceed the valid slot count of 0 to 31. The 1746-C7 cable can be used to connect two chassis together which are mounted horizontally (side by side), and the 1746-C9 cable is longer for chassis that are mounted vertically (one above the other).

Here are some chassis:

The PLC-5 Chassis (With modules): (For dip switch settings on this chassis, refer to page 4-1 and 4-2 of the PLC-5 Quick Reference Guide.)



The SLC-500 Chassis (With modules):



The ControlLogix Chassis (With Modules):



The Flex Chassis:



## *The Power Supply*

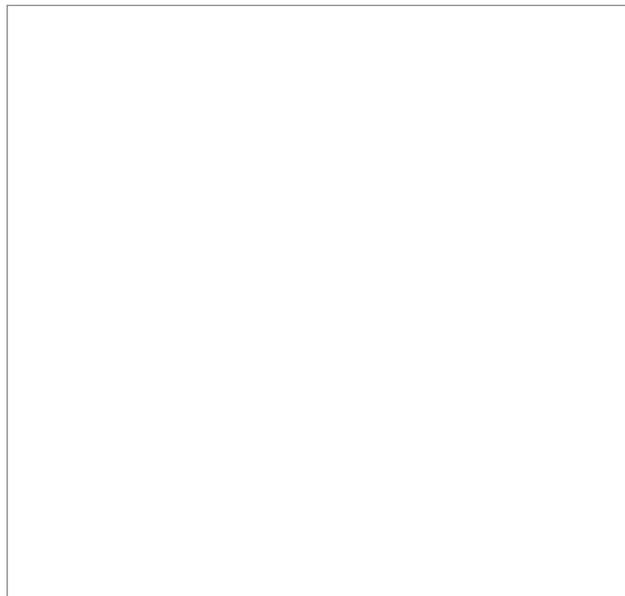
The power supply supplies power to the modules on the backplane. Generally power from field devices DOES NOT come from the power supply. The power supply only provides control power to modules on the backplane. Power for field devices come from a separate source which is connected to the output module. The power supply merely provides the power needed to shut a contact, or fire a triac or transistor to pass power from this external source to the field device.

The power supply has a fuse which protects the AC side of the power supply. If this fuse blows, the power supply is probably defective and in need of repair.

You will also find a jumper on the power supply to switch between low and high voltage. Be sure this jumper is set up properly.

Please answer the following questions:

- 1) Where does power come from to power field devices such as solenoids, lights, and motor starters?
  
- 2) What does a blown fuse indicate on a power supply?
  
- 3) If you get a new power supply out of the stock room, what must you check on the new power supply before putting it in your system?



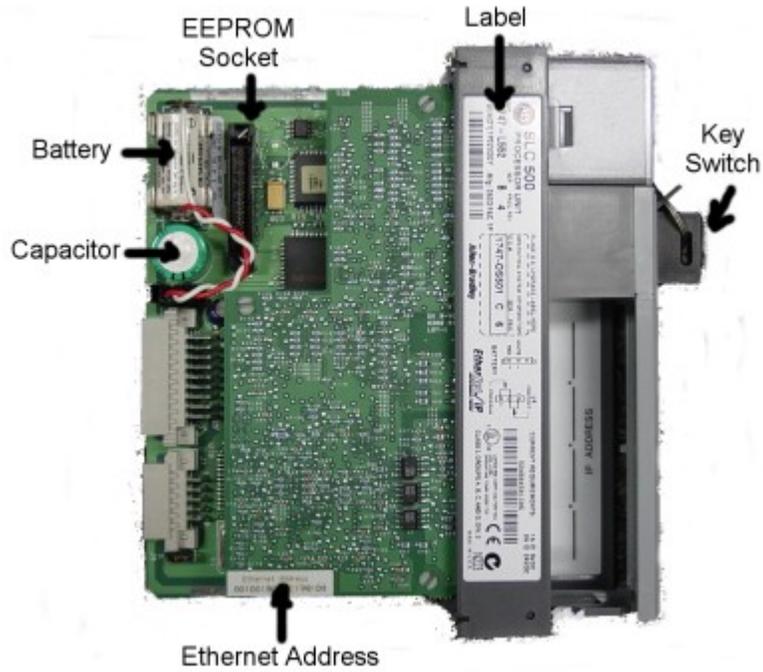
## *The Processor*

The processor is the main part of your SLC 500 system. The processor is where the program is stored that reads the status of your equipment, and based on certain status, makes a decision on what to control. For example: The processor is reading the status of a switch. When the operator energizes the switch, the processor might call for solenoid to energize that extends a cylinder. When the cylinder reaches the end of its travel, it might close a limit switch. The processor will see that a limit switch has been closed, and shut off the solenoid.

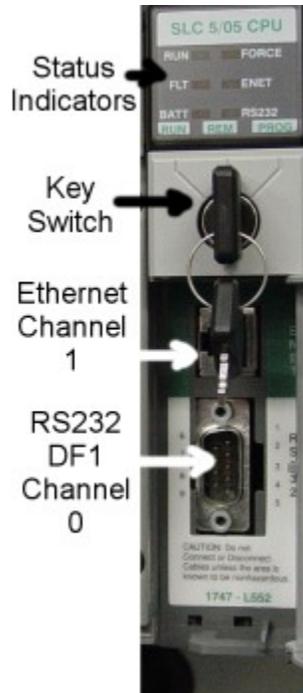
The processor consists of several components:

- 1) The battery: The battery retains the processor's program when the PLC is powered down. Certain AB documentation states that the shelf life of the battery is up to 2.5 years. When the battery is low, you will see a BATT light on the front of the processor. A status bit is also set in the memory of the processor (S:5/11 for the SLC). Once you have an indication of a low battery, change the battery within one to two days. If the battery is on the front of the processor, change the battery while the processor is powered up. If the battery is on the back of the processor (5/03 and higher), you must power down, and remove the processor from its chassis. Once the old battery is removed a capacitor will hold the processor's memory for up to 30 minutes. This gives you time to connect the new battery before losing your program. If you desire to reset the processor back to factory defaults (clear the program), disconnect the battery, and on the back of the main board, you will find two terminals that must be shorted for 1 minute: VBB and GND.
- 2) Next to the battery, you will find a socket for an EEPROM. The EEPROM is an on board backup of the processor's memory, and must be purchased separately. Changes to the PLC program do not automatically go to the EEPROM chip. You must manually store changes to the EEPROM from the COMMS menu in RSLogix.
- 3) On the front of the processor, you will find several status lights:
  1. RUN – Indicates when processor is in RUN Mode
  2. FAULT – If flashing, usually indicates a software problem. Go on line to get a description of the fault. You will find the description in the S2 status file on the ERRORS tab. If the fault light is solid red, this could indicate a hardware problem. You can try the following: reseat the processor, replace eeprom, clear memory and reload program, or replace processor.
  3. BATT-- Indicates the battery is low or missing
  4. FORCE – If flashing indicates forces are installed but not enabled... If solid indicates forces are installed, and enabled in the processor.
  5. Communication lights-- You may also have some indicators to indicate when communications are active depending on the type of processor you have.

**The 5/05 processor:**



**Shown below with ports at default settings:**

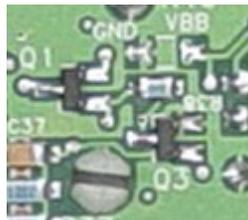
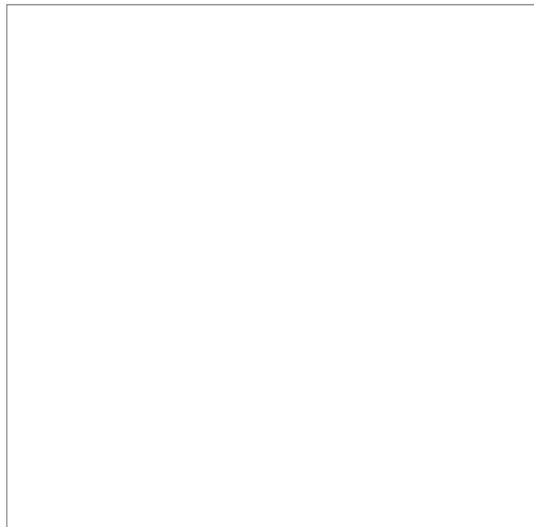


## Factory Default the processor:

On the 5/03 and higher, the processor will hold it's RAM for about 30 minutes (could be less, most of the time much longer). To get the processor back to factory defaults, you can perform the following actions:

- 1) Disconnect the battery on the processor.
- 2) Short VBB to Ground, for 60 seconds.
- 3) Connect the battery and power up the processor.

VBB and Ground are shown below:



## *The Scanner Module*

The scanner module is used to interface the SLC system with Remote I/O. Remote I/O is a legacy protocol used to interface with remote devices such as robots, drives, or other chassis. Remote I/O uses the Belden 9463 blue hose. (Allen Bradley catalog # 1770-CD)

The Remote I/O Scanner supports three baud rates: 57.6k, 115.2k, and 230.4k. These baud rates can be configured by DIP switches on the module itself. Allen Bradley's specification is that the blue wire of the belden cable is connected to terminal #1 of the three pin phoenix connector.

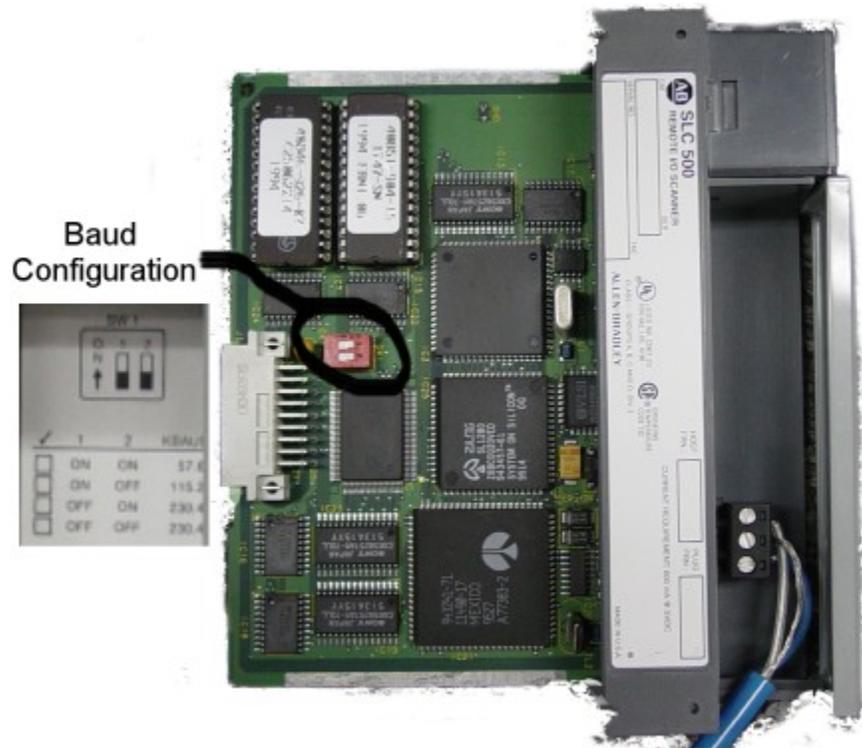
The scanner has two status lights:

**The FAULT LED:** If the Fault LED is flashing red, this usually indicates a configuration error on the scanner. Check the G file configuration. If the Fault LED is solid red, the scanner should be replaced.

**The COMM LED:** If the Comm LED is flashing green, the scanner is not communicating with one or more devices. If the Comm LED is flashing red, the scanner has lost communication with all devices. If the Comm LED is solid red, the scanner should be replaced.



The dip switches must be configured on the 1747-SN module for the correct baud rate. Baud is a term to describe how many bits per second are transferred over the network. Below is a photo of the switches. You will find a sticker inside the door of the scanner displaying the correct switch configuration for each available baud rate.



<i>Switch Position (1 and 2)</i>	<i>Baud Rate (Bits Per Second)</i>	<i>Max Network Distance</i>	<i>Value of Terminators</i>
On, On	57.6k	10,000'	150 Ohms
On, Off	115.2k	5,000'	150 Ohms
Off, On	230.4k	2,500'	82 Ohms

## *Addressing in the SLC*

An Address in the SLC simply points to a memory location in the processor. Some memory locations reflect the state of real world devices, such as inputs and outputs, and some memory locations are internet. Memory locations are basically broken down like this:

File (Reflects the File in the PLC)	Element (Reflects the Slot number of the module)	Word (Group of 16 bits)	Bit or Terminal			
<b>I</b>	<b>:</b>	<b>6</b>	<b>.</b>	<b>0</b>	<b>/</b>	<b>15</b>
	Element Delimeter		Word Delimeter		Bit Delimeter	

Internal memory locations are addressed in a similar manner:

Here is a bit from the binary file:

File	Element	Bit
<b>B3</b>	<b>:</b>	<b>0 / 6</b>

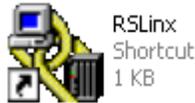
Or for a timer...

File	Element	Bit
<b>T4</b>	<b>:</b>	<b>6 / DN</b>

## Configuring the DF1 Driver in RSLinx

The DF1 Driver is used for point to point communication over RS232 between a COM port on a PC, and the serial port (Channel 0) of a processor. The following steps will take you through a sample configuration of the DF1 RS232 driver.

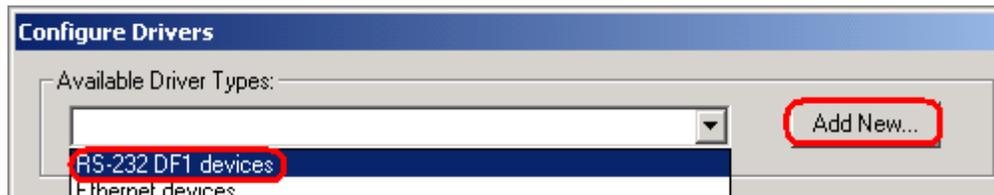
- 1) Open RSLinx Communication Server. If there is no short-cut on the desktop, you can access RSLinx by clicking Start | Programs | Rockwell Software | RSLinx | RSLinx



- 2) Click 'Communication' on the menu bar, then choose 'Configure Drivers'.



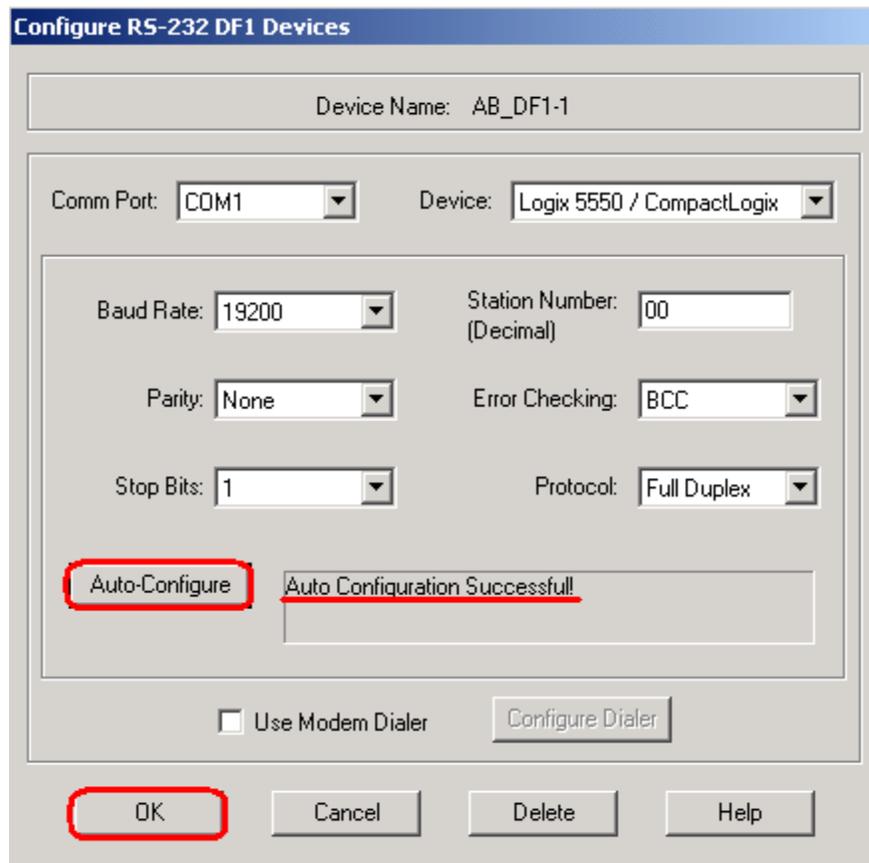
- 3) From the Available Driver Types pull down menu, choose 'RS232 DF1 Devices', then press the ADD NEW button.



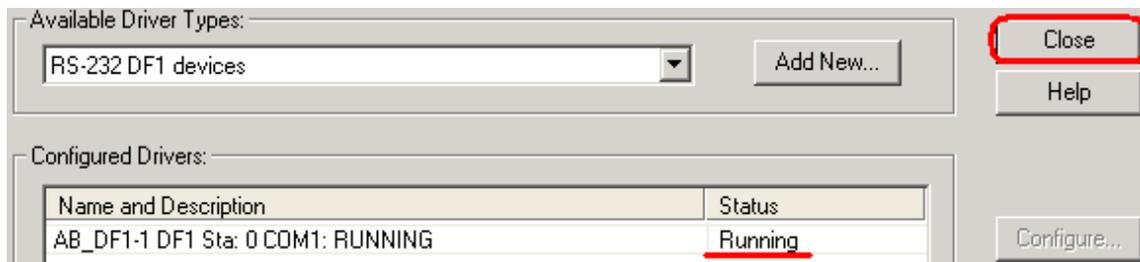
- 4) For this example, the name can be left at default. Press OK.



- 5) Although the communication parameters can be entered manually, if you are currently connected to the processor, just hit the 'AutoConfigure' button. RSLinx will hit the processor with different baud rates, and different settings, until it finds a setting it gets a response on. When this happens, you will get a message that the autoconfiguration was successful. Press OK when finished.



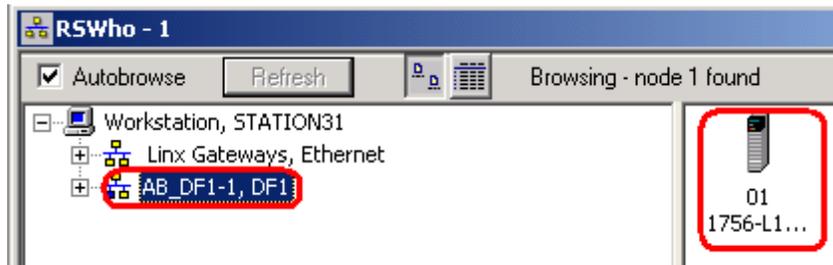
- 6) You will see the driver is now running. Close the "Configure Drivers" screen.



7) To test your drivers, click the RSWho icon in the tool bar of RSLinx.



8) Click the DF1 driver (the one you just configured) on the left side of your screen. The devices you are communicating with will appear on the right.



To go on line, you must go to RSLogix at this point.

## ***RSLinx -- Utilizing the BootP/DHCP Server***

- 1) Write down the Ethernet Hardware Address of the device you wish to configure. This is also called the MAC address. Here is an example of a hardware address:

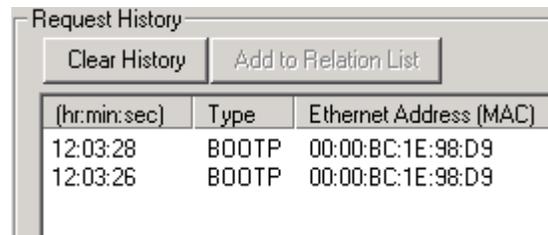
**00:00:BC:1E:98:D9**

- 2) Run the BootP/DHCP utility. You can access this utility if it is installed by clicking Start | Programs (or All Programs in Windows XP) | Rockwell Software | BootP/DHCP Server | BootP DHCP Server.

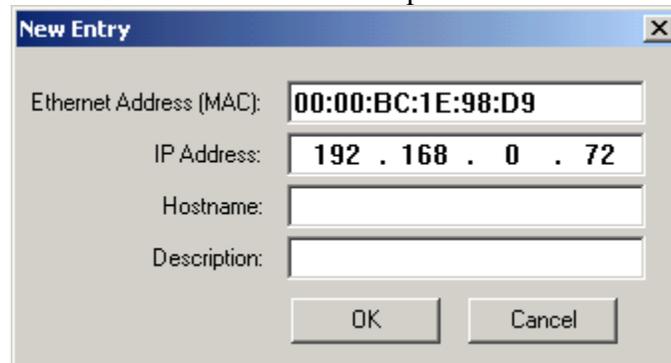


- 3) Once the server is open, it may ask for some specific network information if this is the first time the BootP server has been run. You can obtain this information from your network administrator. For this example, we just set the subnet mask to 255.255.255.0 then click OK.

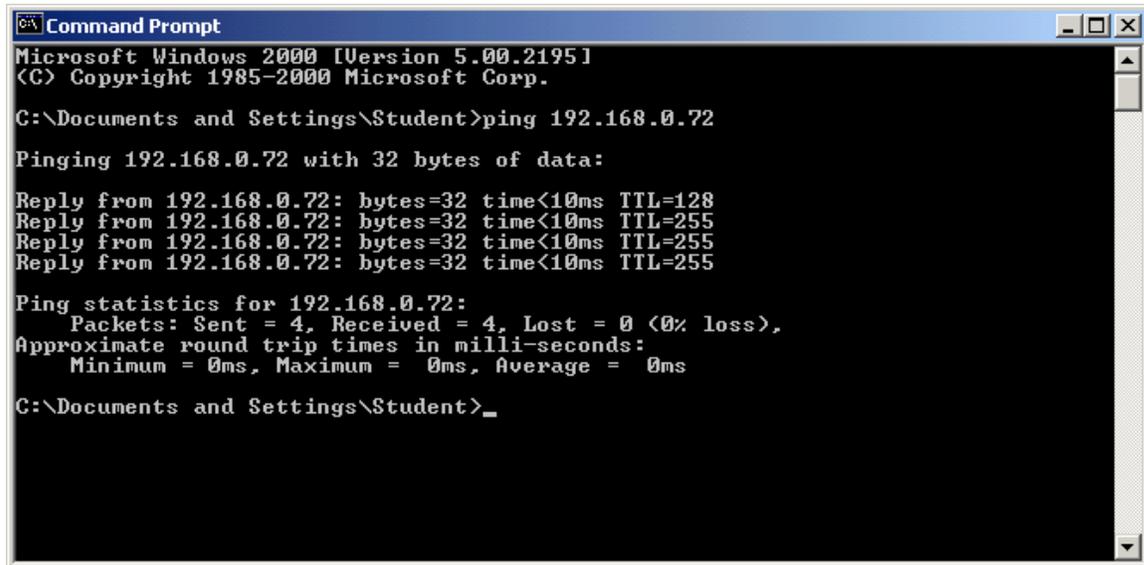
- 4) Power up your processor, and you should see the Ethernet device begin to request and address.



- 5) Double click on the device. You will then be prompted to assign an IP address to the device. Be sure you double click the right device. Entering an IP address into the wrong equipment could have disastrous consequences.



- 6) You could also enter a host name and description at this time if you wish.
- 7) If you wish to verify communication, you can ping the device from the command prompt. By default, the command prompt can be accessed from Start | Programs | Accessories | Command Prompt.



```
Command Prompt
Microsoft Windows [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Student>ping 192.168.0.72

Pinging 192.168.0.72 with 32 bytes of data:

Reply from 192.168.0.72: bytes=32 time<10ms TTL=128
Reply from 192.168.0.72: bytes=32 time<10ms TTL=255
Reply from 192.168.0.72: bytes=32 time<10ms TTL=255
Reply from 192.168.0.72: bytes=32 time<10ms TTL=255

Ping statistics for 192.168.0.72:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Student>_
```

You should get replies. To ping continuously, use the -t flag after the ping command. A control-C will stop the ping command.

## Setting up the Ethernet Driver in RSLinx

The Ethernet driver is used to make a connection to Ethernet Devices, such as an Ethernet PLC-5, or a ControlLogix system. The following steps will walk you through a sample configuration of the Ethernet driver in RSLinx.

- 1) Open RSLinx communication server



- 2) Click 'Communication' on the menu bar, and then choose 'Configure Drivers'.



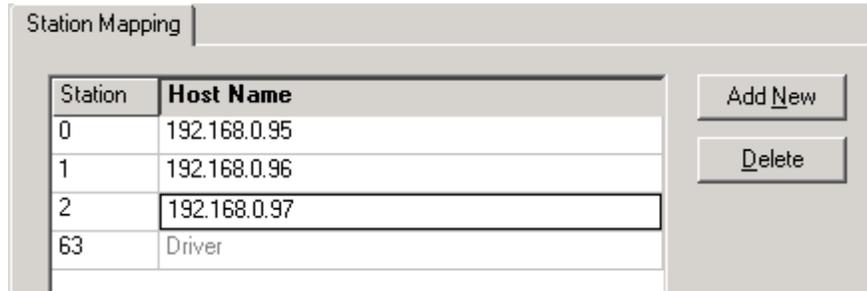
- 3) From the Available driver types pull down menu, choose 'Ethernet Drives', then press the 'Add New' button.



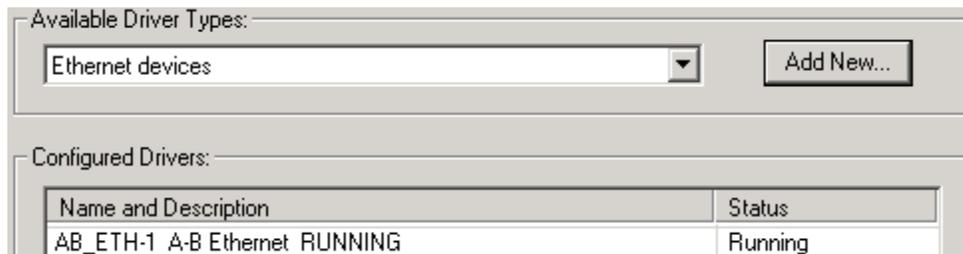
- 4) For this example, the name can be left at default. Press OK.



- 5) Populate the list of host names. If you do not have a way to resolve host names, you can enter the IP address of the devices you wish to connect to (as shown below in the example). The IP address for each device can usually be obtained from the network administrator, drawings, the off line project, or in some cases, the IP address is displayed on the front of the module.



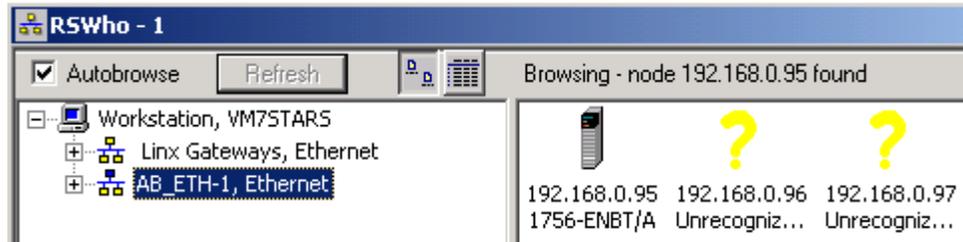
- 6) Press Apply, then OK. You will see the driver is now running. Close the 'Configure Drivers' screen.



- 7) To test your drivers, click the RSWho icon in the tool bar of RSLinx.\



- 8) Click the Ethernet Driver (the one you just configured) on the left side of the screen. The devices you are communicating with will appear on the right. In this example, 192.168.0.95 is a ControlLogix module, and the other devices are not present, or not a recognized PLC module.



- 9) To go on line with a PLC, you must go to RSLogix at this point.

## *RSLinx Backup Restore Utility*

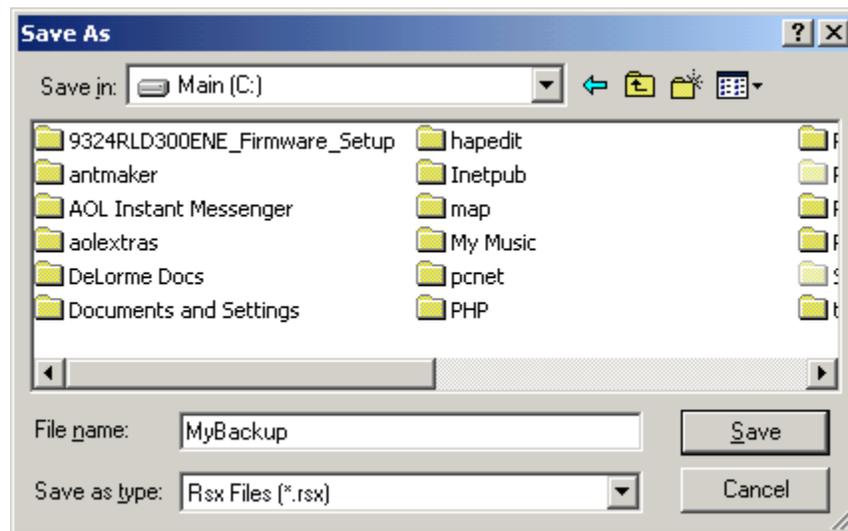
The RSLinx Backup Restore Utility can be used to backup the current driver configuration of RSLinx, or restore the configuration from a previous backup at an earlier time.

### Backing up the current Configuration:

To access the Backup Restore Utility, click Start | Programs | Rockwell Software | RSLinx | Backup Restore Utility.



- 1) Click the 'Backup' button.
- 2) A dialog screen will appear asking where you want to save the backup file. Choose a location from the pull down menu. If you are saving this to a floppy disk, choose the A drive. For this example, the driver configuration will be saved to the C: Drive. You must also enter a file name, and then press SAVE.



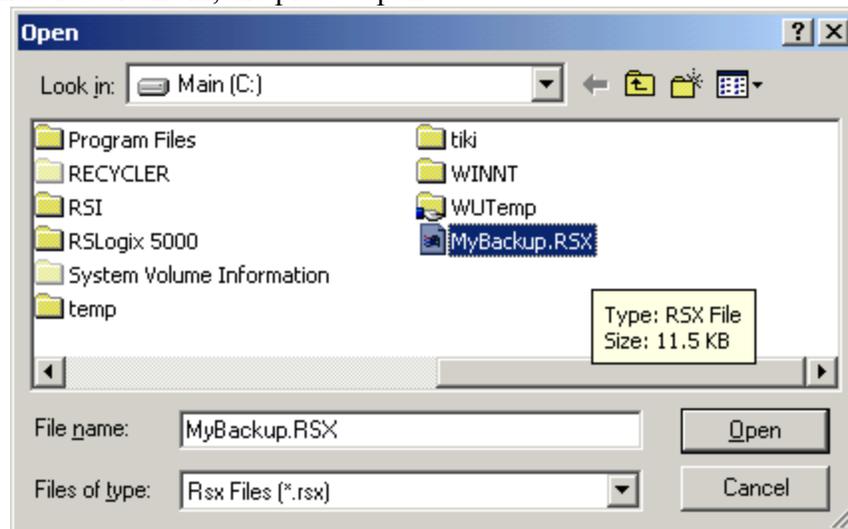
- 3) You will then get a message that the operation completed successfully. Press OK, then you can close the RSLinx Backup Restore Utility.

## Restoring a previous Configuration:

To access the Backup Restore Utility, click Start | Programs | Rockwell Software | RSLinx | Backup Restore Utility. Be aware that RSLinx must be shut down to perform this operation. If RSLinx is not shut down, you will be prompted accordingly.



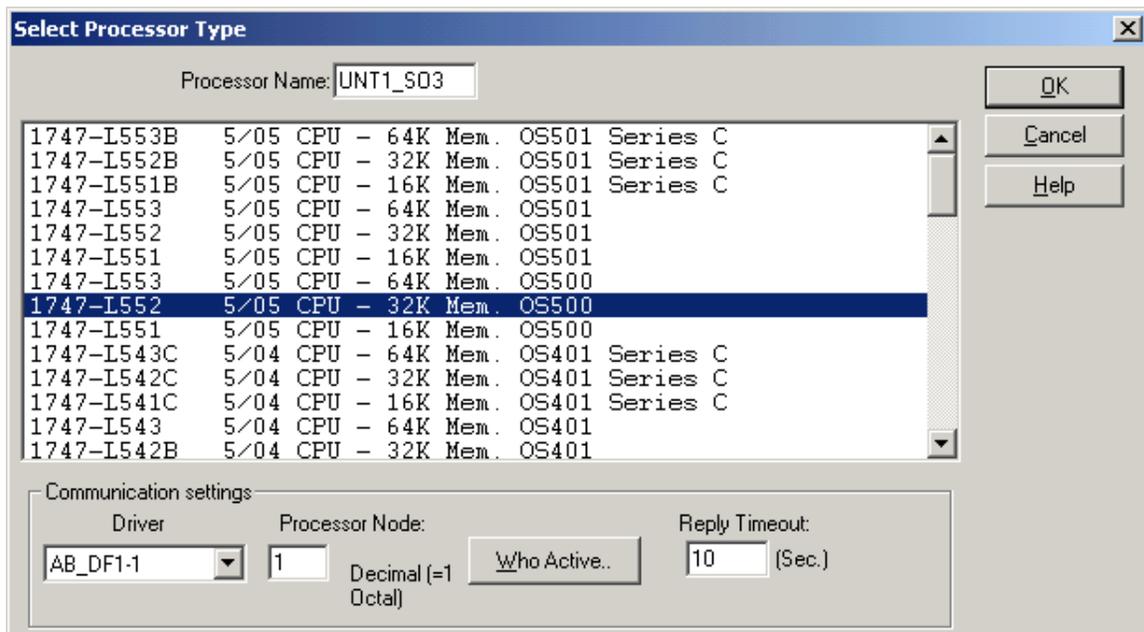
- 1) Click the 'Restore' button.
- 2) A dialog screen will appear asking where the backup file is stored at. Choose a location from the pull down menu. If the backup file was on a floppy disk, you would choose the A: drive. For this example, the backup is on the C: Drive. Click on the file you wish to restore from, the press 'Open'.



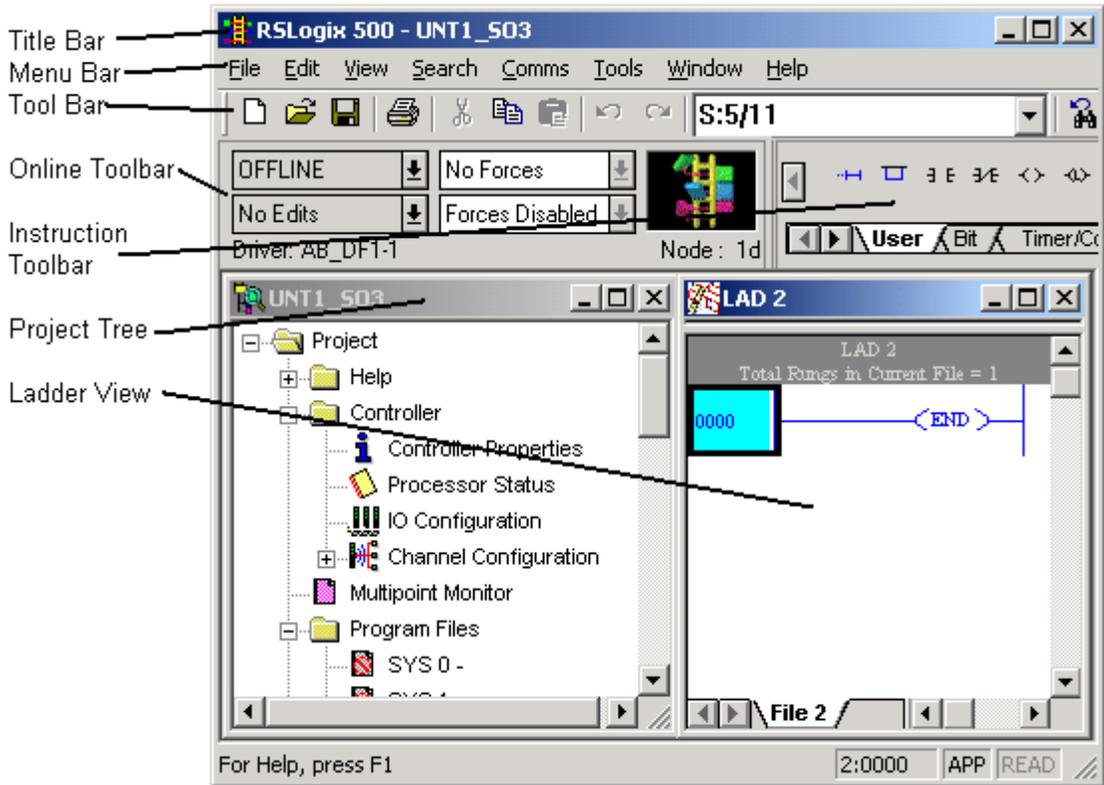
- 3) A dialog box should appear indicating that the operation was completed successfully. If you got an error, try the restore procedure again. In some versions of RSLinx, BRU must be ran twice if RSLinx was open.

## Creating a new project in RSLogix 500

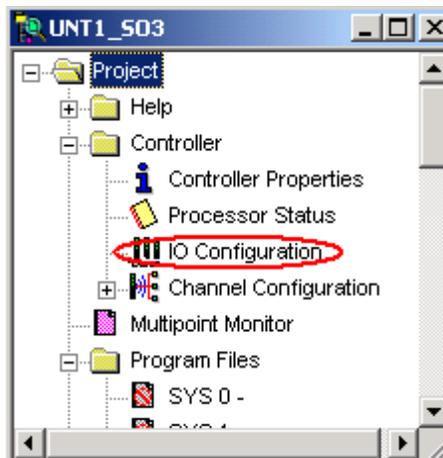
- 1) Open RSLogix 500. You can access RSLogix 500 by clicking Start | All Programs | Rockwell Software | RSLogix 500 English | RSLogix 500 English if there is not a short-cut on your desktop.
- 2) Once RSLogix is open, maximize the RSLogix window.
- 3) Click File | New from the menu bar.
- 4) Enter the name and type of your processor. For this example, the name is unt1\_so3. The processor is a 1747-L552. We are using the DF1 driver to Node 1. If you are unsure what the driver and node is to connect to the PLC, use the Who Active screen to help you. Press OK when finished.



5) Now the project is open. When using this documentation, or the help file that is built into RSLogix, you will need to know the various components of the RSLogix 500 application.

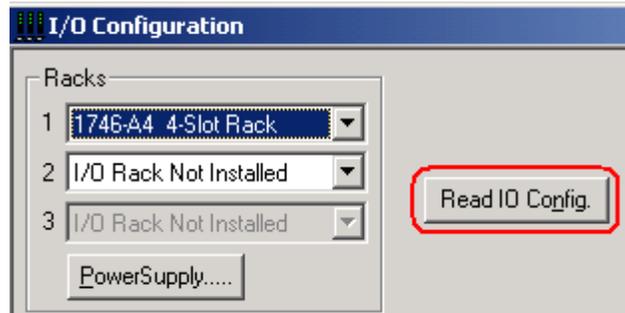


6) When creating a new project, there are two things you should set up before writing the ladder logic: I/O Configuration, and Channel Configuration (if necessary).

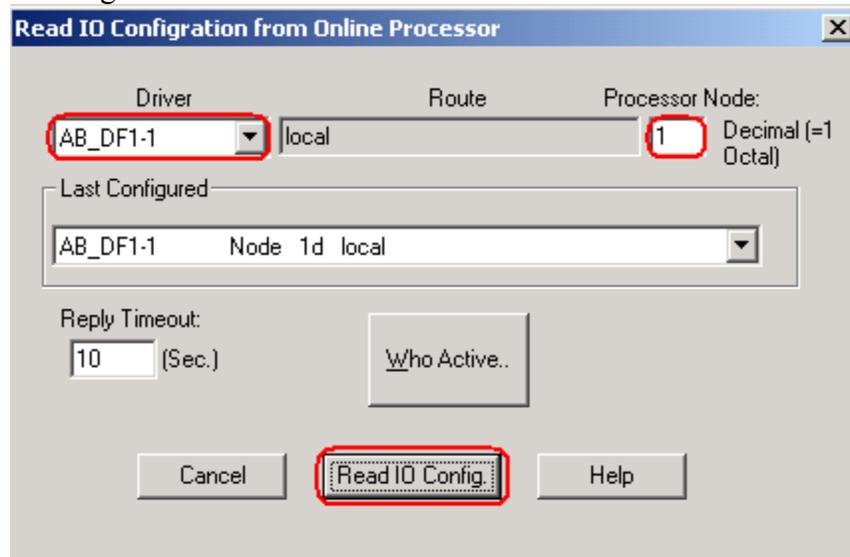


7) Open I/O Configuration in the project tree.

8) Press the Read I/O Config button.



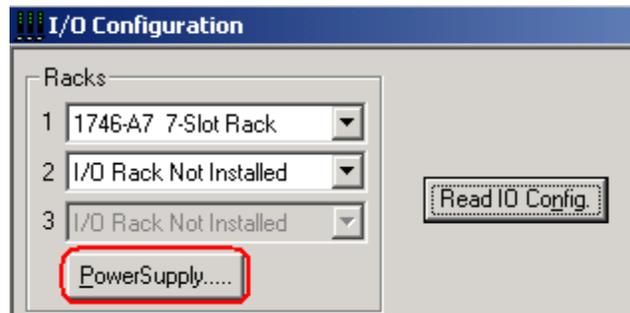
9) Be sure your drive and node number are set up properly, then press the Read I/O Config button again.



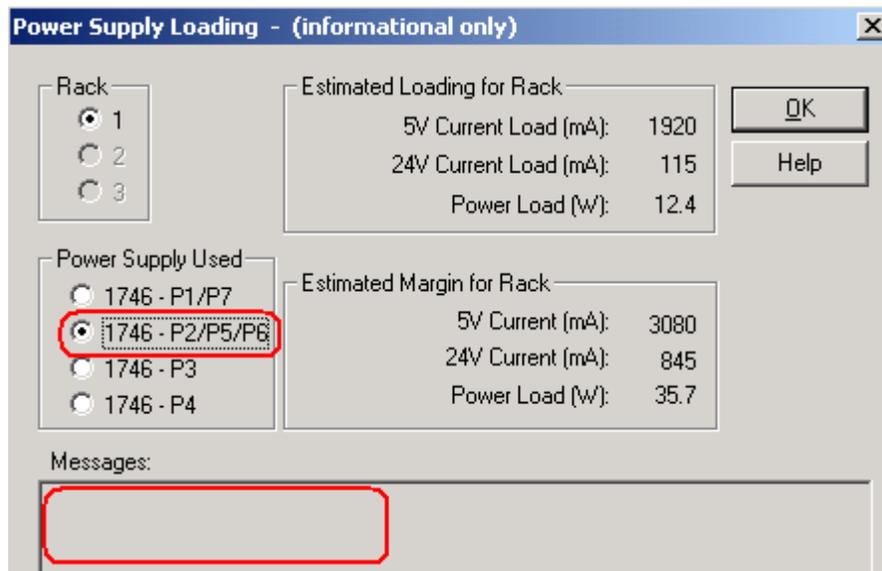
10) The modules in the I/O Configuration list should now match the modules you have in the physical chassis. If you have any specialty modules that need advanced configuration, you can double click the module to configure it at this time.

#	Part #	Description
0	1747-L552B	5/05 CPU - 32K Mem. OS501 Series C
1		
2	1746-IB16	16-Input (SINK) 24 VDC
3	1746-OB16	16-Output (TRANS-SRC) 10/50 VDC
4	1746-NIO4V	Analog 2 Ch In/2 Ch Voltage Out
5		
6	1747-SDN	DeviceNet Scanner Module

11) Click the Power supply button to verify your power supply is not overloaded. You will need to look at the label on the side of your power supply to determine its catalog number.

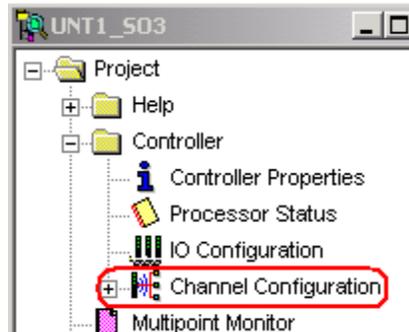


12) Click the power supply you are using, and verify you are not getting a message stating the power supply is overloaded. Click OK, and close out of I/O Configuration.

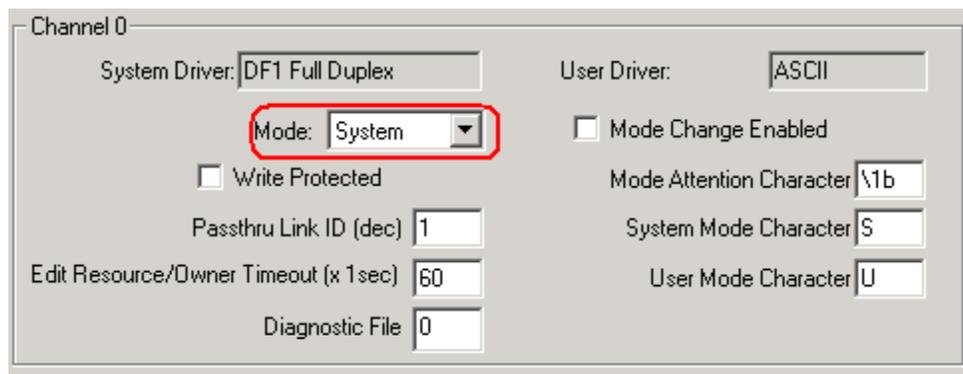


13) If this is a stand alone processor, and no channel configuration is required you may skip to the download section at this time. If this processor is running on a network, you may want to follow the next steps to set up the channel configuration.

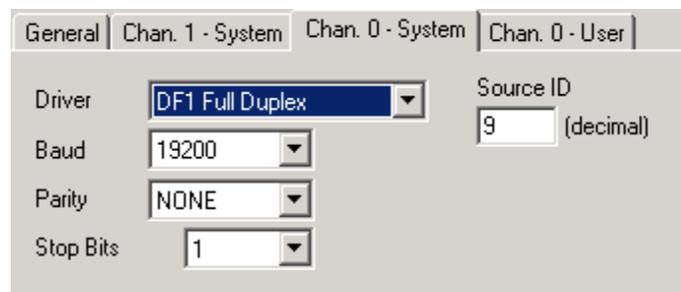
14) Double click channel configuration in the project tree:



15) You will notice on the general tab, that channel 0 is in system mode. If you intend to upload and download to the processor through channel 0, you should leave channel 0 in system mode.



16) Click on the System tab for channel 0 at the top of your channel configuration dialog screen. You will notice that channel 0 is set up for DF1. We will leave this tab at default for this example. If you intend to run this processor on a DH+ network with an AIC+ module, or to a DH485 panelview, you would change the protocol to DH485.



17) Click the channel 1 tab at the top of the channel configuration dialog screen. This will be different for different processors. For example, if you have a 5/04 processor, channel 1 defaults to DH+, and you can set the node address and the baud rate. Since we have a 5/05, we will need to set up the IP address and subnet mask of this module unless we have a BootP server on the network to issue this information to the processor for us. This information can usually be obtained from your network administrator. Do not use the IP address in this example. Ask the instructor what IP address you should use for this class. Be sure to uncheck BootP if you are using a static IP address. When you are finished click OK.

Hardware Address: 00:00:00:00:00:00

DHRID Link ID: 0

IP Address: 192 . 168 . 0 . 25

Subnet Mask: 255 . 255 . 255 . 0

Pass Thru Routing Table File: 0

Gateway Address: 0 . 0 . 0 . 0

User Provided Web Pages:

Starting Data File Number: 0

Number of Pages: 1

Default Domain Name:

Primary Name Server: 0 . 0 . 0 . 0

Secondary Name Server: 0 . 0 . 0 . 0

Protocol Control

Bootp Enable

Msg Connection Timeout (x 1mS): 15000

Msg Reply Timeout (x 1mS): 3000

Inactivity Timeout (x Min): 30

18) You are ready to download. Check the on line tool bar to verify your path and node number are set up correctly. Be sure you are downloading to the correct processor. Downloading to the wrong processor will have disastrous results. Do not download this project to running equipment. This is for classroom purposes only. The processor must go to program mode to download a project.



19) Use the off line pull down tab to download to the processor.



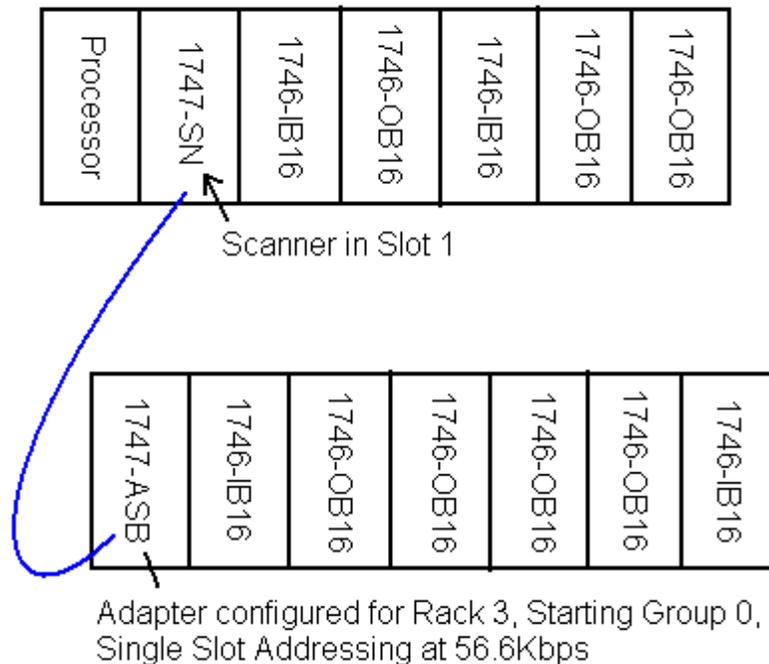
## ***Remote I/O Concepts for the SLC***

There are many reasons you might use Remote I/O. Imagine a conveyor system. Ten thousand feet from the local processor chassis, you have 256 points of I/O. With Remote I/O, you could mount a chassis up to 10,000 feet from the local processor with a Remote I/O adapter, and the processor could control the I/O directly. This would eliminate the need to run 256 cables from the remote location all the way back to the processor.

Remote I/O can also be used to increase the number of I/O points the processor controls that wouldn't normally fit in a standard SLC chassis, or even extended local I/O on the SLC.

A key concept to understand is that the 1747-SN (Scanner Module) converts PLC-5 typed addressing into SLC addressing. (Racks and Groups into Slots and Words)

Consider the Following Image:



## To understand Remote I/O, there are some terms you must learn:

### Bit

This is the smallest unit of information the processor can handle. A bit can only have 2 states: **On or Off**, True or False, 1 or 0, Yes or No (however you wish to express it)

### Word

In the PLC-5 and SLC, a word is a memory location consisting of **16 bits** by definition. The entire input and output data tables are made up of 16 bit words. You can see this clearly when the RADIX is switched to BINARY.

### Group

The Group is a PLC-5 Term. A Group is a memory location consisting of one word of input, and one word of output. (**An input/output pair**).

### Rack

The rack is another PLC-5 term. A Rack is a memory location consisting of 8 Groups. Important: **A RACK ALWAYS consists of 8 Groups**. The Rack is not the same as a chassis on Remote I/O. A chassis can be small enough that it does not take up an entire rack of memory, or may be large enough that it consumes more than one rack of memory. Please do not get the terms Rack and Chassis confused.

### Addressing Mode

Switches located on bank #3 on the SLC adapter, or the backplane of a PLC-5 chassis determine the addressing mode. In 1-slot addressing, an entire group is dedicated to each slot. In 2-slot addressing, a group is dedicated to every 2 slots. In ½-slot addressing (less common), a group is dedicated to every ½ slot (This means you get 2 groups for each slot).

### Baud Rate

The baud rate is the speed at which communication takes place on the network. All devices on the same network **MUST** run at the same baud rate. With a baud rate set set 57.6K, communication is possible for network distances up to 10,000 feet. As the baud rate increases, the allowed network distance shortens. For a baud rate of 115.2K, your network is limited to 5000 feet, and for a baud rate setting of 230.4K, the maximum network distance allowed is 2500 feet.

### Terminators

**BOTH ENDS** of the network must be terminated to hold the network impedance at the proper value. The value of the resistor depends on the baud rate. For baud rates of 57.6K and 115.2K, you will terminate both ends with a 150 Ohm resistor. For 230.4K, use an 82 Ohm resistor. The resistor should be ½ Watt (non-inductive) with a 1% tolerance.

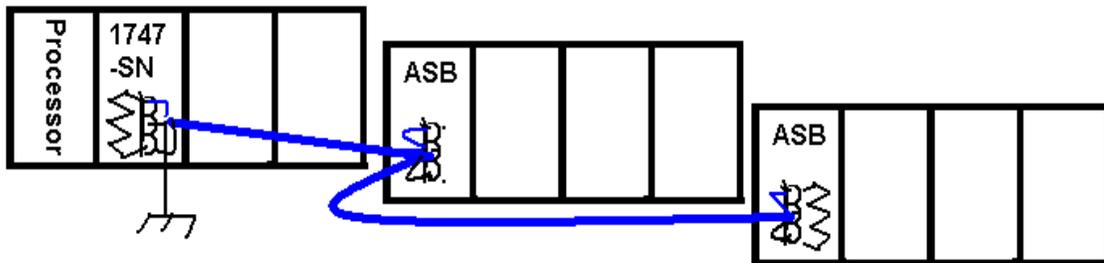
*(Note: For extended node capability (Not as common), you will always use an 82 Ohm resistor)*

## Physical Topology

There are no restrictions on the space between devices on the network as long as the total network length does not exceed the specification for the chart below (*with no extended node capability*)

<i>Baud Rate</i>	<i>Maximum Network Distance</i>	<i>Value of Terminating Resistor</i>
57.6 Kbps	10,000 '	150 Ohm
115.2 Kbps	5,000 '	150 Ohm
230.4 Kbps	2,500 '	82 Ohm

Here is a typical network.



Note: When wiring the phoenix connectors on the scanner and adapters, you should never have more than 2 wires under any 1 screw terminal. Your devices are daisy-chained together or connected in a trunkline/dropline configuration with a special connector box. Star topologies are NOT supported.

Ensure the shield is only grounded at one end. Shielding the ground a both ends could cause a ground loop.

## Gathering Information:

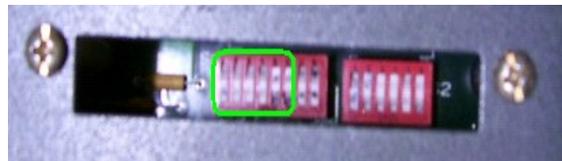
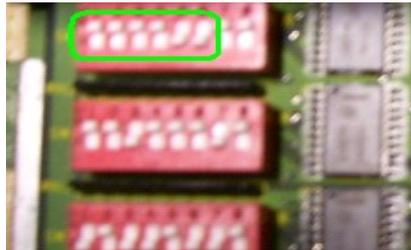
There are 4 things you must know about any device that runs on Remote I/O.

- Rack
- Starting Group
- Baud Rate
- Size (In Racks)

If we had a chassis running on remote I/O, and our system was down, we could remove the adapter from the chassis to gather all the information we need.

Below is a SLC adapter module, A PLC-5 Adapter module, and the PLC-5 backplane. Let's explain how to find or set the **Rack Number** for each type of adapter:

The SLC adapter (top) PLC-5 Adapter (bottom): Switches Switches 1 to 6 on Switch Bank 1:

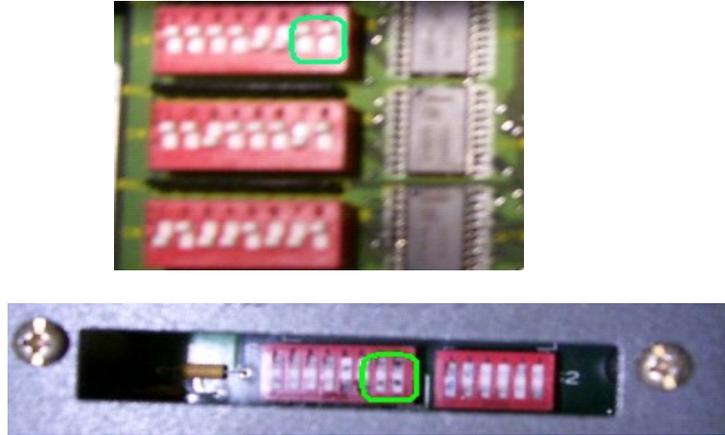


Here are a few settings for the switches

<i>Rack #</i>	<i>Switch 1</i>	<i>Switch 2</i>	<i>Switch 3</i>	<i>Switch 4</i>	<i>Switch 5</i>	<i>Switch 6</i>
0	ON	ON	ON	ON	ON	ON
1	ON	ON	ON	ON	ON	OFF
2	ON	ON	ON	ON	OFF	ON
3	ON	ON	ON	ON	OFF	OFF
4	ON	ON	ON	OFF	ON	ON
5	ON	ON	ON	OFF	ON	OFF
6	ON	ON	ON	OFF	OFF	ON
7	ON	ON	ON	OFF	OFF	OFF

## Starting Module Group (SMG)

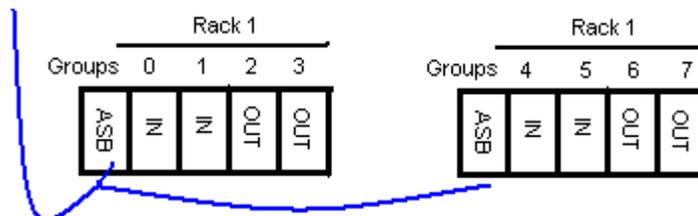
The Starting Module Group (SMG) can be found by looking at switches 7 and 8 on Switch Bank 1 on either the SLC Adapter (top) or the PLC-5 Adapter (bottom)



Use the following table to determine the Starting Module Group:

<i>Starting Module Group</i>	<i>Switch #7</i>	<i>Switch #8</i>
0	ON	ON
2	ON	OFF
4	OFF	ON
6	OFF	OFF

Why have a Starting Module Group? Remember the difference between a Rack and a Chassis. A Rack is a memory location, and a chassis is the physical device that modules are placed in. A Chassis could be so small that it only consumes half a rack of memory (Groups 0 to 3), so then we could put another chassis in the same rack starting at module group 4, then it would consume groups 4-7. Below is an example of this. All 8 groups of Rack 1 are consumed by 2 chassis.

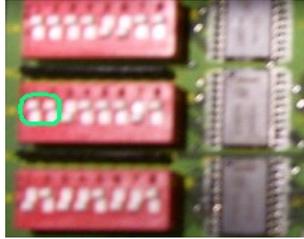


**Baud Rate:**

The baud rate is the speed at which communication takes place on the network. All devices on the same remote I/O network must run the same baud rate.

The baud rate switches for the PLC-5 and the SLC are in the same location on switch bank 2, but the charts are separate for determining the baud rate.

***SLC Adapter (Switch Bank 2 – Switches 1 and 2):***



<b><i>Baud Rate</i></b>	<b><i>Switch 1</i></b>	<b><i>Switch 2</i></b>
57.6 K	ON	ON
115.2K	ON	OFF
230.4K	OFF	OFF

***PLC Adapter (Switch Bank 2 – Switches 1 and 2)***



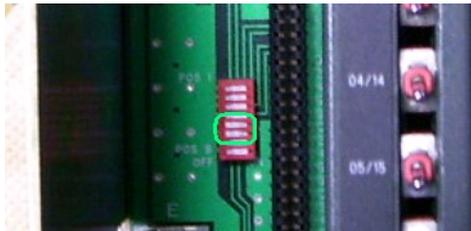
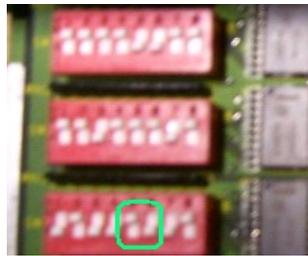
<b><i>Baud Rate</i></b>	<b><i>Switch 1</i></b>	<b><i>Switch 2</i></b>
57.6K	ON	OFF
115.2K	OFF	OFF
230.4K	OFF	ON

## Image Size:

The next piece of information we need to gather is the image size. This can be done by looking at the backplane switch on the PLC-5 Adapter's chassis, or Switch #3 on the SLC Adapter. *(Note: Since there are no backplane switches on the SLC, switch bank 3 was necessary on the SLC adapter to handle the functions of the backplane switch on the PLC-5 chassis)*

Once we determine the addressing mode, we can figure out what size the chassis is. Let's find what addressing mode your chassis is configured for, then we'll figure out the size.

Look at Switch Bank 3 on the SLC Adapter, or the Backplane switches on the PLC-5. (Switches 5 and 6)



<i>Addressing Mode</i>	<i>Switch 5</i>	<i>Switch 6</i>
2- Slot	OFF	OFF
1-Slot	ON	OFF
½ Slot	OFF	ON

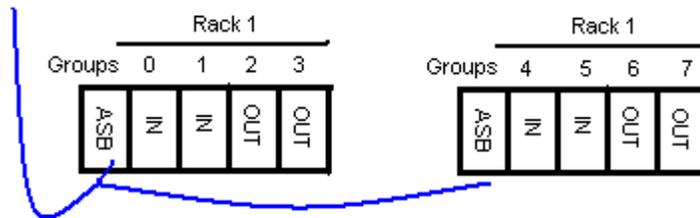
## Deriving the Size:

Remember the definition of Addressing Mode:

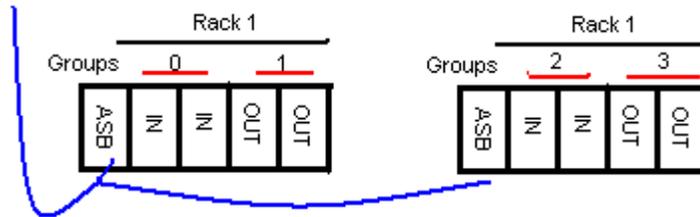
Switches located on bank #3 on the SLC adapter, or the backplane of a PLC-5 chassis determine the addressing mode. In 1-slot addressing, an entire group is dedicated to each slot. In 2-slot addressing, a group is dedicated to every 2 slots. In 1/2-slot addressing (less common), a group is dedicated to every 1/2 slot (This means you get 2 groups for each slot).

Let's look at some examples – You will notice the adapter is not counted as an I/O group:

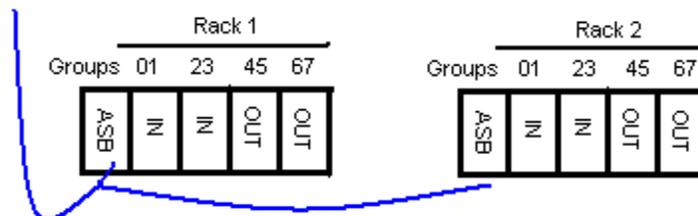
**1-Slot Addressing** – Each group is dedicated to a slot (each chassis below consumes 1/2 Rack)



**2-Slot Addressing** – Each group is dedicated to every 2 slots (each chassis below consumes 1/4 Rack)



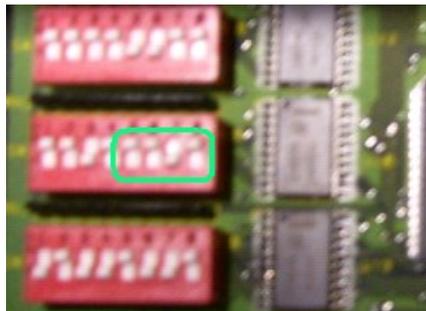
**1/2-Slot Addressing** – Each group is dedicated to every 1/2 slot. (2 groups per slot) Each chassis below consumes an entire rack of memory. This mode is generally used with 32 point modules.



You can use the chart below as a cross reference to derive the size if you know how many slots are in the chassis, and the addressing mode.

<i>PLC-5 Chassis Size</i>	<i>2-Slot Addressing</i>	<i>1-Slot Addressing</i>	<i>½ Slot Addressing</i>
4-Slots	¼ Rack	½ Rack	FULL (1 Rack)
8-Slots	½ Rack	FULL (1 Rack)	2 Racks
12-Slots	¾ Rack	1 ½ Racks	3 Racks
16 Slots	FULL (1 Rack)	2 Racks	4 Racks

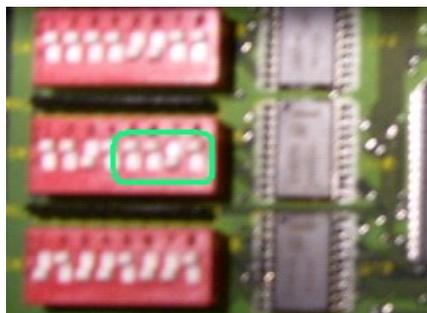
The SLC chassis do not generally come out to even numbers, but are generally set according to the guidelines. For example: If you have an SLC chassis in 1 slot addressing, the adapter would go in the first slot of the chassis. If there were six other slots left in the chassis, we would say the chassis consumes ¾ or a rack. Because the SLC chassis do not always come out even, an additional switches are used on the SLC chassis to manually assign the size. This size setting is located on Switch Bank 2, switches 5-8.



The next page will explain each setting configuration. This step is NOT required for a PLC-5 chassis. The adapter automatically recognizes the size in the PLC-5.

### SLC Size Settings for the 1747-ASB

<i>Image Size (In Groups)</i>	<i>Switch 5</i>	<i>Switch 6</i>	<i>Switch 7</i>	<i>Switch 8</i>
2	ON	ON	ON	ON
4	ON	ON	ON	OFF
6	ON	ON	OFF	ON
8	ON	ON	OFF	OFF
10	ON	OFF	ON	ON
12	ON	OFF	ON	OFF
14	ON	OFF	OFF	ON
16	ON	OFF	OFF	OFF
18	OFF	ON	ON	ON
20	OFF	ON	ON	OFF
22	OFF	ON	OFF	ON
24	OFF	ON	OFF	OFF
26	OFF	OFF	ON	ON
28	OFF	OFF	ON	OFF
30	OFF	OFF	OFF	ON
32	OFF	OFF	OFF	OFF



## Making it work!

The SLC adapter will generally run if you do 2 things: Check or Set the Adapter Parameters, and Tell the scanner to scan the adapter. (Your cabling system must also be in tact)

For our classroom system, write down the parameters you found on the adapter you will be scanning:

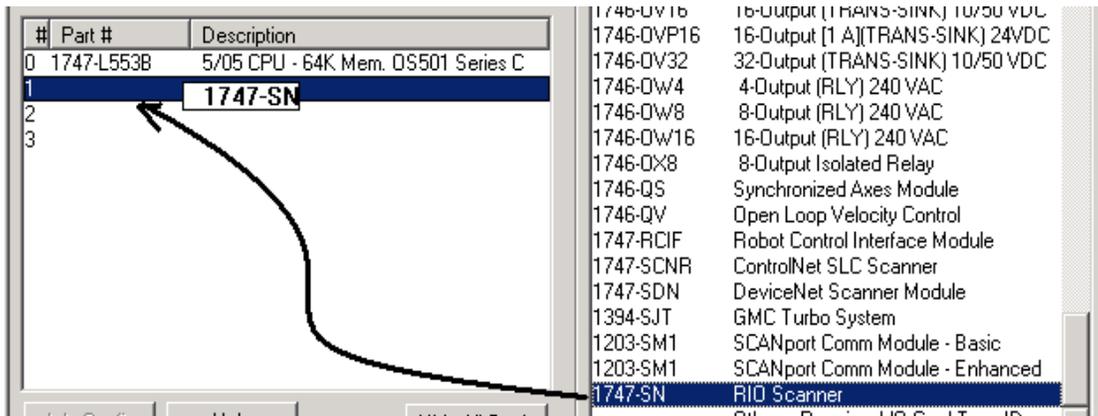
Rack \_\_\_\_\_

SMG \_\_\_\_\_

Baud Rate \_\_\_\_\_

Size \_\_\_\_\_

Now we need to tell the scanner to scan the adapter. We'll do that in the I/O Configuration in RSLogix 500. Be sure the 1747-SN module is in the I/O Configuration. If not, Just drag the scanner module from the list of Available modules, and drop it into the appropriate slot.

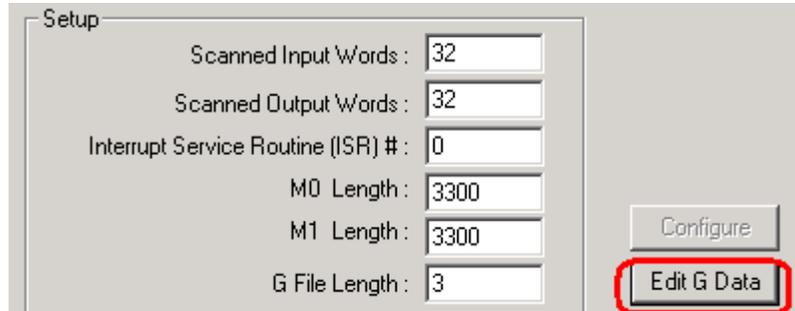


The series of the scanner is printed on the side of the module, so be sure to select the appropriate series.

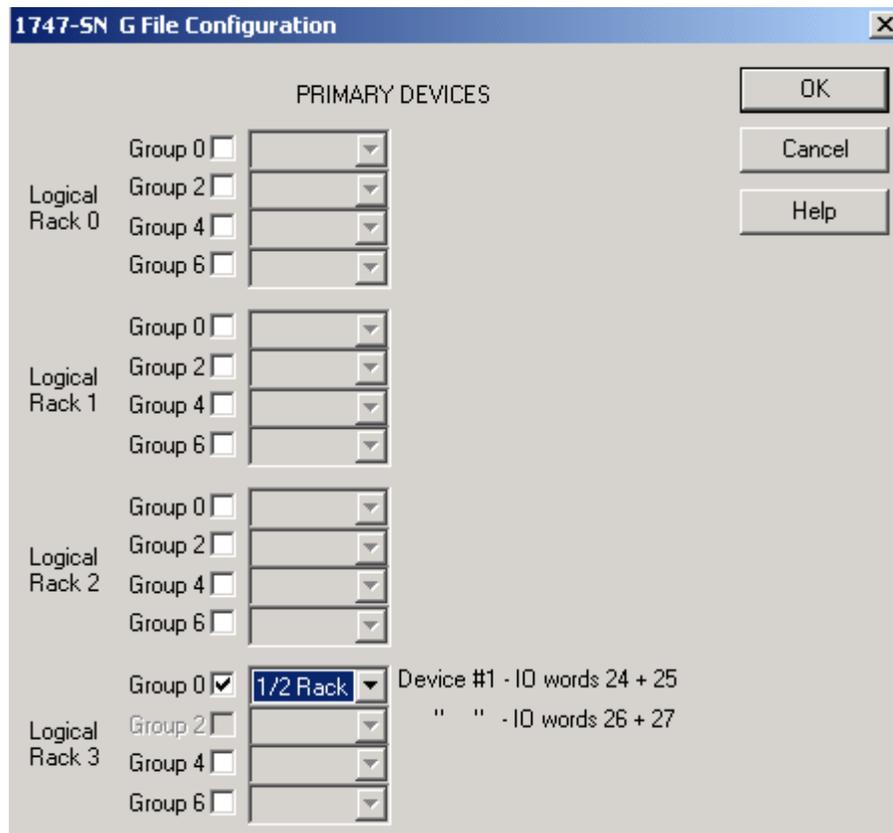


Double click the scanner module in the I/O Configuration, or highlight the scanner, and click “Advanced Configuration”.

Choose “Edit 'G' Data” This is the scanner configuration data.



Next, set up the scanner to scan your adapter(s). In this example, I'm going to scan Rack 3, SMG 0, 1/4 Rack. Be aware that the DIP switches on the Scanner module itself must be set up for the same Baud Rate as the adapter, or vice-versa.



You are ready to download and test your work. Make sure the Comms light is solid green on the scanner, and that the adapter is active.

### Interpreting the Data:

We stated earlier that the SLC does not understand the racks and groups that all remote I/O addressing uses. These Racks and Groups must be “translated” to slots and words.

In this example, we only have one slot to work with for all Remote I/O. That is slot 1. Any device that is wired to the scanner will show up on a word in slot 1.

Let's look at the input data table. (Note: The output table will be very similar for slot 1).

I:1.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner
I:1.31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1747-SN - RIO Scanner

You will notice that by scrolling down, there are 32 words total for slot 1 (0 to 31). The output table is the same.

### Lets do the math:

- Remember that a Group is 1 word of input AND on word of output (an input / output pair).
- In slot 1, we have 32 words of input AND 32 words of output.
- Therefore, we have 32 Groups in slot 1
- We know there are 8 groups in a Rack
- Therefore we have 4 Racks available in slot 1 (32/8)

These 4 racks are going to be numbered 0 to 3 – Let's visually look and see how these 32 words are broken down into racks.

<b>Rack #</b>	<b>Group #</b>	<b>SLC Address (E = Slot # of Scanner T = type (I or O))</b>
0	0	T : E . 0
0	1	T : E . 1
0	2	T : E . 2
0	3	T : E . 3
0	4	T : E . 4
0	5	T : E . 5
0	6	T : E . 6
0	7	T : E . 7
1	0	T : E . 8
1	1	T : E . 9
1	2	T : E . 10
1	3	T : E . 11
1	4	T : E . 12
1	5	T : E . 13
1	6	T : E . 14
1	7	T : E . 15
2	0	T : E . 16
2	1	T : E . 17
2	2	T : E . 18
2	3	T : E . 19
2	4	T : E . 20
2	5	T : E . 21
2	6	T : E . 22
2	7	T : E . 23
3	0	T : E . 24
3	1	T : E . 25
3	2	T : E . 26
3	3	T : E . 27
3	4	T : E . 28
3	5	T : E . 29
3	6	T : E . 30
3	7	T : E . 31



## *Adding Ladder Logic Off line*

Ladder logic is still the primary programming language for Allen Bradley PLC's. Ladder logic was designed to simulate relay logic when PLC's were first introduced into the industry. With the wide use of Object Oriented concepts in today's programming languages, and the limitations of ladder logic for programming, one day ladder logic may be a language of the past.

Ladder 2 is the only main routine in the SLC, and by default Ladder 2 is the only Main Routine in the PLC-5. Other ladders can be added called Subroutines. As with many other programming languages, if you are going to have subroutines, there has to be some means of calling those subroutines so they will be executed by the processor. In the PLC-5 and SLC 500, this is the JSR instruction (jump to subroutine). The maximum routine number is 255 for the SLC, and 999 for the PLC-5.

Usually the JSR instruction is placed in the main routine, but JSR instructions can also be placed in subroutines if the subroutine itself is being called from a JSR. For example... The main routine might jump to ladder 3, and ladder 3 could jump to ladder 4. This is called nesting subroutines. Each processor has limitations on how deep the programmer can nest routines. Consult the help file for your processor.

For this example, Ladder 2 will be the main routine. We will also add other routines: Ladder 3, Pumps; Ladder 4, Timers; Ladder 5, Counters; Ladder 6, Data; Ladder 7; Analog.

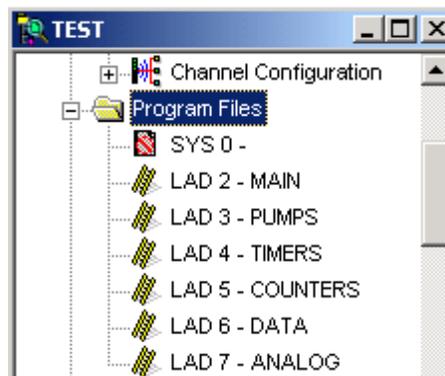
- 1) Rename Ladder 2 to Main. In the project tree, right click Ladder 2, and select Rename. Type MAIN as the routine name. Press enter to accept. Right click on MAIN, and select properties. Type Main Routine as the description. Now, move your mouse over the main routine. You will see the description appears as a tool tip.



2) Next Right click on the program file folder and select 'new' to add other routines to your project. Set up your routines as follows:

Number	Name	Descriptions
3	Pumps	Logic for All Pumps
4	Timers	Project Timers
5	Counters	Project Counters
6	Data	Calculations
7	Analog	Block Transfers for Analog

When finished, your project tree will look as shown below:



3) Now double click on main to ensure you are looking at the main routine in the ladder window to the right.

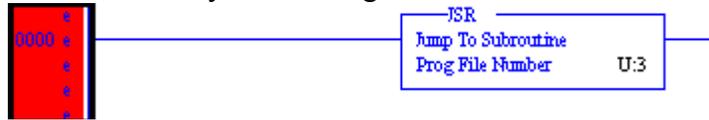
4) One method of adding logic is the ASCII method. You can just type the 3 character instruction followed by certain parameters. We will need 5 JSR statements... One for each subroutine. Do not jump to the main routine from the main routine. A processor fault may result.

5) Highlight Rung 0 so it is red as shown below:



6) Type the following text: jsr 3

Now press the enter or return key. Your rung should now look like this:

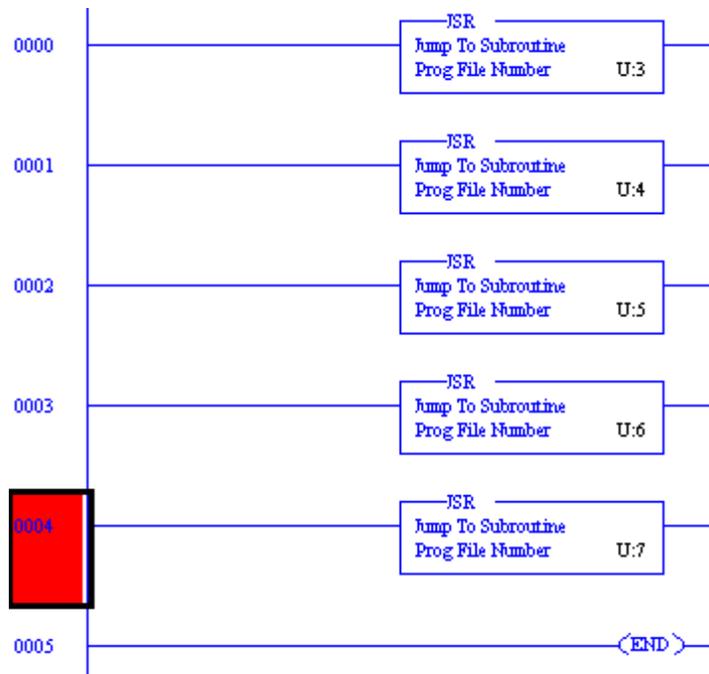


7) Now, click on the end rung again, and type the next jump statement: “jsr 4”, then press enter. Go ahead and all the rest of the jump statements (jsr 5, jsr 6, and jsr 7) Be sure to click on the end rung each time or you will over write your work.

8) Notice the e's in the margin. This indicates that we have edits that have not yet been verified by RSLogix. Verifying edits will make sure the PLC can understand the instructions we are giving it. Verifying **only tests to see if the processor can understand what we are telling it to do. It will not ensure that we are telling it to do the right thing!** Move your mouse over these two icons on the standard tool bar:



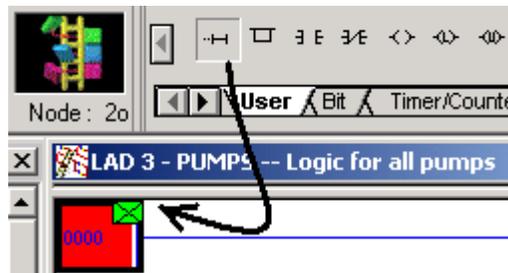
The first icon will verify edits too the file we are currently in (ladder 2). The second icon will check all ladders. Click the first icon, and you will notice the e's no longer appear in the margin. Your ladder should now look as shown below:



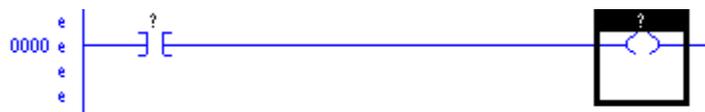
*Note: Advanced users of the JSR instruction may use extra parameters to pass information to and from subroutines. For our examples, we will keep our logic simple.*

9) Next, double click Ladder 3. This is where our pump logic belongs. Look in the title bar of the ladder view to ensure you are entering logic into ladder 3. You will also notice at the bottom of the ladder view a new tab is created each time a new routine is opened. This allows you to quickly switch between routines you are interested in.

10)The first method you learned when adding logic was the ASCII method. For the next few rungs, let's look at the Drag and Drop method. You cannot add any logic to the end rung, so you will need to drag a new rung into your ladder view. The first icon on the instruction tool bar is 'new rung'. Click on the new rung, and hold the left mouse button down. Drag the new rung down into the ladder view, and you will see red boxes indicating where you can drop the new rung. Move close to a red box, and it will turn green. You can then let go of the left mouse button, and the new rung should be in place.



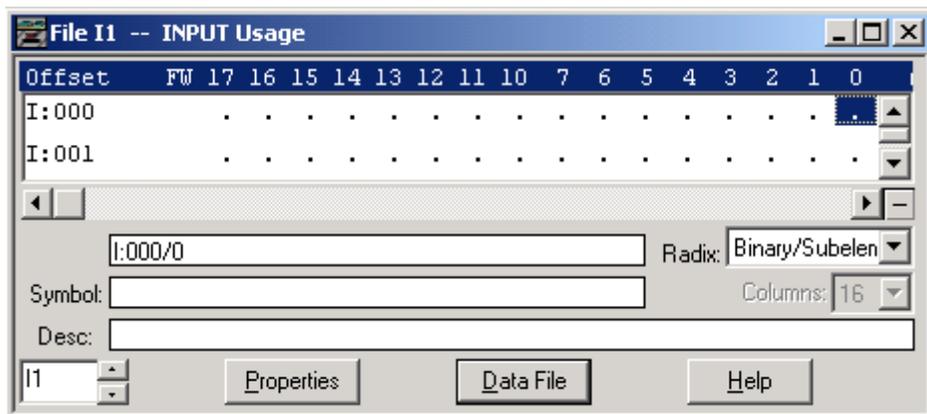
11)Use the same drag and drop method to add an xic and an ote to the new rung. When you are finished your rung should appear as shown below:



*Note: An easier method for you might be the single click method. Be sure rung 0 is highlighted so the rung number is red. Click the new rung icon one time, then click the XIC icon, then the OTE icon. You will get the same result.*

12) Notice the question marks above the instructions. We have to associate the instructions with addresses. There are several methods you can use for this. One method would be to click on the instruction and start typing the address, or open a data table, and drag the addresses from the data table. For this example, we will use a PLC project, and drag the addresses onto the instructions.

13) Open the input data table. Although we can drag directly from the data table, let's click the usage button at the bottom of the data table, so we can see what addresses are already in use.



14) Drag an address from the data table into your logic window. When you see a green square on the XIC instruction, drop the address.

15) In the bottom left corner of the data table, click the down arrow to navigate to data table 0 (output table). Drag an address from the output table onto the OTE instruction.

*Note: The XIC instruction can read most any bit in any data table, and the OTE instruction can write to most any bit in any data table. We are using the input and output tables just for example.*

When finished your logic should look similar to this (with your own addresses):

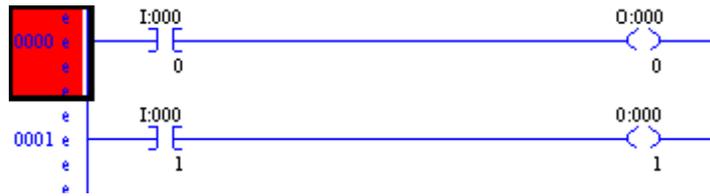


16) Download and test your work.

## Finding Errors

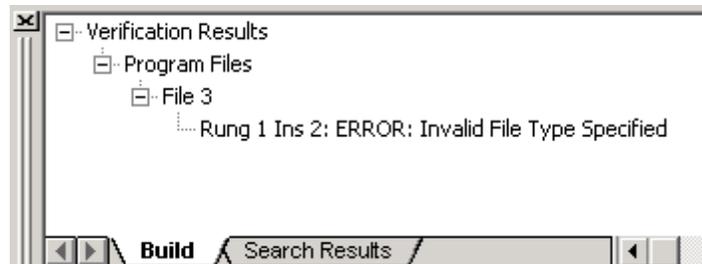
Errors you make are sometimes tricky to find. RSLogix has a results window that will show you where your mistakes are. Mistakes are usually found when verifying a project. If there is an error in your program, a results window will appear at the bottom of the ladder view, informing you where the error is, and a description of the error. By double clicking the error itself in the results window, RSLogix will take you to the exact location of the problem. Sometimes this might be behind the results window, so you may have to close the results window to see it.

Can you spot the error with this logic?



This error is very difficult to find. I will verify the project, and the results window will appear.

I had to click the + on file 3 to see what the error was, but here is the result:

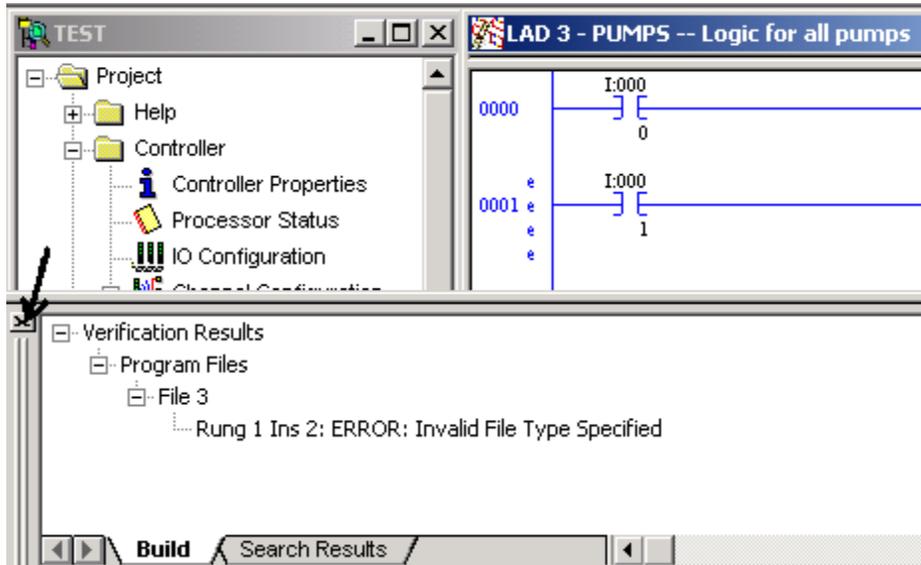


Still can't see the problem, I will double click the error, and RSLogix will highlight where the problem is:



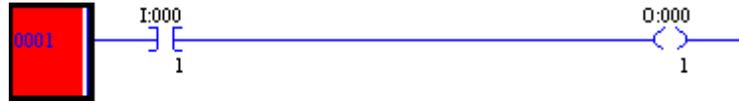
Do you see the problem yet? The first character is the number 0, and should be the letter O.

Note: You should exit the results window by clicking the X in the upper left hand corner of the window. If you resize the results window, you may not see it next time you have an error in your logic.

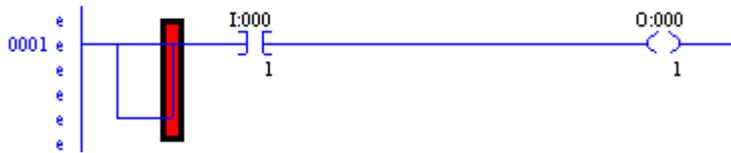


## Branching

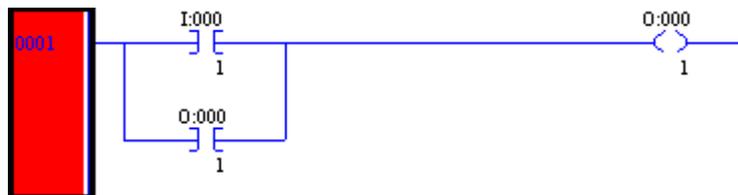
Branching can also be accomplished in ASCII mode using the BST (Branch Start), NXB (Next Branch), and BND (Branch End) commands. For this lesson, we will discuss the drag and drop method. In the example below, Let's seal around the input when the output is energized.



17)The next step is to drag a new branch from the instruction tool bar. You can drop the new branch at any location a red box appears. Be sure to move close to any red box, and when it turns green, your branch can be dropped. For this example, drop off line.

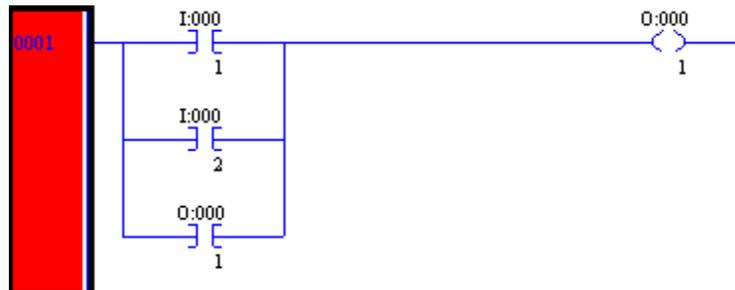


18)Only the right side of the branch can be moved. We have two options: You can move the XIC inside the branch, or you can drag the right branch leg to the other side of the XIC instruction. When finished, the rung will look like this:

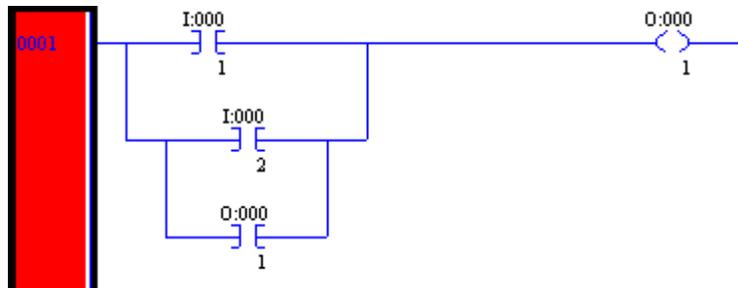


19) If other inputs are required, you can continue to extend the existing branch down. Do not drag new branches down from the instruction tool bar unnecessarily. When possible, extend the existing branch. Nesting branches has limitations, and increases the processor scan time. Look at the examples below:

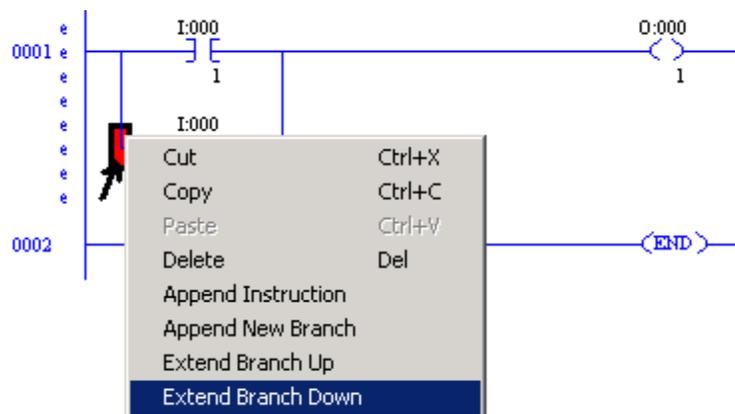
### Correct



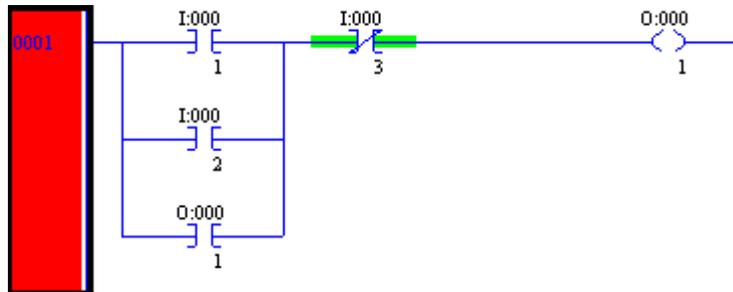
### Incorrect



To extend an existing branch, right click on the bottom left corner of the branch leg, and choose 'extend branch (up or down)'.



20) There is only one thing left to do. Once this branch seals in, there is no way to make the output false again without restarting the processor. Here is what our final rung might look like:



### On Your Own!

Write logic that will perform the following actions:

1. If input 2 (is on), then (energize) light 2.
2. If input 3 AND 4, then (energize) light 3.
3. If input 3 OR 4 OR 5, then light 4.
4. If input (4 OR 5) and (NOT 6) then light 5.

# Basic Instructions

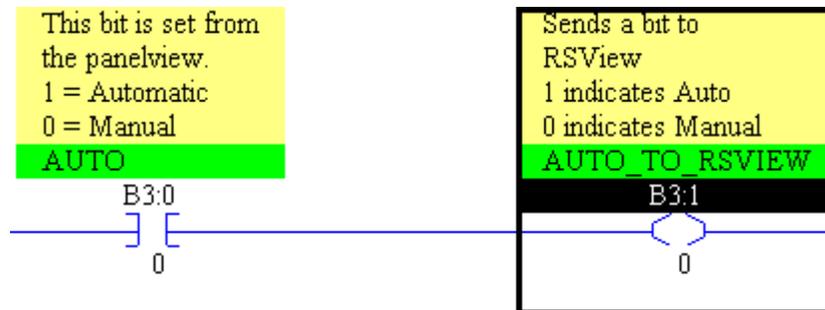
## A Little History

A common programming language used in PLC's is called Ladder Logic. Ladder Logic was developed years ago to help electricians adapt to PLC's. Ladder logic is still widely in use today although this language appears to be weakening. Ladder logic is similar to Assembly Language in many ways which was widely used to program computers years ago. Since then, higher level languages such as PASCAL have come along. In the last few years we have seen a more Object Oriented approach to programming in languages such as Java. The ControlLogix processor seems to be following the Object Oriented approach with it's User Defined datatypes (UDTs), and event driven tasks.

Here are some of the instructions available in Ladder Logic:

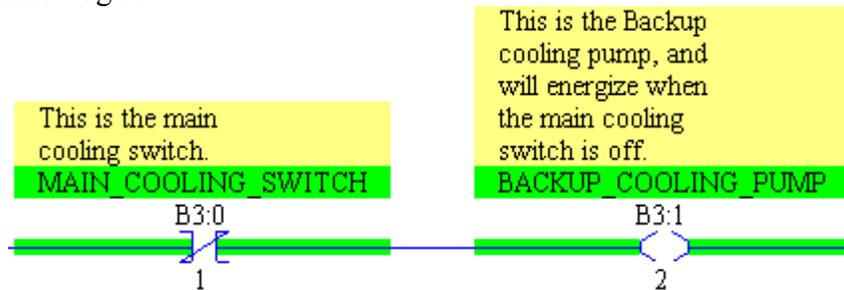
## Examine If Closed (XIC)

You will find that most instructions in the SLC, PLC-5, and ControlLogix consist of three character pneumonics. The XIC looks at any given bit of memory in the processor. If this bit is on, then the XIC will intensify indicating logical continuity through the instruction. Here is what the XIC looks like in logic.



## Examine If Open (XIO)

The XIO is just the opposite of the XIC instruction. The XIO can look at any bit in memory. If the bit is a 0, then the XIO is true. It will intensify indicating logical continuity through the instruction, and the next instruction in the rung will be examined. This is usually referred to as a NOT instruction because the address the instruction points to must NOT be on for the instruction to be true. Here is an example of how the XIO will appear in ladder logic:

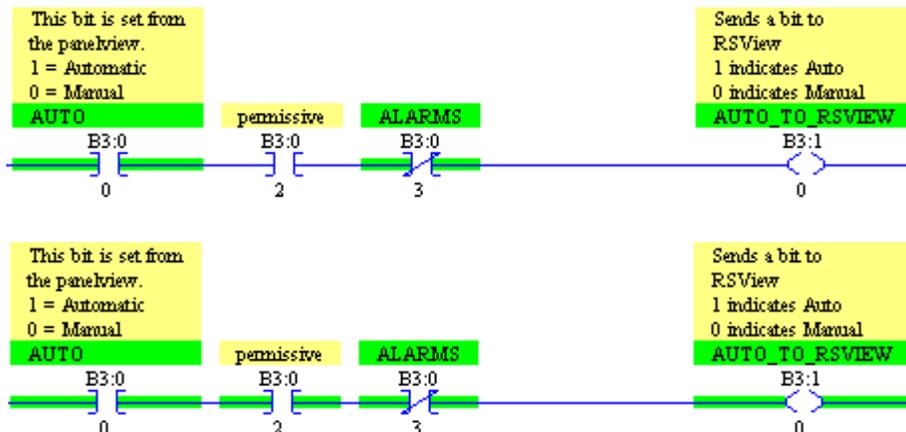


In the above example, you can see that as soon as the Main Cooling Switch is shut off, a bit is set to run the backup cooling pump.

## Output To Energize (OTE)

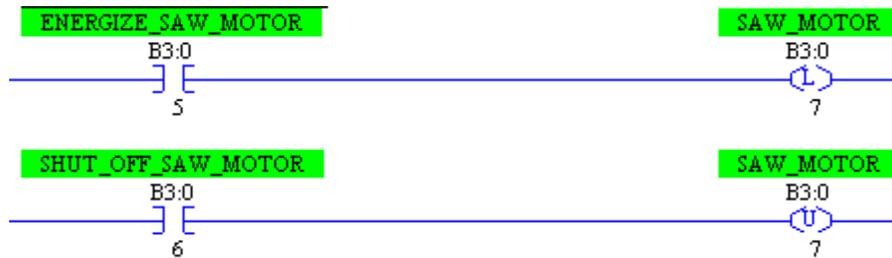
The output to energize simply turns a bit on when it is evaluated as true, and shuts a bit off when the instruction is evaluated as false. Using the same address on an OTE in two different places in the program is considered bad programming practice. The two OTE's can interfere with each other, and makes troubleshooting difficult.

Below you will find two different states of the same rung. The first state shows the rung as false, so a zero is written to B3:1/0. The second state is true, and a 1 will be written to B3:1/0.

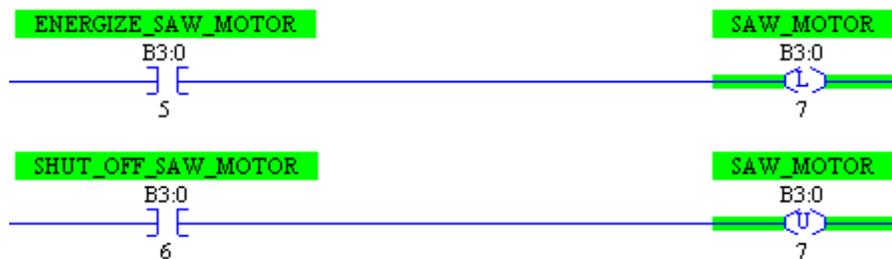


## Output To Latch (OTL) and Output To Unlatch (OTU)

The Output To Latch instruction will write a 1 to it's address when true. When the OTL goes false again, the output address will remain a 1 until another instruction such as the Output to Unlatch shuts it back off. ***This is true even if the processor powers down, and is brought back up!!*** You must use caution when using the Latch/Unlatch when controlling real world devices. Here is what the Latch/Unlatch will look like in logic:



If the output address is off, both the latch and unlatch instructions are not intensified, but once the bit is turned on, you will see both the latch and unlatch intensified even though both inputs are shut off.

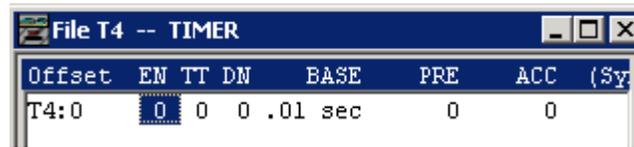


Due to the processor scan cycle, since the unlatch is placed after the latch, if both inputs were to go true, the Unlatch instruction would win, and the output address will be shut off. If the latch was after the unlatch, then the latch would be the last instruction scanned, and therefore the bit would be left in the energized state.

# Timers

Timers are generally used for delaying an event from taking place, or to delay a device from shutting off either on an on transition or an off transition. There are three types of timers: The Timer ON delay (TON), Timer Off delay (TOF), and the Retentative Timer On delay (RTO).

By default, timers are stored in the T4 Data file, however other time files can be created.. A timer consists of the following components: Preset word (PRE), Accumulate word (ACC), Done bit (DN), Timer Timing bit (TT), and Enable bit (EN). For Timers, the Enable bit follows the rung condition.



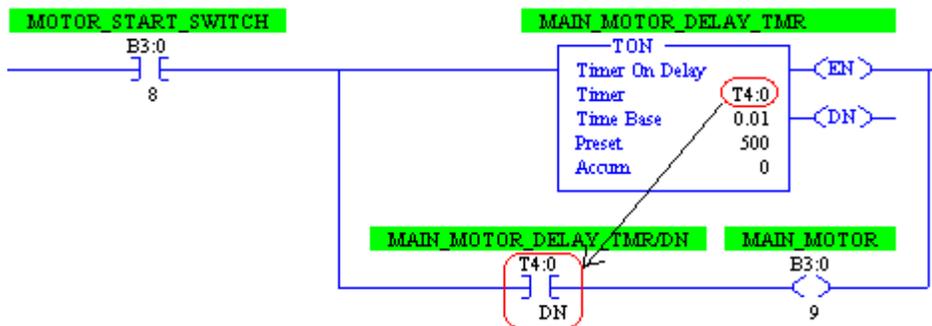
Offset	EN	TT	DN	BASE	PRE	ACC	(Sym)
T4:0	0	0	0	.01 sec	0	0	

The entire timer is addressed by it's element (example: T4:0) Pieces of the timer can be used in logic however such as the DN bit on an XIC (T4:0/DN), or the Accumulated value in a MOV statement (T4:0.ACC)

## Timer On Delay (TON)

The Timer On delay delays an event from taking place. Once the timer becomes true, the enable bit becomes true instantly. The timer will also start timing instantly, so the TT bit becomes high. Since the timer is timing, the accumulated value will increment.

Once the Accumulated value reaches the preset, the done bit (DN) will go high, and the timer will stop timing. The accumulated value remains at (or near) the preset until the rung goes false again. Here is what a typical timer might look like in logic:



When the switch is energized, the timer will begin timing. When the ACC value reaches the PRE value, the DN bit goes high, and the main motor will start. Since the Time Base is .01, therefore 500 (preset) times .01 (timebase) = 5 second delay.

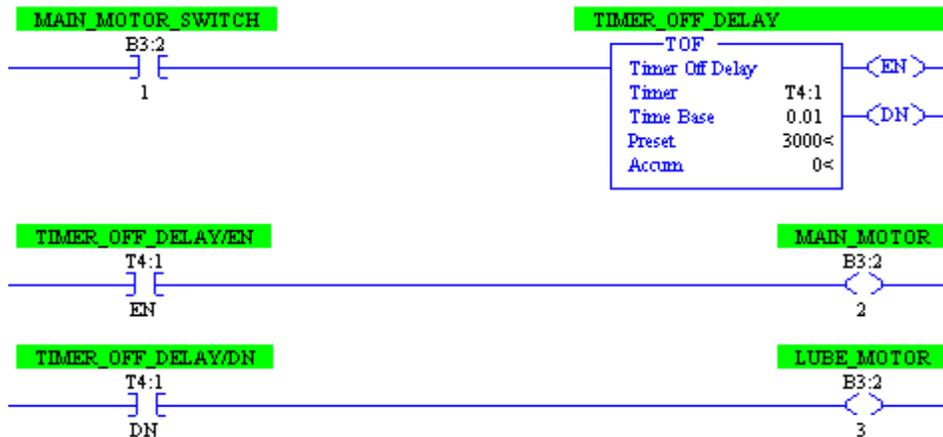
## Timer Off Delay (TOF)

The Off Delay Timer is generally used to delay an event from shutting off. Image a lube system on a large motor. As long as the main motor is turning, the lube pump should be running. When the main motor shuts off, you wouldn't want to shut off the lube pump immediately because the main motor needs time to coast down to zero RPM's. The Main motor could run off the EN bit, and the Lube motor could run off the DN bit.

On the Off delay timer, as soon as the rung goes true, The EN bit goes true as it does for all timers. Since the Off delay timer does not delay the DN bit from shutting off, the DN bit goes high immediately. Remember, the TOF instruction delays the DN bit from shutting off, not turning on. (Plus if we are delaying the DN bit from shutting off, it needs to be high to begin with). While the rung is true, the timer is not timing, and the ACC value is at zero.

When the rung is shut off, the EN bit shuts off immediately. The ACC value will start timing until it reaches PRE then the DN bit will shut off.

Here is what the TOF instruction might look like in logic:

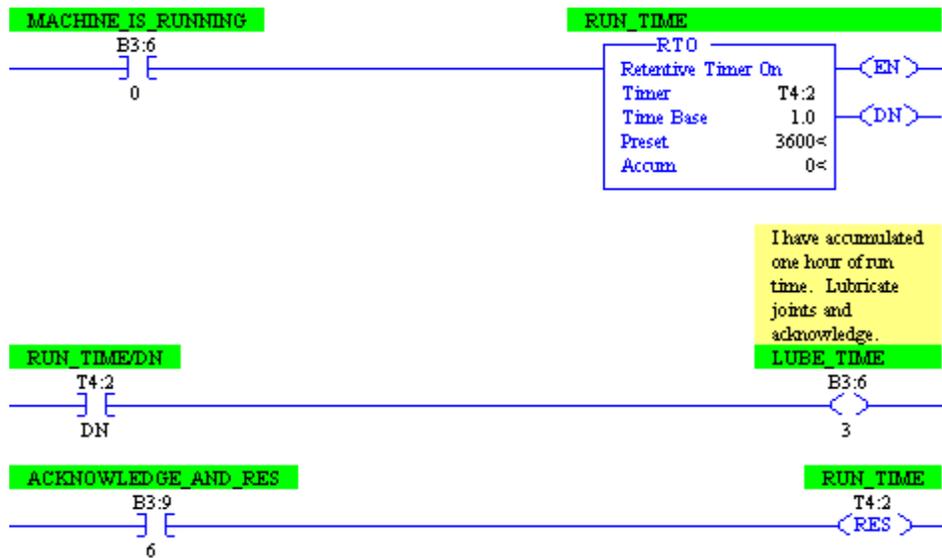


When the motor switch is energized, both the main motor and the lube motor will energize immediately. When the main motor switch is shut off, the main motor shuts off immediately, but since the TOF delays the DN bit from shutting off, the Lube motor will shut off 30 seconds later. Warning: Using the RES instruction on a TOF instruction could cause unpredictable operation.

## Retentive On Delay Timer (RTO)

The RTO instruction works a lot like the TON instruction with one main exception: When the rung goes false on the RTO instruction, it will retain the ACC value. When the rung becomes true again, the ACC value will pick up from where it left off. One good application for the RTO would be an hour meter to indicate total runtime for machinery.

Since the RTO does not reset itself when the rung goes false, the RES instruction must be used to reset a timer. Here is a practical application:



In this example, once the machine accumulates 1 hour of run time, a light might come on indicating that a lubrication needs to be engaged. Once the operator lubricates the machine, he can reset the hour meter.

# Counters

Counters count rung transitions. The CTU runs the accumulated value of the counter up on the false to true rung transition, and the CTD instruction runs the accumulated value down. The CTU and CTD can be used in conjunction with each other.

**Counters consist of the following components:**

ACC	Accumulated Value	PRE	Preset Value
CD	Count Down Bit	CU	Count Up bit
OV	Overflow Bit	UN	Underflow bit

By default, data file C5 stores counters, however, other counter files can be added as well. Below is how the C5 Data file would appear:

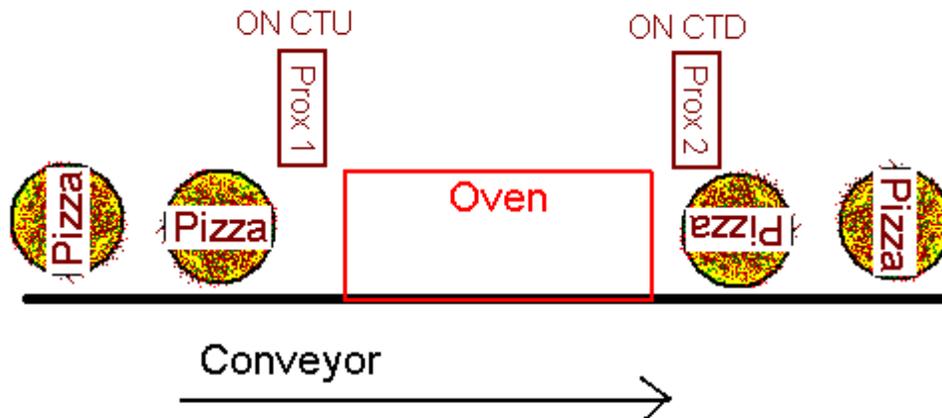


Offset	CU	CD	DN	OV	UN	PRE	ACC	(Symb
C5:0	0	0	0	0	0	0	0	

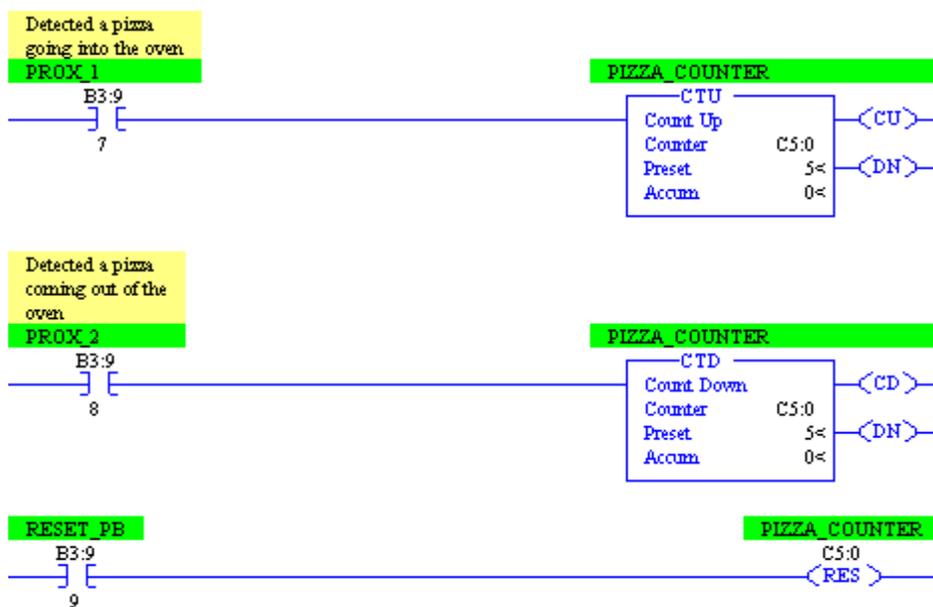
**For the CTU instruction:** The CU bit is high when the CTU instruction is true. The ACC value increments by the value of 1 each time the CU bit goes high. When the ACC reaches the PRE, the DN bit will be set. The CTU will continue to increment the accumulated value until it reaches the maximum possible value for a 16 bit signed integer (32767). If the CU bit goes high one more time, the OV bit will be set, and the ACC value will go to -32768. Each time the CU bit goes high, the ACC value will still continue to increment (become less negative).

**For the CTD instruction:** The CD bit is high when the CTD instruction is true. The ACC value decrements by the value of 1 each time the CD bit goes high. Any time the ACC is above or equal to the PRE, the DN bit will remain set. The DN bit is reset if the ACC falls below the PRE at any time. The CTD will continue to decrement the accumulated value until it reaches the minimum possible value for a 16 bit signed integer (-32768). If the CD bit goes high one more time, the UN bit will be set, and the ACC value will go to 32767. Each time the CD bit goes high, the ACC value will still continue to decrement (become less positive).

Here is a practical example of a CTU/CTD implementation:



Each time a pizza goes into the oven, the ACC value is incremented by one. Each time a pizza comes out of the oven, the ACC value is decremented by one. Therefore, the ACC value represents how many pizzas are in the oven at any given time. The DN bit could be used to shut the conveyor down if pizzas are going into the oven and not coming out!

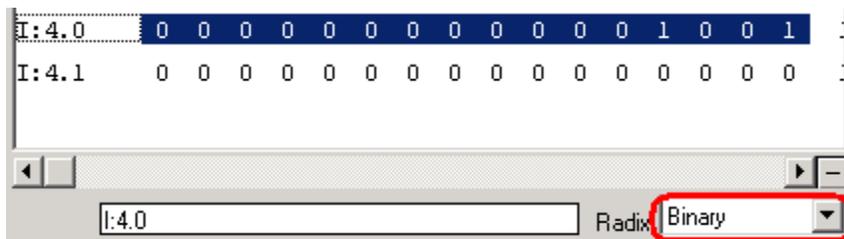


## Working With Analog

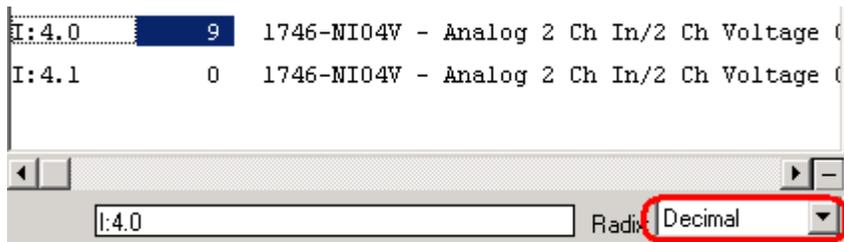
Most of the instructions we've worked with so far have been used for discrete I/O which is either on or off. Discrete I/O required only one bit of memory. Analog values are derived from a variable range such as 4 to 20 milliamps, or 0 to 10 volts. Such a range cannot be expressed with a single bit, so they are addressed to the WORD level (16 bits). When you change your RADIX to 'Decimal', all 16 bits of a given word are converted to a decimal value. For example, we have sixteen bits in the following word: 0000 0000 0000 1001. This binary value would be converted to the number 9 when the radix is switched to decimal.

Look at the following example:

### I:4.0 is shown in Binary

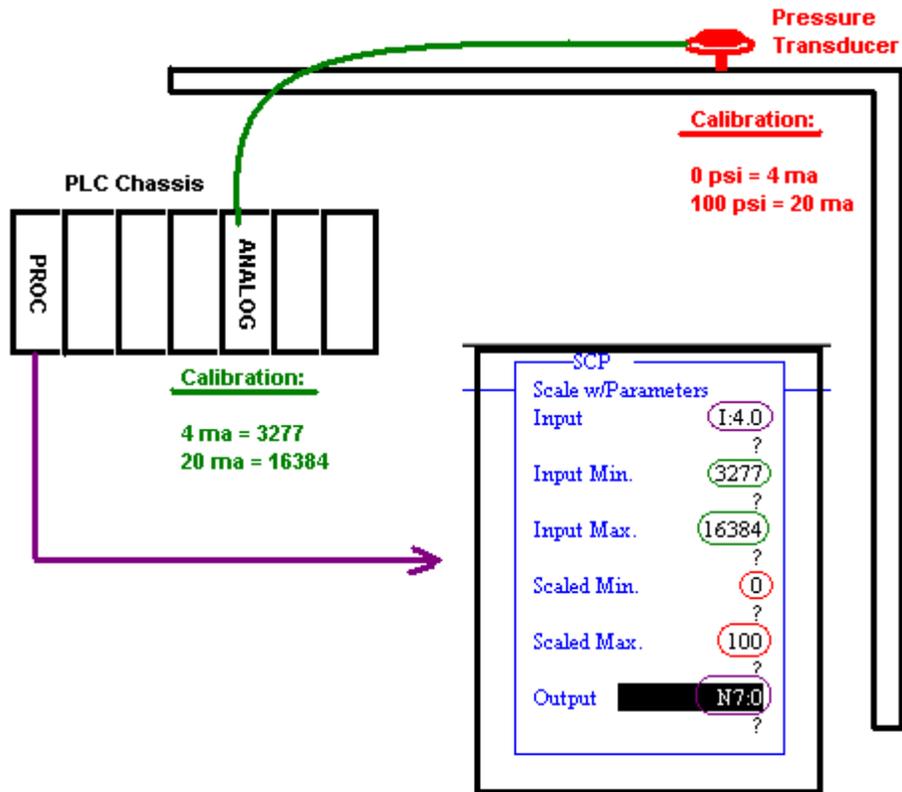


### I:4.0 is shown in Decimal



Analog values are generally displayed in Decimal because we want to see what value the 16 bits represent. Spin your potentiometer, and verify the value from your pot is getting into the input image table.

Let's look at some examples of how analog signals can be used.



Three conversions are taking place in this image.

- 1) The first conversion is at the transducer. The transducer converts 0 to 100 psi of pressure into a 4 to 20 milliamp signal.

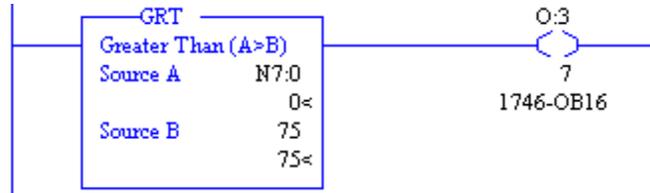
- 2) The second conversion takes place at the Analog Module in the PLC chassis. The analog module converts the 4 to 20 ma signal into a raw value that the processor can understand. To find out what raw value the module will send to the processor, we must go to the manual for our analog module. In this case, when the analog module receives a 4ma signal, it will send the value of 3277 to the processor. When the analog module receives a 20ma signal, it will send a value of 16384 to the processor. Some modules allow these values to be changed (scaled) on board the module. Other modules (such as this one) require the scaling to be performed manually in logic.
- 3) The third conversion is the math in the logic that takes the raw values from the analog module, and scales these values back into engineering units. In this example, we see the SCP instruction being used. We get our raw data from I:4.0. The min and max values we would expect to see at this location are 3277 to 16384. The engineering units these values need to be scaled back to are 0 to 100. We can store the result in N7:0 where it can be used in our ladder logic, or sent to an HMI for an operator to view.

## Compare instructions:

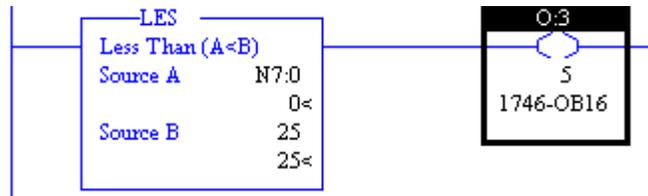
Compare instructions allow us to look at a value in the data table to detect whether that value has gone above, or dropped below a certain value. Compare instructions are non-destructive which means they only look at a value, and do not change the value directly.

Here are some examples:

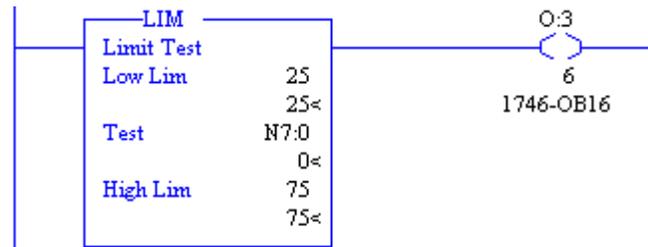
- 1) We need O:3/7 to energize if the pressure exceeds 75 psi.



- 2) We need O:3/5 to energize if the pressure drops below 25 psi.



- 3) We need O:3/6 to energize if the pressure is between (or equal to) 25 and 75 psi.



## **PID Demonstration—SLC 500**

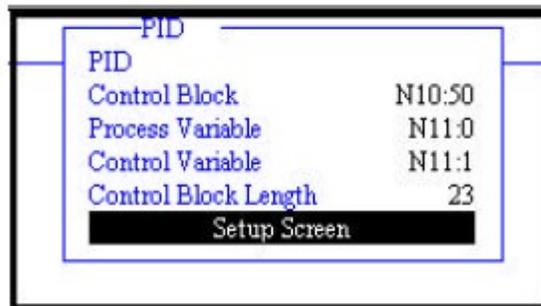
- 1) In Ladder 9, you will find a PID instruction. This instruction is not currently controlling any process. We are going to use this instruction simply for the purpose of seeing how the proportional controller reacts to error. Click “Setup Screen” within the PID instruction, and you will see various parameters to control the PID instruction. Set up the parameters as follows:

The screenshot shows the 'PID Setup' dialog box with the following parameters:

Section	Parameter	Value
Tuning Parameters	Controller Gain Kc	1.0
	Reset Ti	0.0
	Rate Td	0.00
	Loop Update	1.00
	Control Mode	E=SP-PV
	PID Control	AUTO
	Time Mode	TIMED
	Limit Output CV	NO
	Deadband	0
	Feed Forward Bias	0
Inputs	Setpoint SP	100
	Setpoint MAX(Smax)	100
	Setpoint MIN(Smin)	0
	Process Variable PV	100
Output	Control Output CV (%)	0
	Output Max CV (%)	100
	Output Min CV (%)	0
	Scaled Error SE	0
Flags	TM	1
	AM	0
	CM	0
	OL	0
	RG	0
	SC	0
	TF	0
	DA	0
	DB	0
	UL	0
LL	0	
SP	0	
PV	1	
DN	0	
EN	1	

- 2) You will notice that the Controller Gain (Kc) is going to be our proportional gain. Be sure this gain is set for the value of 1 for our first experiment. Ti and Td are your integral and derivative timing values. We will work with these later. The loop update time is how often the PID loop calculates what our output should be. The control mode is reverse acting because an increase in output decreases the error. The error is not an absolute value. It will have a negative value when the Process Variable (PV) is above the Set point (SP). Our control mode is “AUTO” which means the PID equation is writing to the Control Variable (CV). There are two Time Modes: TIMED, or STI. For our purposes, we are going to use TIMED. We are not going to limit the CV, and we will have no dead band. Your SP will start at 0 and its range will be from 0 to 100. Your process variable will be coming in to the processor from the real world process (remember there is none right now). Since we are in Automatic Mode, the CV% will be calculated by the controller. The maximum CV% will be 100 and the Minimum will be 0. The SE is simply going to be the percent error which in this case is also the raw error.

- 3) Next we are going to change the SP to the value of 10. When you changed the Set point to the value of 10, how much output did the controller provide? What is the error?
- 4) If we change the SP to the value of 30, how much output would you expect from the controller? If the PV is still at 0, what is the error going to be?
- 5) Try changing the SP to 30 and see if you get the results you would expect.
- 6) Now lets imagine for a moment that the output we are providing is starting to cause things to happen in the real world.... Such as a flow control valve that is starting to open. We are continuously reading the flow rate from this system. The sensor that is measuring the flow rate is calibrated in such a way that if we get a value of 16383 (20mA), then our flow rate is 100%. The value of 0 returned from the sensor will reflect a flow rate of 0. We can find the conversion from amperage to raw data in the manual for the analog module. Look at your PID equation. What memory location represents the value being returned from the sensor?



- 7) Remember that the value we are sending to the flow valve is called the Control Variable. The value being returned from a sensor which is measuring actual flow is called the Process Variable. The process variable is going to show up in memory location N11:0. Remember how our sensor is calibrated. If we see the value of 16383 at N11:0, then we know we have 100% flow. We are going to simulate feedback from a flow meter by entering the value 3120 into N11:0. After entering the value of 3120 at N11:0, go back to the Setup Screen for your PID instruction.

Data File N11 (dec) -- NOPROCESS			
Offset	0	1	2
N11:0	3120	1794	0
N11:10	0	0	0

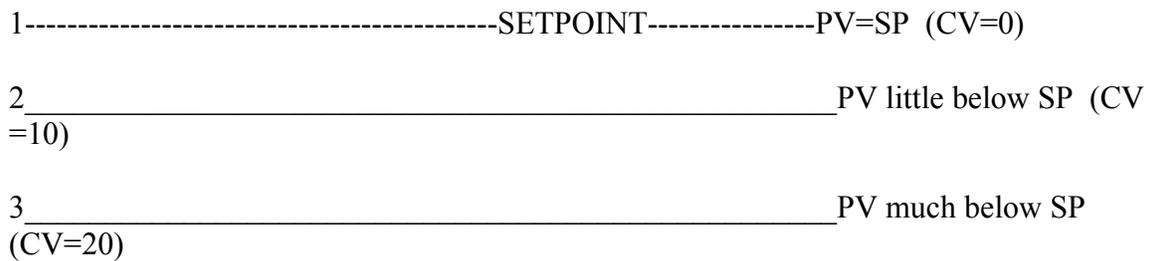
- 8) With your Set point still at 30:
  1. What is the PV reading?
  2. What is the CV reading?
  3. What is the ERROR?
  4. Note that 3120 is 19% of 16383.
- 9) Increase N11:0 to the value of 3800 (23%). What happens to the output as the Process Variable begins to reach the set point?
- 10) Now lets see how a change in Kc gain causes the controller to react. Reset N11:0 back to 0, go to the setup screen for the PID loop in Ladder 9 and change the Kc Gain to the value of 2. If you were to change the set point to the value of 10, how much output would you expect from the controller?
- 11) Change the SP to 10. Did you get the value you expected to see?
- 12) What output would you expect at the CV if you change the SP to 30?
- 13) Change the SP to 30. Did you get the output you expected?
- 14) Change N11:0 to 3800 (23%). What output would you expect to see with a SP of 30?
- 15) Go back to the Setup Screen for the PID instruction in Ladder 9. Was your answer correct?
- 16) Now you understand how the proportional controller alone works. The controller provides an output that is proportional to the error. Think about how the proportional controller alone works. Under normal circumstances, can the proportional controller alone bring the process variable up to the set point?
- 17) Why or Why Not?

18) Think about a heating process:

When you start the heating process, the controller will provide a very large output. As you approach the set point, the output will decrease, until the output is zero at the set point. With an output of zero, you are not putting any heat into the system. Due to ambient and load losses, the system will begin to cool. Eventually, at some point below the set point, the amount of heat the controller is adding to the system will be equal to the amount of heat being removed from the system. At this point your system has reached steady state. The error between the steady state temperature and the set point is called the OFFSET.

If the load is increased, then more heat is being drawn out of the system. The controller is no longer providing sufficient output to maintain the previous offset. A greater error must be present in order to increase the amount of heat the controller is adding to the system.

Now verify your answer to question 10. We will demonstrate this effect in the next example.



Position 1) If the PV was at the SP, then the controller is not adding any power to compensate for ambient and load losses. Temperature cannot be sustained.

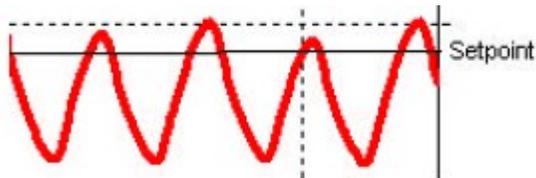
Position 2) If our PV was at this point, we have an error of 10, and therefore the processor is calling for 10% power. Here we can reach steady state if the losses and loads combined are not taking out more heat energy than we are putting into the system. If the load increased, we could not maintain this temperature since our output can only be 10% at this position.

Position 3) If we was at position 2 and our load began to increase, we are taking more heat out of the system than what we are putting in. The temperature begins to decrease. This causes an increase in the error, and the controller will increase power. The system will reach steady state at a lower temperature. This will be the temperature at which the controller is providing enough output to make up for the increased load.

- 19) If the  $K_c$  is increased, then the offset is going to be decreased because we can have the same output with a smaller error. However, if  $K_c$  is increased too high, the controller will become unstable and start to oscillate. Allen Bradley states in the Instruction Set Reference Manual that a good rule of thumb used to set  $K_c$  is half the value which causes oscillations. This will vary depending on the process.
- 20) In this example  $K_c$  is set okay. The PV overshoot the SP, but did eventually reach steady state. Since this is an example of proportional control only. Steady state is actually being reached at a lower temperature than the set point. The reason is because the temperature at which we reached steady state provided the exact error we need for the proportional controller to generate enough output to make up for losses and load. If  $K_c$  is increased, then the error will be decreased, and the steady state temperature will be closer to SP.



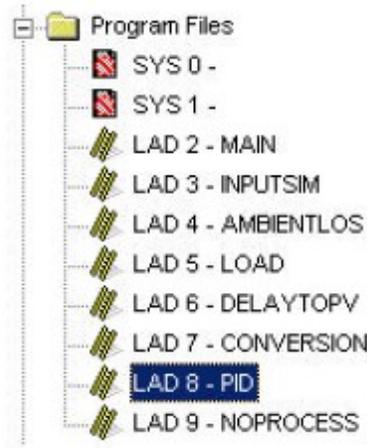
- 21) In this example, we increased  $K_c$  too much. This caused the controller to become unstable and start to oscillate. The minimum value for  $K_c$  to cause this type of oscillation is important for calibration. In many cases, once you find the value of  $K_c$  that causes the controller to oscillate, set  $K_c$  to half this value. The amount of time required for one complete oscillation is called the “natural period” of a sine wave. This natural period will be useful when setting the integral and derivative later on in this lesson.



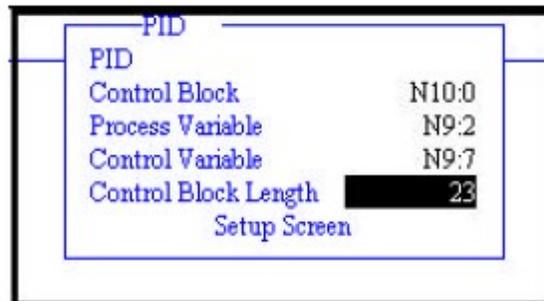
- 22) In the next lesson, we are going to simulate a real world process. During this lesson you are going to determine what is the maximum value we can use for  $K_c$  without the controller becoming unstable.

## Proportional Gain Worksheet

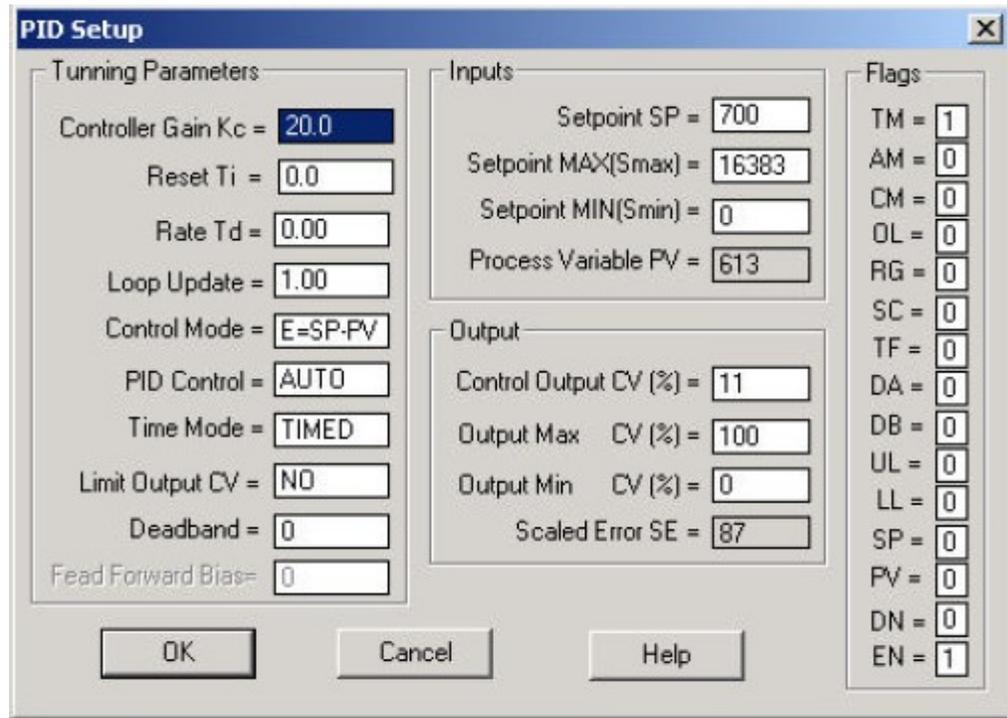
- 23) To start this exercise, double-click ladder 8 in the project tree of RSLogix 500. On the right hand side of your screen, you will see the PID controller for this lesson.



- 24) Notice the PID instruction at the top of your ladder diagram:



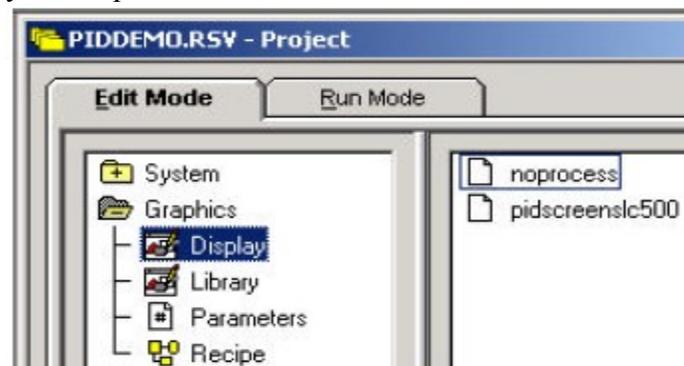
25) Double-click “Setup Screen” within your PID instruction



26) Set the Controller Gain (Kc) to 20.0

27) Next, follow your instructor to open a project in RSView. RSView will be the Graphical User Interface (GUI) for a heating process. The program in your SLC500 will simulate the process. You can find RSView by clicking Start| Programs|Rockwell Software|RSView32|RSView 32 Works. Once RSView is open, Click File, and then open the project called piddemo.rsv

28) Next, Double click on the + next to Graphics. The single click display. A list of the screens available for this project show up on the right hand side. Double click the display called pidscreenslc500.



29) Click the Start button on your tool bar.



30) Now we are set up for the next exercise. In this next exercise you are going to adjust the proportional gain ( $K_c$ ) to find out what value causes the loop to become unstable. Use the following chart: For each row, go to RSLogix and enter the  $K_c$  value into the PID controller in Ladder 8. Come back to RSView, With your load at 0, change the load on the system to 100%. Watch the response. Circle whether the controller is still stable after the transition, or whether it has gone unstable. Then change your load from 100% to 0% and watch the graph in RSView. Record the stability, then move to the next row.

Proportional Gain $K_c$	Load transition 0 to 100%	Load transition 100 to 0%
35.0	Stable   Unstable	Stable   Unstable
40.0	Stable   Unstable	Stable   Unstable
45.0	Stable   Unstable	Stable   Unstable
50.0	Stable   Unstable	Stable   Unstable

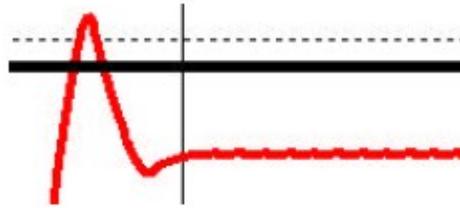
31) What value of  $K_c$  started causing instability during certain load transitions?

\_\_\_\_\_

32) What is the natural period of the oscillations (in minutes)?

\_\_\_\_\_

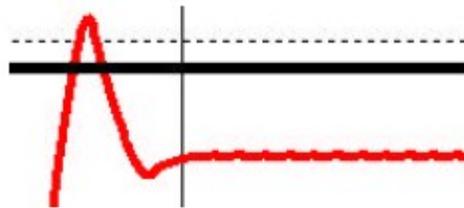
33) Your Set point is 700 degrees. Record the steady state temperature under 0 load with your new  $K_c$ . \_\_\_\_\_ You will see that the steady state temperature is significantly below the set point. That is because of the error required to generate enough output to make up for ambient losses. Put your load at 100% and record the steady state temperature again. \_\_\_\_\_ You will see that it is even lower. Because we are taking more heat out of the system, we must have greater error to generate more output.



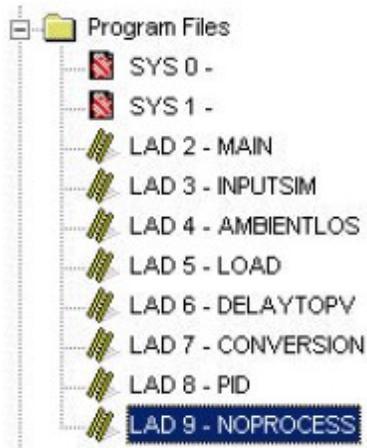
34) We will compensate for this offset in the next lesson

## Integral Worksheet

- 35) You noticed in the last lesson that by using the proportional control alone, we cannot achieve the desired set point, and sometimes cannot even get the PV close to the SP without the loop becoming unstable. We need another component of the controller that can increase the output to push the PV up to the SP. The integral controller controls proportional minutes per repeat. Therefore, the longer the PV stays below the SP, the more output is generated. The closer the PV gets to the SP, the slower the output increases. Remember proportional control generates an output simply based on the amount of error. The integral output takes this one step further. If we still have an error, then we are going to continuously keep adding output ever so slightly until we reach the set point.



- 36) Ladder 9 has a PID instruction set up that will allow us to see the effect of the controller alone when it is not attached to a real world process. Lets bring up RSLogix and double click ladder 9 in the project tree.



37) Before we start, let's set the Process Variable to 0 in the N11 Data File. If this were a real process, the feedback from the process would show up in N11:0. We will do this manually for this example.

Offset	0	1	2
N11:0	0	16383	0
N11:10	0	0	0
N11:20	0		

38) Click 'Setup Screen' on the PID instruction. Configure your PID instruction as follows:

**PID Setup**

**Tuning Parameters**

- Controller Gain Kc = 1.0
- Reset Ti = 0.0
- Rate Td = 0.00
- Loop Update = 1.00
- Control Mode = E=SP-PV
- PID Control = AUTO
- Time Mode = TIMED
- Limit Output CV = NO
- Deadband = 0
- Feed Forward Bias = 0

**Inputs**

- Setpoint SP = 10
- Setpoint MAX(Smax) = 100
- Setpoint MIN(Smin) = 0
- Process Variable PV = 0

**Output**

- Control Output CV (%) = 10
- Output Max CV (%) = 100
- Output Min CV (%) = 0
- Scaled Error SE = 10

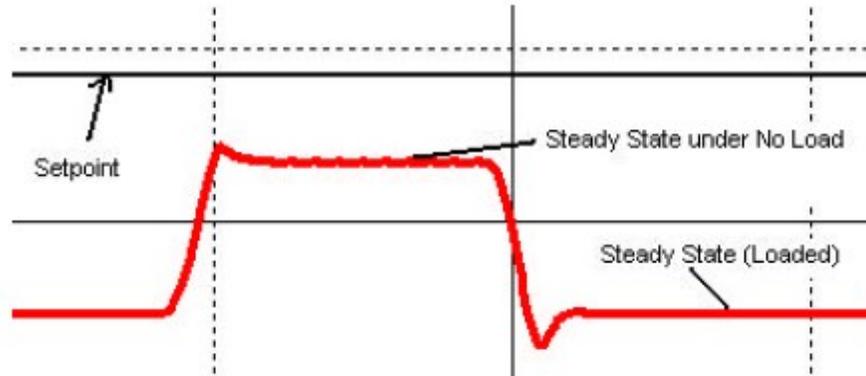
39) Recall that from a previous lesson, using the proportional control only, if our error is 10%, and the Controller gain is 1.0, then the output is going to be 10%. Here we have a condition where our PV is below the set point and is not increasing (just like in Ladder 8 on our heating process).

- 40) Change your Reset Ti to the value of 1.0. If you make a mistake in entering this value (such as entering .1), enter 0 to clear the integral influence on the output, then enter 1.0 again.

The image shows a 'PID Setup' window with two main sections: 'Tuning Parameters' and 'Inputs'. The 'Tuning Parameters' section includes fields for Controller Gain Kc (1.0), Reset Ti (1.0), Rate Td (0.00), Loop Update (1.00), Control Mode (E=SP-PV), PID Control (AUTO), Time Mode (TIMED), Limit Output CV (NO), Deadband (0), and Feed Forward Bias (0). The 'Inputs' section includes Setpoint SP (10), Setpoint MAX(Smax) (100), Setpoint MIN(Smin) (0), and Process Variable PV (0). Below the 'Inputs' section is an 'Output' section with Control Output CV (%) (20), Output Max CV (%) (100), Output Min CV (%) (0), and Scaled Error SE (10).

- 41) Observe what is happening to the CV%. Since we entered Ti as minutes per repeat, after 1 minute, the proportional gain will be repeated, and you will have an output value of 20. After 2 minutes you will have an output value of 30. If the PV starts to increase, the error will decrease, and the CV% will increase more slowly. Even if the PV is only 1% below the SP, the CV% will still be increasing (although very slowly by this time) until the PV is equal to the SP.
- 42) Change your Reset Ti to 0 to remove the integral influence, then set Ti to .5. You will see the Integral control is repeating the proportional gain every 30 seconds now which is twice as fast. If the value of Ti is set too high it could cause the system to become unstable. The Instruction Set Reference Manual recommends that the Integral value be set to the natural period of the oscillations that occurred when calibrating the Kc.
- 43) Now that you understand how integral control works, lets try this on your heating process.

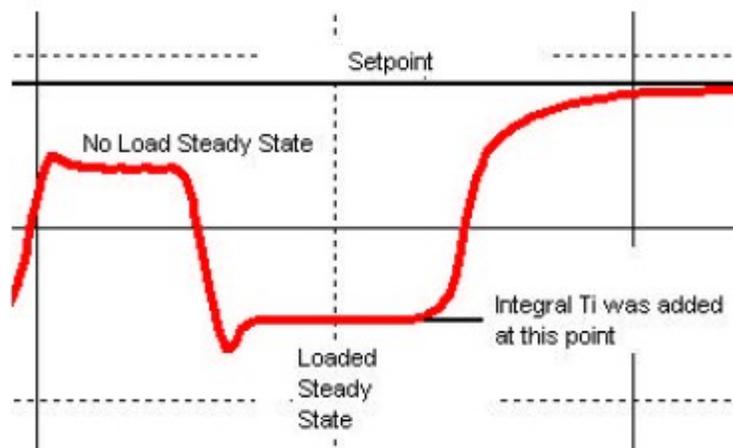
- 44) Go back to RSView, and for review, change your load to 0% and observe how the loop reacts. Then change your load to 100% and observe how the loop reacts. Also notice that the steady state process variable is well below the setpoint.



- 45) We are going to set the  $T_i$  value to the natural period of the  $K_c$  calibration that you recorded earlier. This should push the PV (temperature) up to the setpoint.

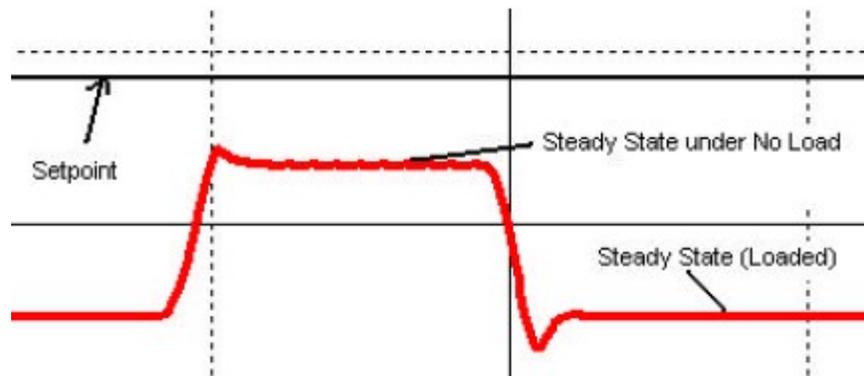
- 46) Go to RSLogix, and look at Ladder 8. Set the  $T_i$  value of the PID instruction. Immediately go back to RSView and observe what effect Integral control is having on your system.

- 47) You can see that right away when  $T_i$  is added to the equation, the output increases sharply due to the large error. The integral control continues to add output to the control variable at increasing slower rates until the PV is driven up to achieve the Setpoint. Now change your load between 0 and 100 to see how stable the loop is. You will see fluctuations immediately, however, no matter what the load, the Setpoint is achieved and the loop becomes stable within a reasonable period of time.



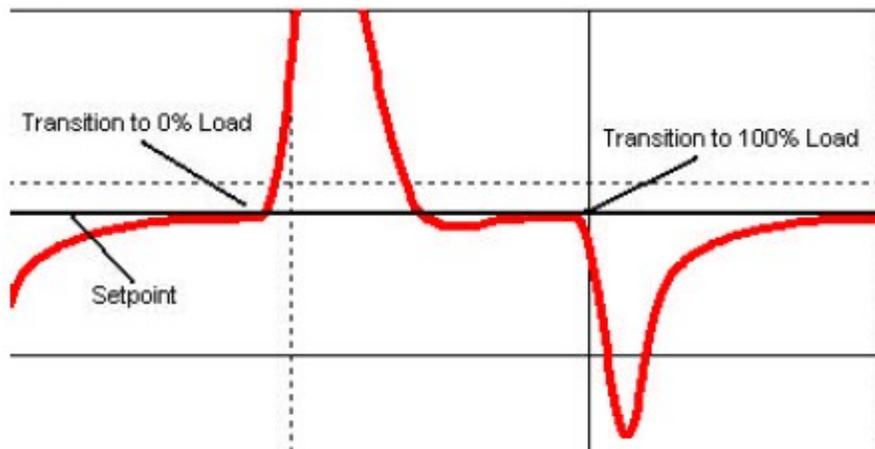
48) When removing the load, the PV immediately goes out of range of the chart. The reason for that is that for 100% load, the control variable required is very high. When the load is removed, and is no longer removing heat from the system, the heat bands are still hot, and take some time to begin to cool.

49) Similarly, when heating up the system, the heat bands are cold, and require some amount of time to become hot enough to start adding heat to the system. For this example, the power level the PID controller is wanting to achieve is called the requested power. The amount of energy the heating bands are adding to the system is called the actual power. When there is a change in requested power, it takes some time for the actual power to achieve the requested power level.



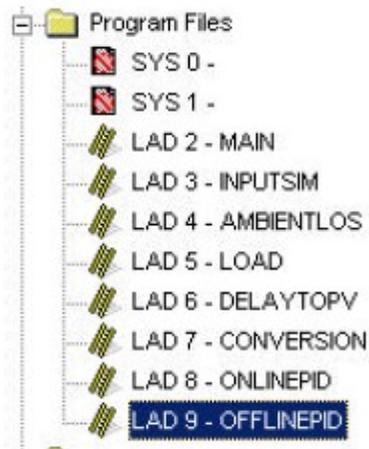
50) Now Compare these 2 graphs. To the left is a graph of how the system reacted with proportional control only. Notice the steady state temperatures. The larger the load, the lower the temperature of the system. Even at no load, due to ambient losses, the steady state temperature was well below the setpoint.

51) Now that we have  $T_i$  added to our controller, you will see that we now achieve the SP.



## Derivative Worksheet

- 52) The proportional and integral control work well in maintaining a desired temperature. The last part of PID that we are going to discuss is the DERIVATIVE component. With derivative action, the controller output is proportional to the rate of change of the measurement or error. You do not always need derivative action, and on some processes, it should never be used. Since derivative is based on the rate of change of error, if you have a noisy PV, large amounts of gain can be generated causing the loop to become unstable. Derivative action on noisy loops can cause the valve to become 'jittery' and decrease the life of the valve. An example might be a flow process at a sewage treatment facility. The material flowing through the system might have various degrees of viscosity causing intermittent large changes in the PV. If the controller attempts to act on these changes, the loop could become unstable. Because of this effect, derivative action usually causes more harm than good for flow control processes and therefore should not be used.
- 53) Imagine a heating process. Derivative action can be useful on a heating process by anticipating changes. If load is dramatically increased on a system, the PV is going to begin to decrease rapidly. The proportional and integral gain are not going to start adding large amounts of output to the CV until the error increases significantly. The derivative action will immediately detect that the PV is rapidly decreasing, and start to generate output immediately.
- 54) The SLC-500 Instruction Set Reference Manual states that a good rule of thumb to use when setting the derivative  $T_d$ , is one-eighth the value of  $T_i$ .
- 55) Because of the nature of the heating process, we have been using, Derivative action would not be significantly noticed. We are going to use our test PID instruction in Ladder 9 to observe the effect of the derivative on a process.



56) Let's go into the Setup Screen for our PID instruction, and configure the parameters as follows:

The screenshot shows the 'PID Setup' window with the following parameters:

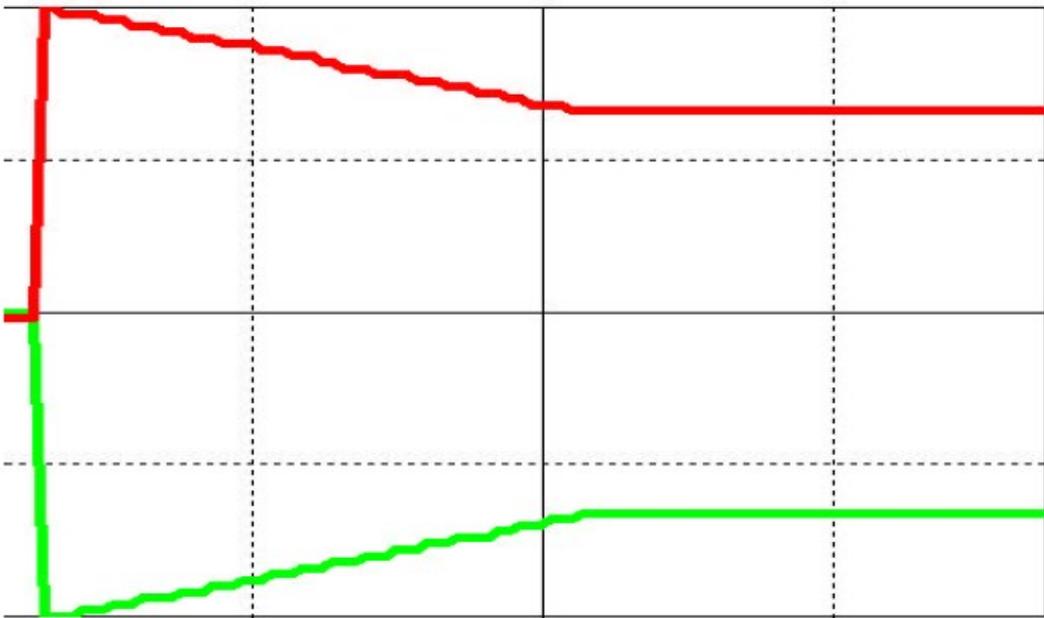
Section	Parameter	Value
Tuning Parameters	Controller Gain Kc	1.0
	Reset Ti	0.0
	Rate Td	0.04
	Loop Update	1.00
	Control Mode	E-SP-PV
	PID Control	AUTO
	Time Mode	TIMED
	Limit Output CV	NO
	Deadband	0
Inputs	Setpoint SP	100
	Setpoint MAX(Smax)	100
	Setpoint MIN(Smin)	0
	Process Variable PV	0
Output	Control Output CV (%)	100
	Output Max CV (%)	100
	Output Min CV (%)	0
	Scaled Error SE	100

57) Since derivative gain is based on a rate of change, we are going to use a signal generator to feed a ramping PV into our system. Go back to your RSVIEW project, and open the display called "noprocess". This screen will allow you to control the ramping signal generator, monitor the CV%, and change the Td of the controller.

0  
Rate Td (/100)

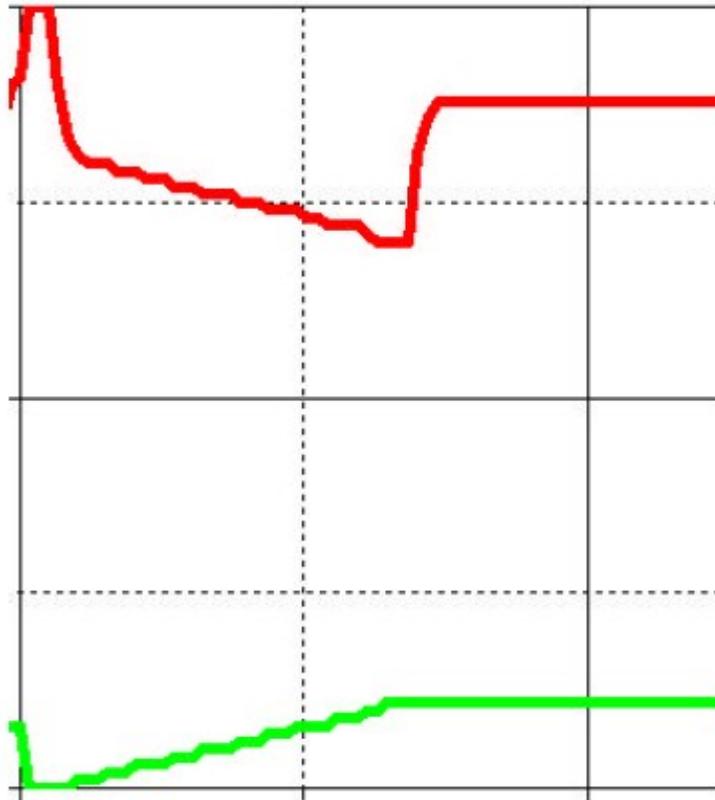
Active	Not Active	Enabled
Ramp Up	Ramp Down	Enable Ramping
		Reset

- 58) Set your Rate Td at 0. Enable your ramping action by clicking the enable button, and then start your signal generator ramping up.
- 59) Watch your trending chart in RSView. You will see the PV (green line) at the bottom fo your chart. This PV might be representing a temperature. The red line represents your power output. Since our Kc gain is set to 1, the CV is an exact mirror of the PV.
- 60) The chart below is what you observed with no derivative action. Notice that the output power decreases proportionally as the PV increases. If the PV stops increasing and remains the same, the Output power stops decreasing and remains the same as well.



- 61) Next we are going to put some derivative into the process, so you can see its effect on the output power. Before we do this, lets try to anticipate how the derivative action is going to change the output power. Remember the derivative action attempts to resist change. As long as our PV is changing, we are going to have some derivative action. If our PV stops changing, then we are no longer going to have the derivative action. The derivative sees that our temperature is increasing rapidly. Therefore, it will attempt to decrease the CV as long as this fast rate of change exists. Similarly, if the derivative sees the temperature falling rapidly, it will attempt to add output to the controller to compensate.

- 62) To keep you from having to switch back to RSLogix, you have a numeric entry in RSView that will allow you to change the Td value. We are going to enter the value of 100 into RSView, and this value will show up on the setup screen of the PID instruction as 1.00.
- 63) Press the RESET button in RSView, and then enable the ramping action. Start your PV ramping up. Notice the way the Control Variable reacts. Immediately it is decreased because of the rate of change of the PV. Now stop the ramping action. You will see the CV goes back to the proportional value, and derivative action no longer exists.



## Documentation

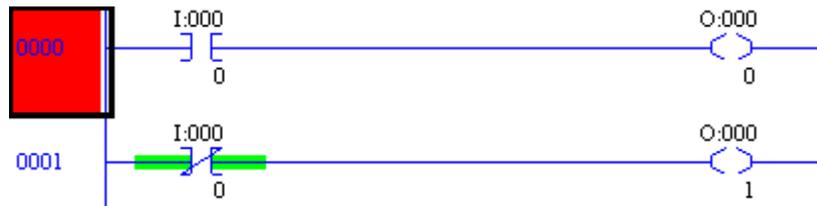
Documentation is very important in any project to help the programmer keep the project organized, and an essential aid to anyone who has to troubleshoot a project after it has been developed. In a properly documented project, a user can quickly locate an output, and easily navigate through a project to find what field devices are preventing an output from energizing. Four main forms of documentation are available for the project: symbols, descriptions, comments, and titles. Symbols and descriptions document individual addresses while comments and page titles document entire rungs or a group of rungs.

### Symbols

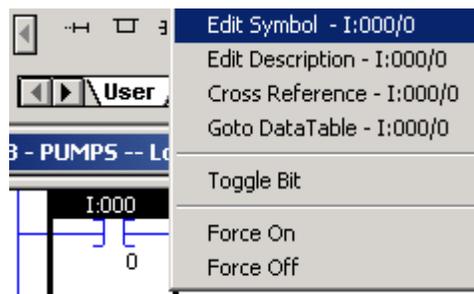
The first documentation we will discuss is the symbol. Symbols are synonymous with address. Once a symbol is attached to a bit in memory, no other bit can be assigned the exact same symbol. When writing logic, the symbol can be used as an address. Symbols can also be used to reflect designators such as the brass tags on valves in a system.

Symbols can also be grouped into categories. These methods will be discussed later in this document.

- 1) Enter these two rungs of logic (Use address that apply to your system). These two rungs control a main and backup vacuum pump. No one can know this without documentation, so let's start putting some symbols on the addresses.



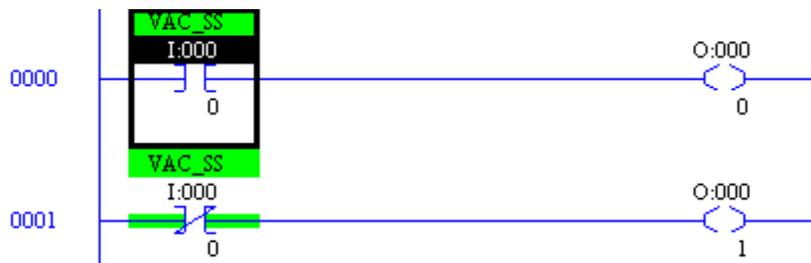
- 2) One way to add a symbol to an address is to right click the bit in logic, and choose 'Edit Symbol'.



3) Type the symbol name you would like to attach to the bit, and press enter. There are a few rules to follow when entering symbols.

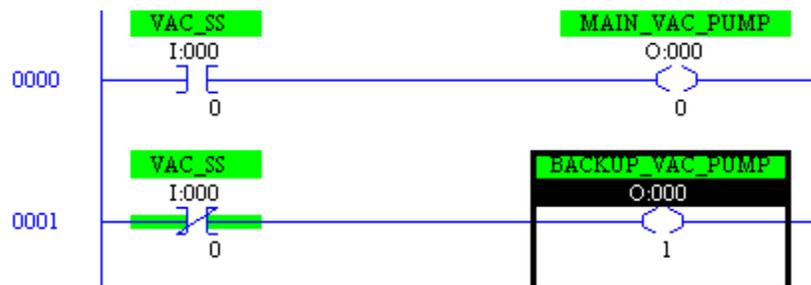
- 20 character limit
- No spaces (underscores are allowed)
- Cannot be only numbers
- Cannot end in a reserved bit or word name such as `_pre`, `_acc` or `_dn`
- My not contain the following characters: `~`!@#$%^&*()+-=[ ] { } \ : ; "<>' , . / ? |`
- Cannot consist of a number followed by a radix designator: D, O, H, E, or B (The software will interpret this as a number with a specific radix)
- The letters O, I, and S cannot be followed by only a number. This will be interpreted as an address

Notice what happens when you type `Vac_SS` as the symbol:

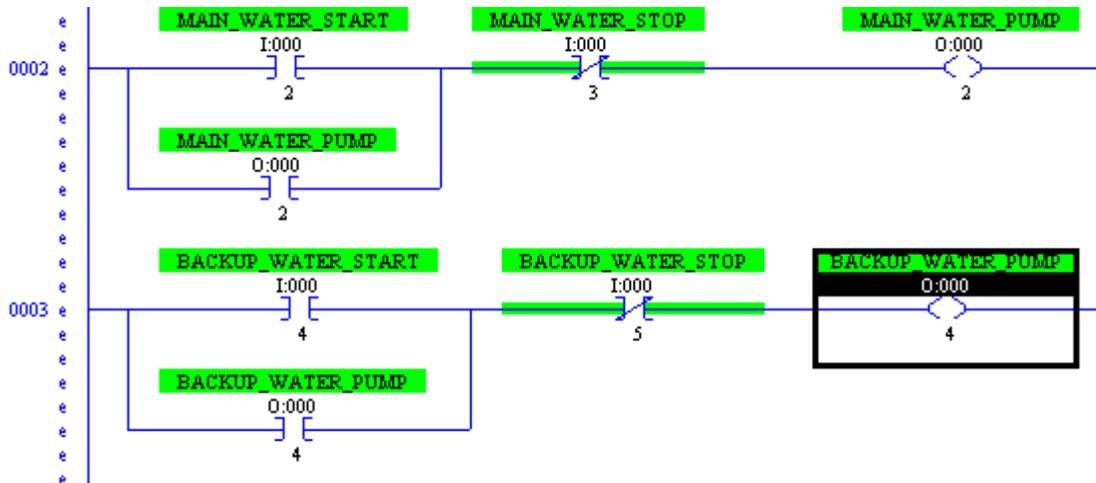


Wherever the address appears in logic, you will also see the symbol.

4) Go ahead and document the outputs as follows:



5) For more practice, enter the following logic and documentation into your project. (Use addresses that will work on your system)



Another method to add symbols is to use the Address/Symbol editor found in the Database folder at the bottom of your project tree.



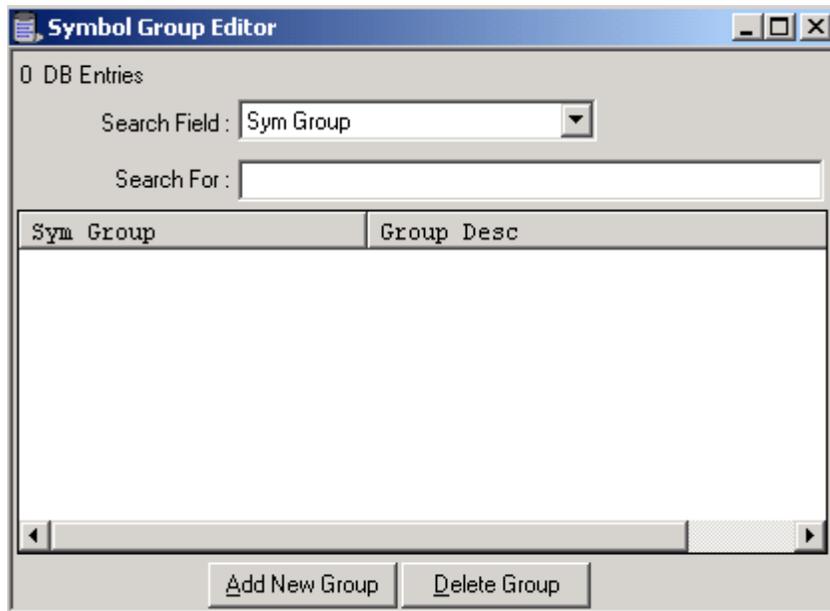
6) When entering a symbol in logic, the symbol actually goes into the documentation database. Open the Address/Symbol editor, and locate the symbols we typed in the ladder view.

Address	Symbol	Scope	Sym Group	Description
I:000/0	VAC_SS	Global		
I:000/2	MAIN_WATER_START	Global		
I:000/3	MAIN_WATER_STOP	Global		
I:000/4	BACKUP_WATER_START	Global		
I:000/5	BACKUP_WATER_STOP	Global		
O:000/0	MAIN_VAC_PUMP	Global		
O:000/1	BACKUP_VAC_PUMP	Global		
O:000/2	MAIN_WATER_PUMP	Global		
O:000/4	BACKUP_WATER_PUMP	Global		

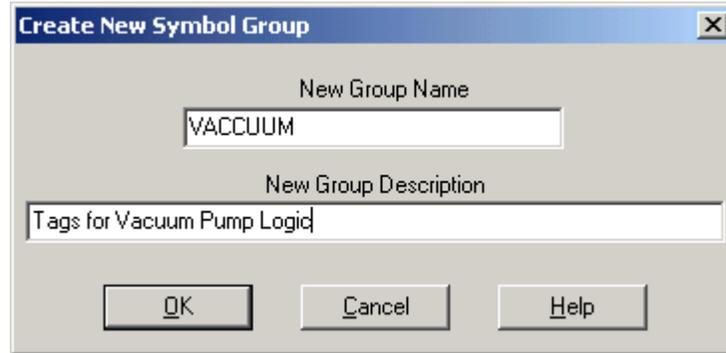
7) You will notice the symbol editor allows you to enter or delete records from the data base directly. Notice a field for the symbol group. If you are going to add logic using symbols only as your form of addressing, you will need a way to organize your tags into groups so they are easier to find. You do not have the ability to change the group yet because we have not created any. Look at the symbols you have so far, and think about how you would organize them. For this example, we'll create two groups. One for Vacuum symbols, and another for Water symbols. Close out of the Address/Symbol editor, and open 'Symbol Groups' from the database folder in the project tree.



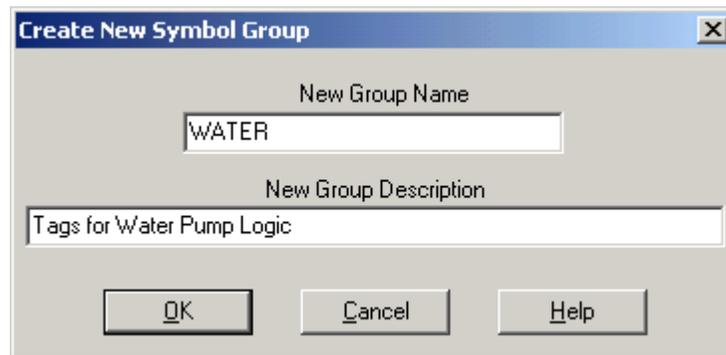
8) You will see a screen similar to the one shown below. Click 'Add new Group' at the bottom of the window.



9) Set up your new group as shown below for Vacuum pumps, then press OK.



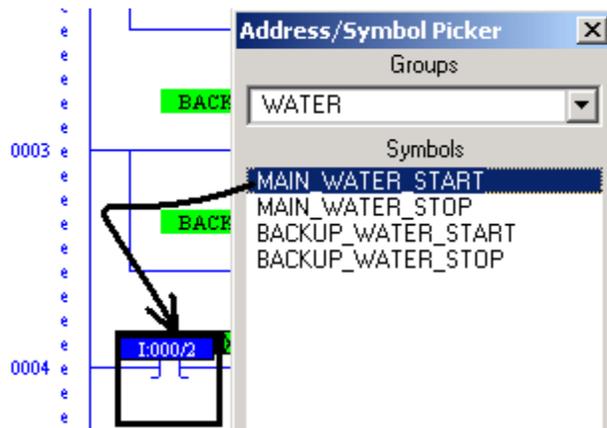
10) Next, Add another new group as shown below:



11) Now go back to the Address/Symbol editor, and you will be able to assign each tag to a group as shown.

Address	Symbol	Scope	Sym Group
I:000/0	VAC_SS	Global	VACCUUM
I:000/2	MAIN_WATER_START	Global	WATER
I:000/3	MAIN_WATER_STOP	Global	WATER
I:000/4	BACKUP_WATER_START	Global	WATER
I:000/5	BACKUP_WATER_STOP	Global	WATER
O:000/0	MAIN_VAC_PUMP	Global	VACCUUM
O:000/1	BACKUP_VAC_PUMP	Global	VACCUUM
O:000/2	MAIN_WATER_PUMP	Global	VACCUUM
O:000/4	BACKUP_WATER_PUMP	Global	VACCUUM

12) Now that the symbols are in groups, you could drag symbols from the Address/Symbol picker, and drop them in your logic. Go ahead and open the Address/Symbol picker, and create a test rung to give this method a try.

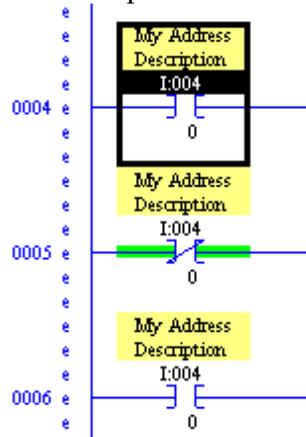


13) Delete your test rung when you see how this feature works. Be sure to explore the other options on the symbol picker such as config, and show descriptions.

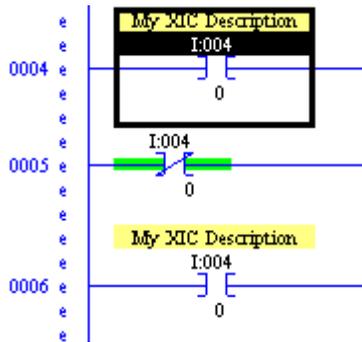
## Descriptions

Symbols were synonymous with the addresses, and had many limitations. Symbols were just a designator for a bit, but descriptions have fewer limitations, and can be used multiple times throughout a project. There are two forms of descriptions... Instruction descriptions, and address descriptions. Address descriptions are probably the most commonly used, and appear wherever the address is used in logic. Descriptions can be no longer than 5 lines with 20 characters on each line. Instruction descriptions appear wherever the address is used with a particular instruction. Look at the example below:

Address Descriptions work like this



Instruction Descriptions work like this

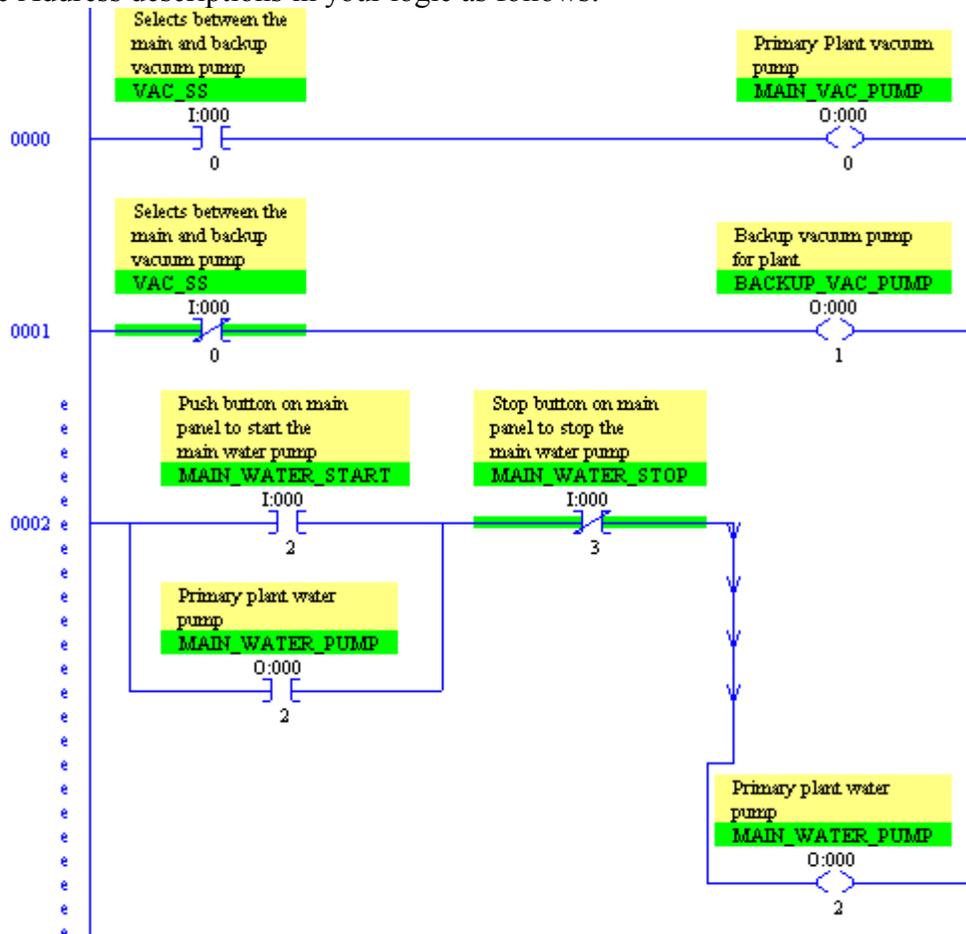


In each case, the description was only applied to the first XIC. When an address description was applied, it appeared at all three locations where the address was used. When the instruction description was applied to the first instruction, it appeared every where the address was used on an XIC instruction throughout the project.

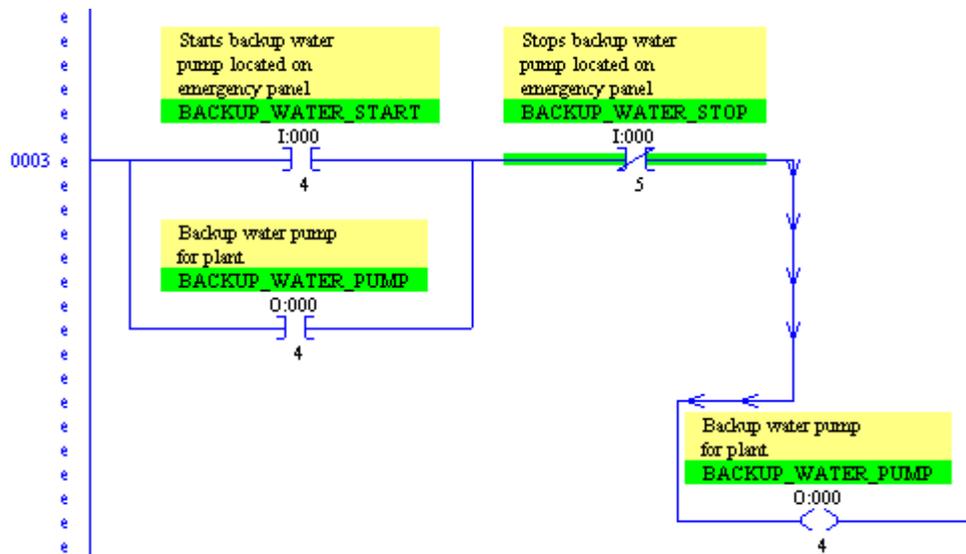
There are several ways to add descriptions to bits. One way is to right click the bit in logic. Another way is go use the database. Address descriptions appear in the Address/Symbol Editor, and Instruction descriptions appear in the Instruction Comments editor.



1) For our example, Right click on the instructions in logic, and choose 'Edit Description'. Use Address descriptions in your logic as follows:



Continued on next page



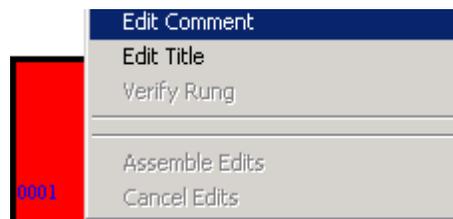
If you look in the Address/Symbol editor, you will see your changes have been logged to the database. Be sure to save your work.

I:000/0	Selects between the main and backup vacuum pump
I:000/2	Push button on main panel to start the main water pump
I:000/3	Stop button on main panel to stop the main water pump
I:000/4	Starts backup water pump located on emergency panel
I:000/5	Stops backup water pump located on emergency panel
I:004/0	
O:000/0	Primary Plant vacuum pump
O:000/1	Backup vacuum pump for plant
O:000/2	Primary plant water pump
O:000/4	Backup water pump for plant

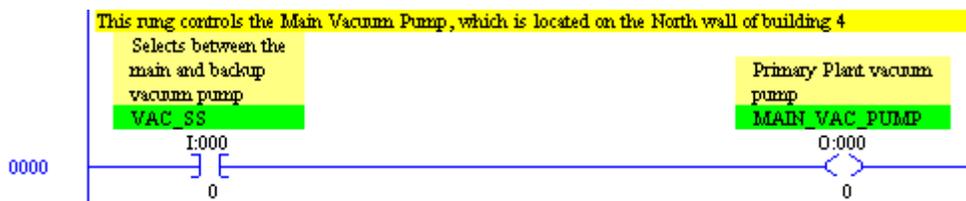
## Rung Comments

Rung comments describe the purpose of a rung in ladder logic. Rung comments can be up to 64,000 characters long, and can be attached to the rung number or the output. If the comment is attached to the output address, the rung comment will appear wherever the address appears as an output on a rung. With good programming practice, your program will probably only have one OTE instruction for any given address, so the comment will appear only one time. However, in the case of a Latch and Unlatch instructions, the comment would appear at both locations if the Latch and Unlatch are in separate rungs.

- 1) To add a description, right click the rung number just to the left of the rung, and choose 'Edit comment'.



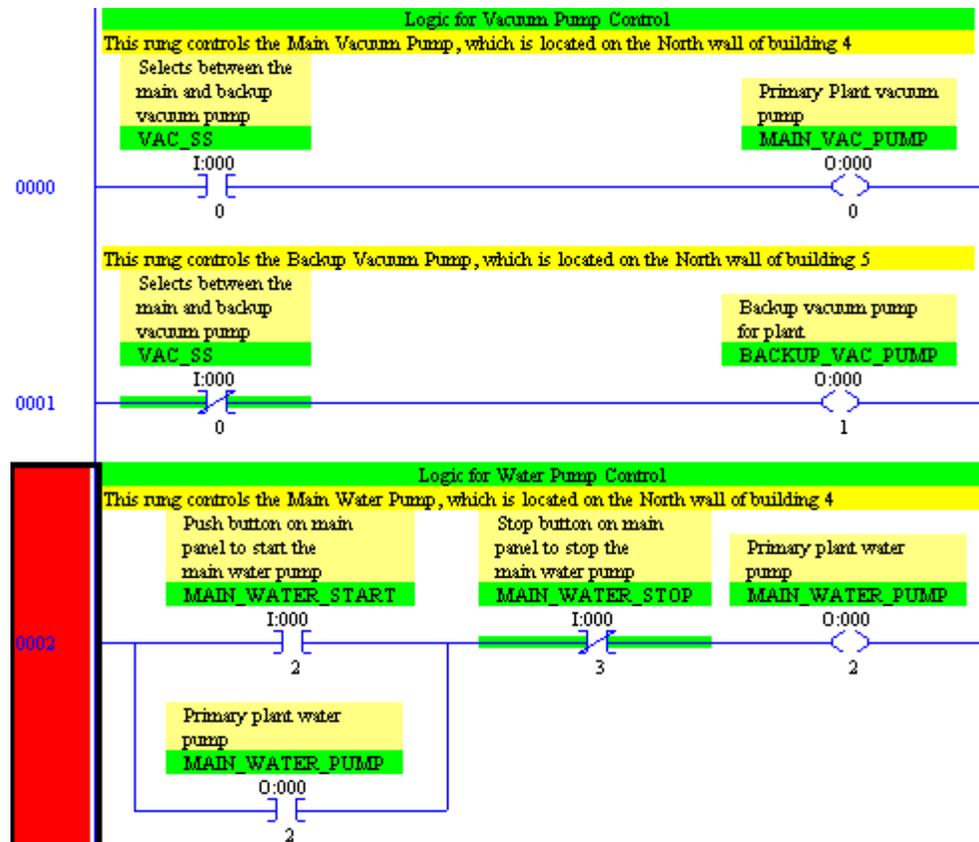
- 2) Go ahead and add comments to all four rungs of logic explaining the purpose of the rung. Below is an example of rung 0.



## Page Titles

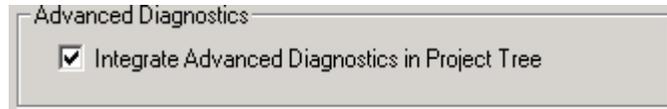
Page titles are a very important part of documentation. Page titles should be used as 'thumb tabs' in routines to mark sections or blocks of logic that perform a specific purpose. Advanced Diagnostics uses page titles to quickly locate an output in the program, so the troubleshooter can then use cross referencing to find what field devices have failed. In later versions of RSLogix, Page Titles (Advanced Diagnostics) can be integrated into the project tree. With page titles, you are limited to 80 characters.

- 1) Adding Page Titles are very similar to adding rung comments. Right click the rung number and 'Edit Title'. Titles and comments can be modified from the same window. In our sample logic, there are two sections. One for Vacuum Pumps, and the other for Water pumps. Add your Titles to reflect the purpose of each section as shown below.

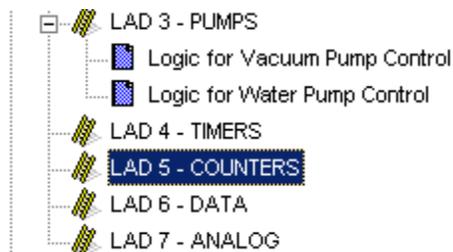


2) To integrate page titles into the project tree, click Tools | Options on the menu bar.

In the middle of the window, choose 'Integrate Advanced Diagnostics Into Project Tree'. Apply your changes then click OK.



3) Now look in your project tree. You see a + next to ladder 3. Expanding ladder 3 reveals the sections (page titles) we just created. Double click any page title, and RSLogix will take you directly to that section of logic.



## Advanced Diagnostics

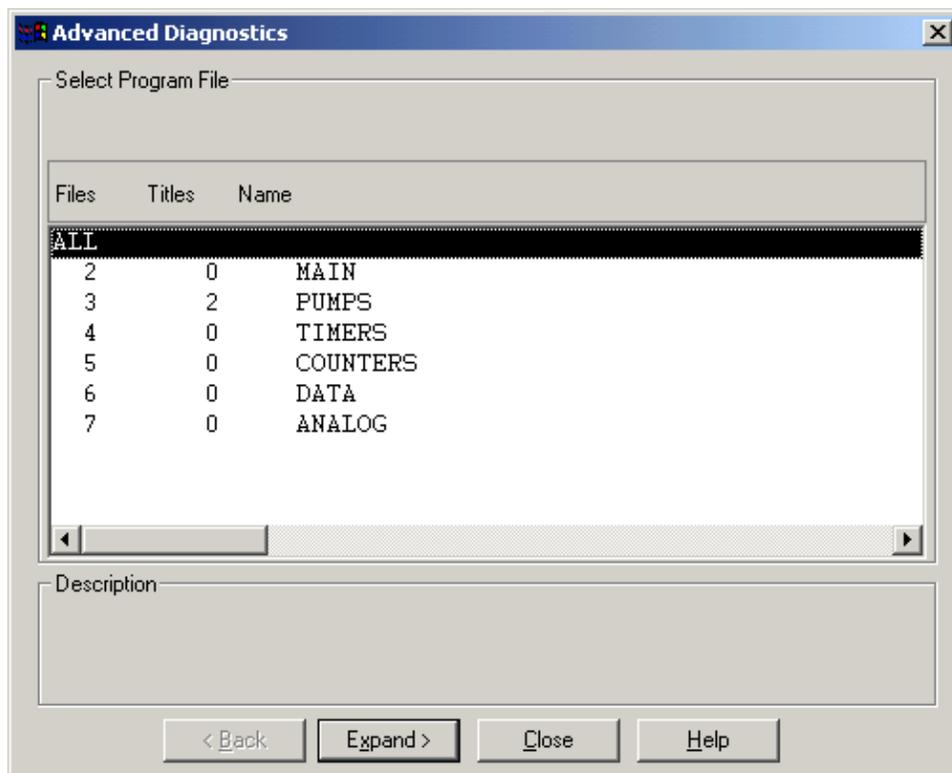
Advanced Diagnostics is a power troubleshooting tool built into RSLogix. Advanced diagnostics helps a user quickly located an output in the ladder logic. There are three steps: 1) Choose from a list of routines; 2) Choose from a list of page titles within the routine; and 3) Choose the output within the page title (block of logic) that you are looking for.

*In this example, we want to figure out what needs to happen for the backup water pump to run.*

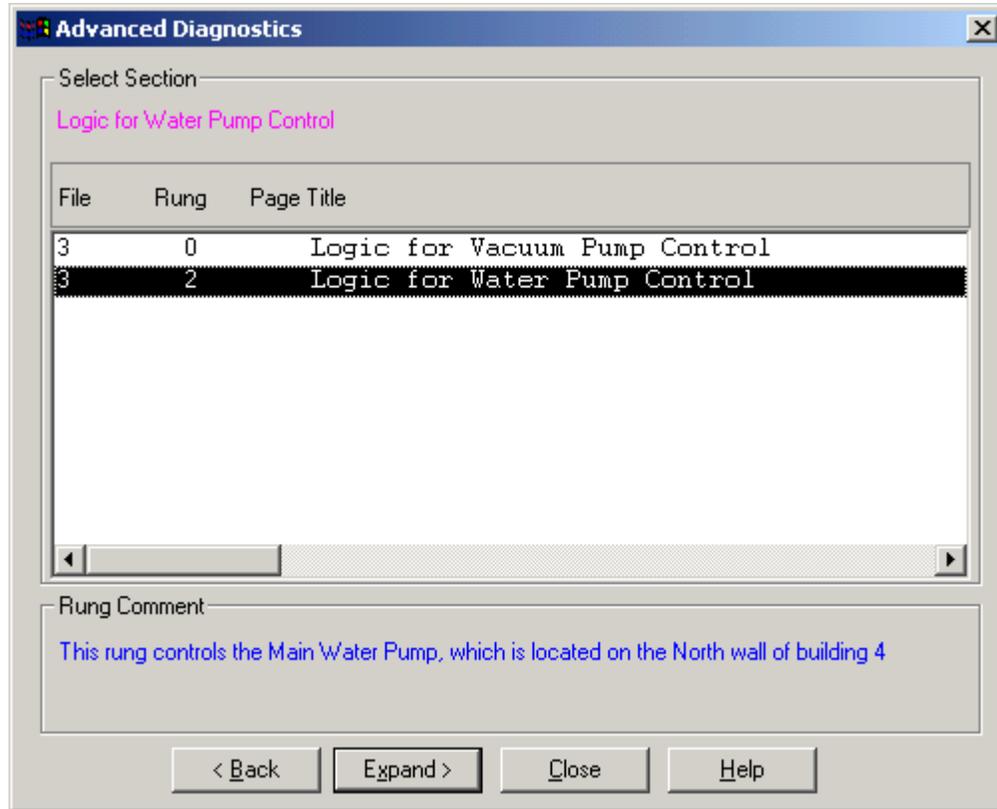
- 1) To start Advanced Diagnostics, you must have a project open then click Search | Advanced diagnostics from the menu bar. You will notice the short cut is to press CTRL-D.



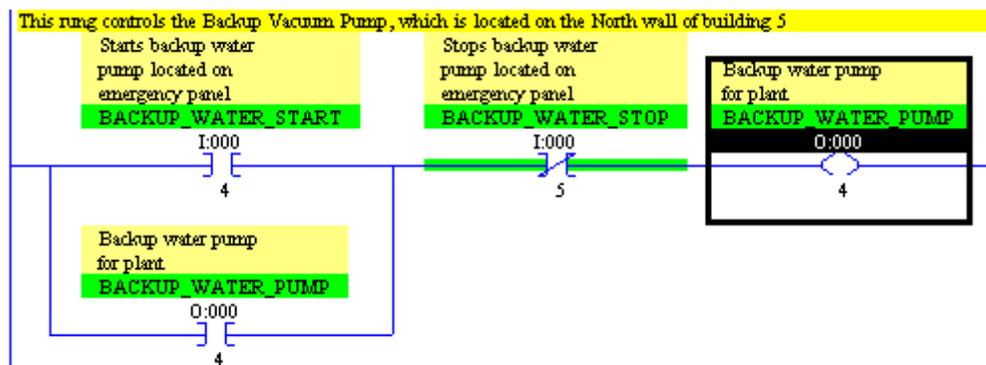
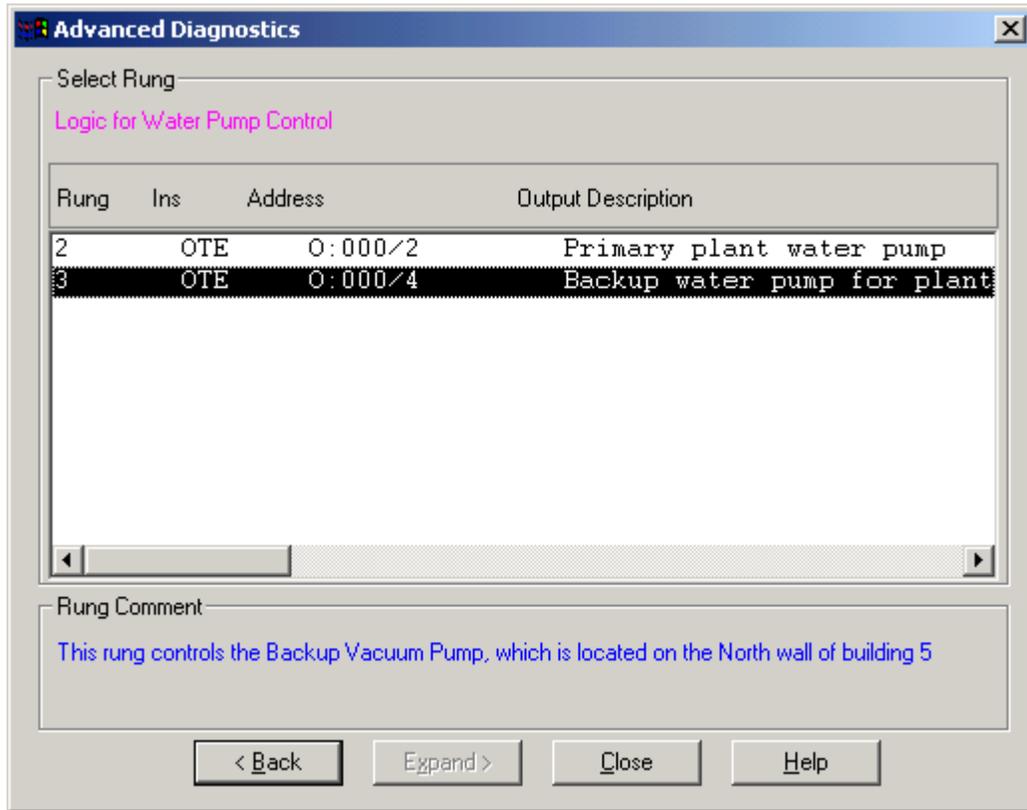
- 2) The first screen that appears is asking what routine you want to work with. From this screen, you can also see how many titles are in each routine. Since our Water Pump is down, double click 'PUMPS', or highlight 'PUMPS', then click 'Expand'.



- 3) Next, RSLogix is asking what page title you would like to view. Notice when you highlight a page title, the rung comment also appears for the rung in which the page title resides. Since our problem is with the backup water pump, open the page title for water pump control.



- 4) Next, you are asked which output to look at within the page title. Notice when you click an output, the rung description appears at the bottom. You also see the description of the output in the description field of the list. If you click on the backup water pump, nothing appears to happen on the advanced diagnostics screen, but something did happen in logic. Close the advanced diagnostics screen, and you are at the location in logic where you need to be.



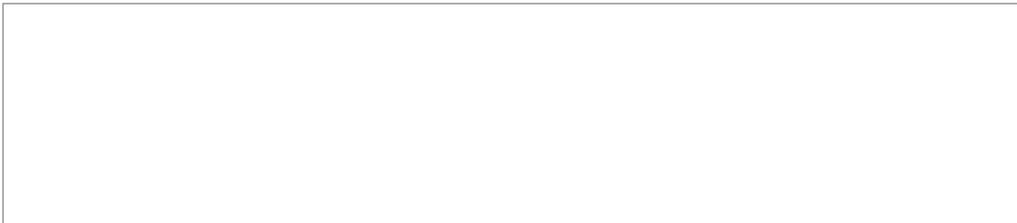
You can now see why the backup pump is not on. In this example, all instructions point to real devices, but if internal bits were used, you would now have to use Cross-Referencing to trace the field devices.

## *Importing and Exporting the Documentation Database*

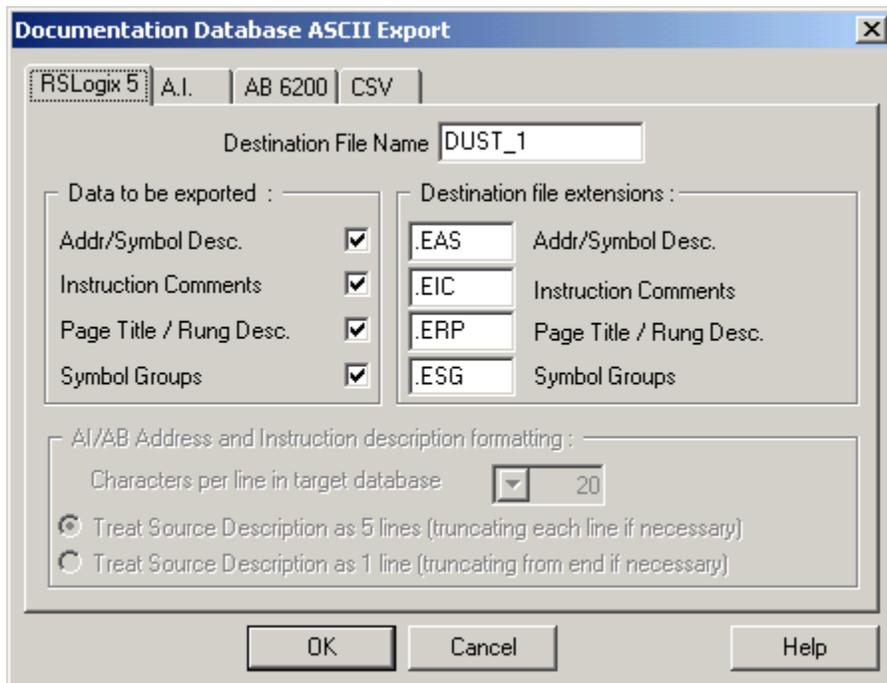
If documentation of a program gets lost, and you can still locate the original program file (from some time back), it is possible to export the documentation from the original program, and import documentation into the current running program. This would save you from having to re-document every address.

### **Exporting the Database:**

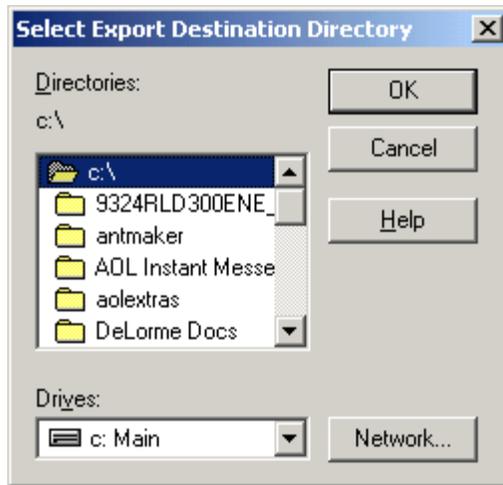
To export the documentation database, open the original program containing documentation, then click Tools | Database | ASCII export from the menu bar.



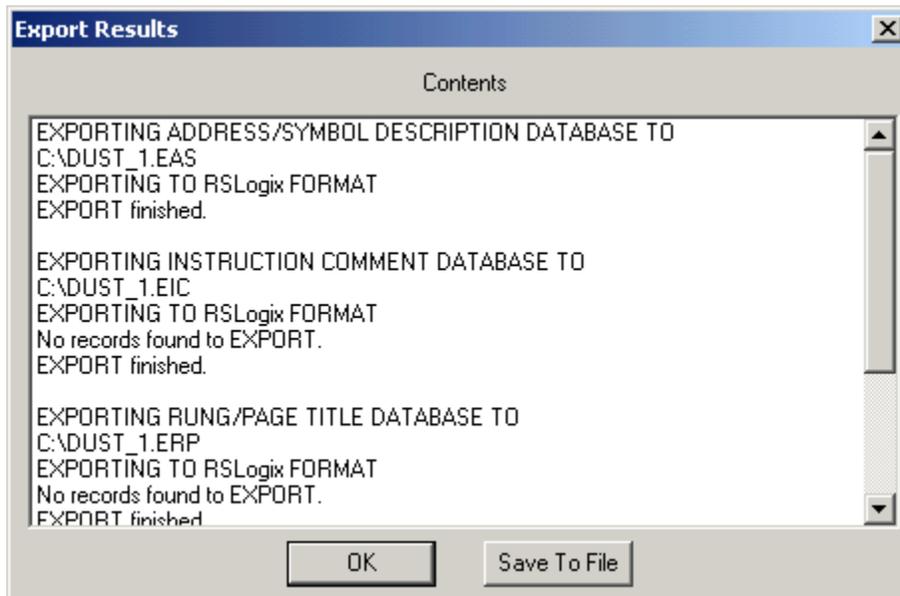
Next choose the format and files in which you wish to export the documentation. If you are importing all documentation into another RSLogix 5 project, press OK.



Next, you are asked for a destination directory. Choose the destination for all four files that are to be exported. For this example, we choose the C: drive. If you are saving the documentation to a floppy drive, choose A:.



You will now see a report of the results of the export. You can choose to save this report to a file, or just click OK.

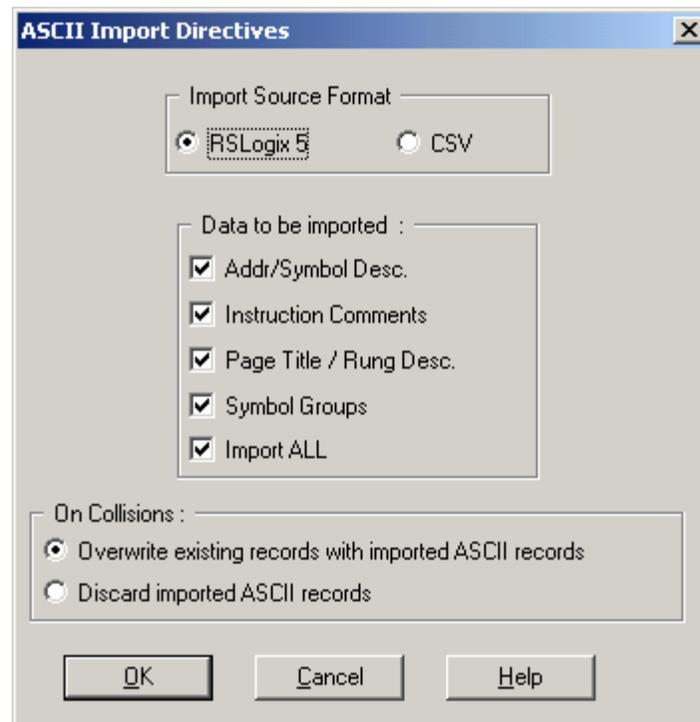


## Importing the Database

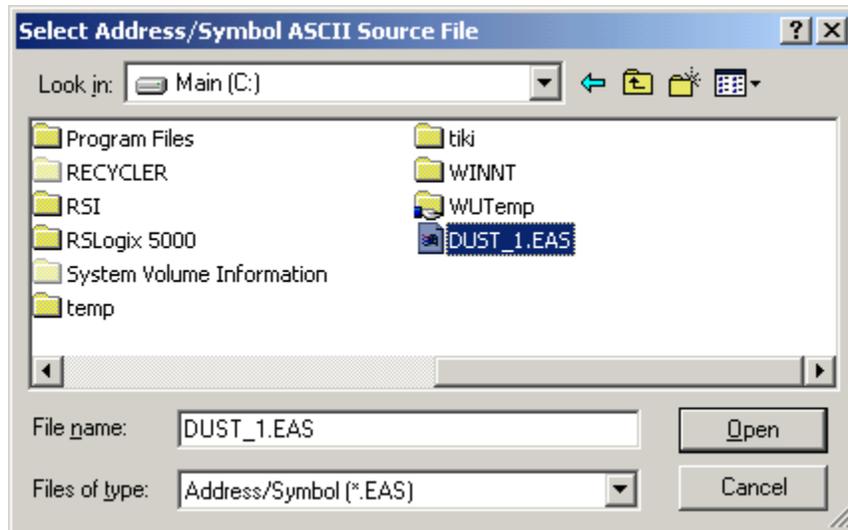
To import a documentation database, be sure you have the undocumented project open, then click Tools | Database | ASCII Import from the menu bar.



Next you are asked the format of the documentation you are importing. If all documentation has been exported by RSLogix 5 for RSLogix 5, then press OK if you want to overwrite existing records with imported records. If you already had some documentation in your program that you want to keep (and not import new documentation for those addresses that are already documented), Then choose to discard imported records where collisions occur.

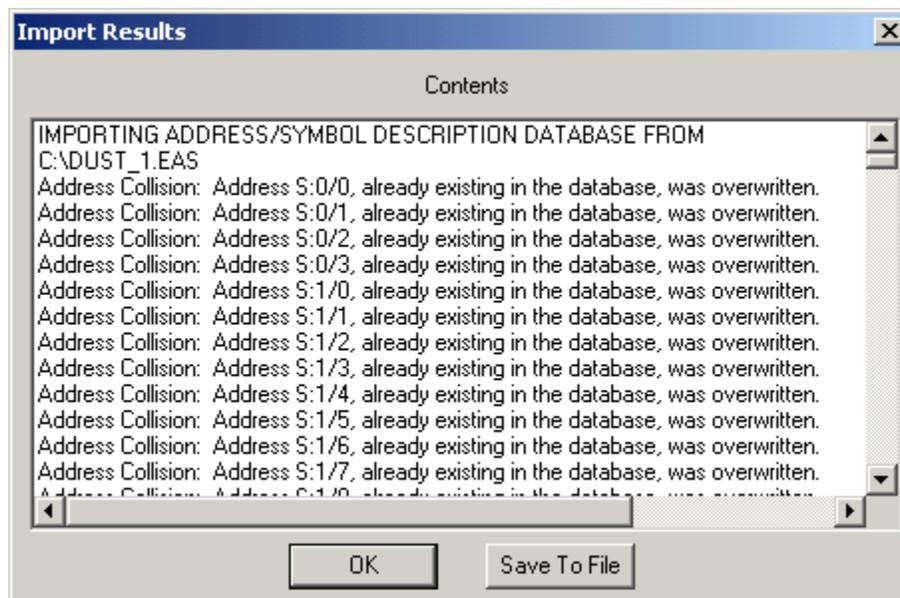


In the pull down menu of the Look in box. Choose the location of the documentation files that we exported earlier. For this example, our documentation was on the C: Drive. Click on the documentation file (EAS extension), as shown below, and then press *OPEN*.



Repeat this procedure for the other 3 documentation files... EIC, ERP, and ESG.

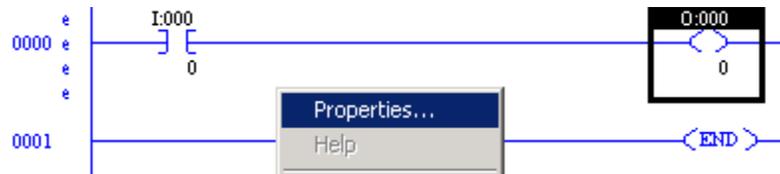
You will then be told the results of the import operation. You can choose to save the results to a file, or hit OK to finish. You should not have documentation.



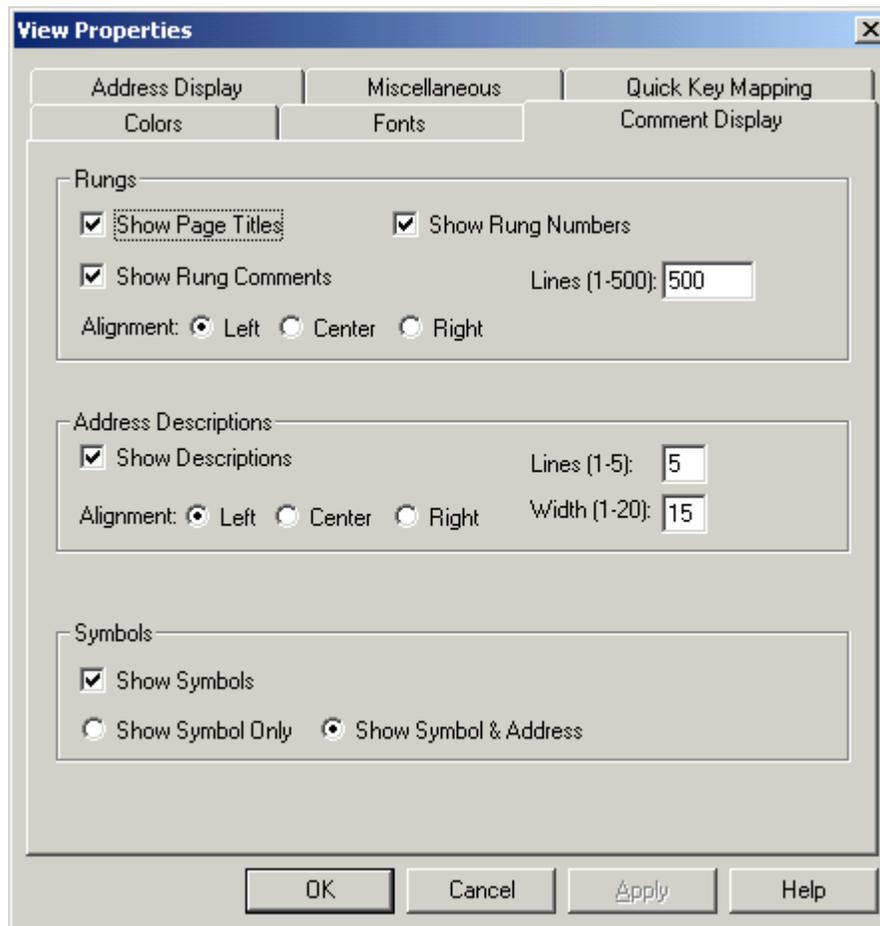
## Changing the View Settings

View settings are available for the user to set his own environment under RSLogix. Each user of a professional windows operating system (based on the windows login) can have his own view settings.

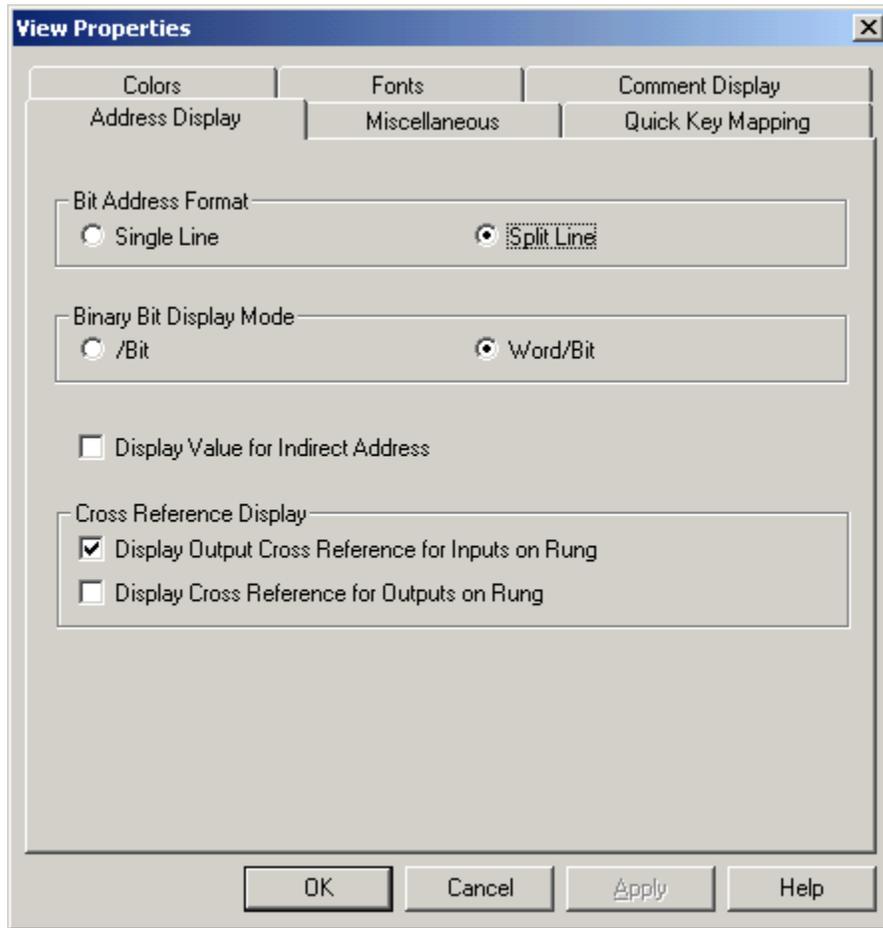
- 1) To access the view settings, right click the background of the ladder view and choose properties.



- 2) The Comment display tab allows the user to show or suppress certain pieces of documentation.



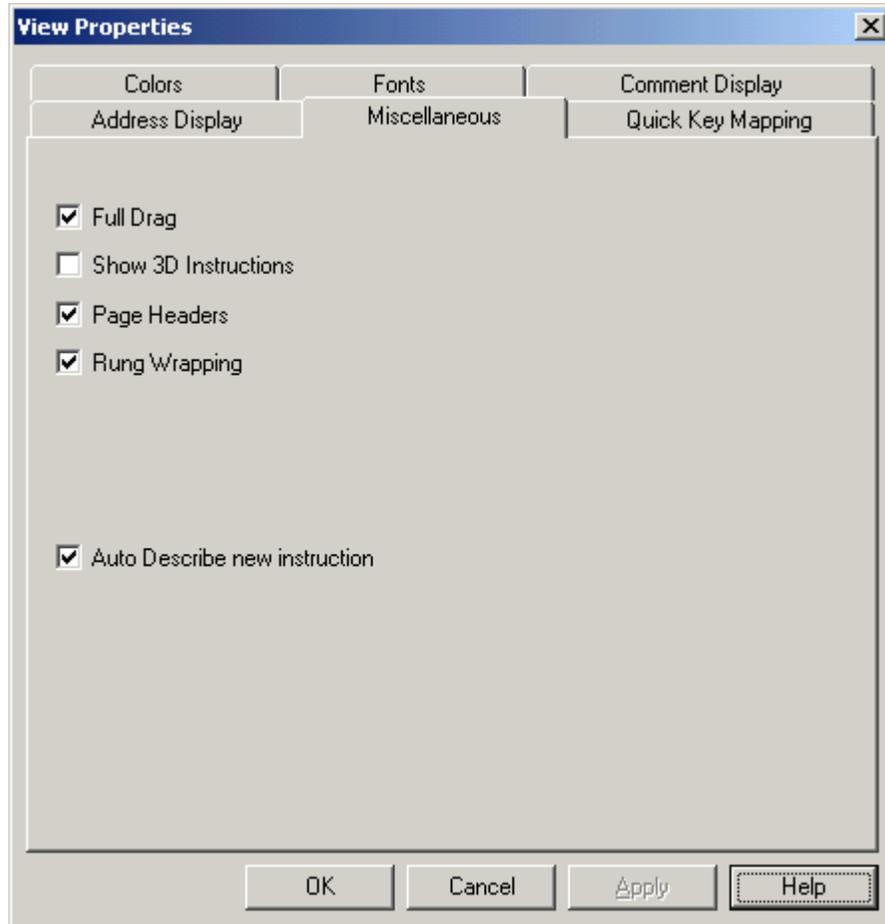
- 3) The address display tab shows information about the way addresses appear in the ladder logic. For the Bit Address format, if 'Single Line' is chosen, the entire address will appear above the instructions. In split line mode, the bit address is displayed underneath an instruction. Below is an example of Single Line (to the left) and Split Line mode (to the right).



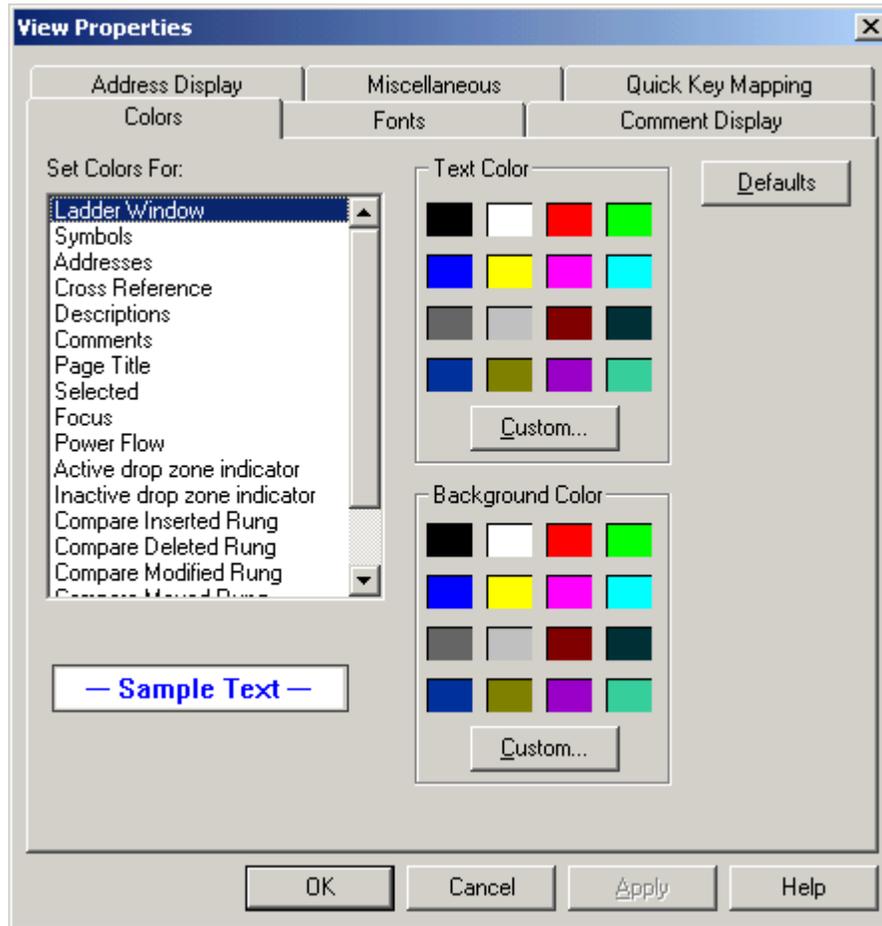
For the Binary Bit Display Mode: in the /bit format, all bits in the file are numbered consecutively. In the Word/Bit format, the Word level is added to the address. B3/16 and B3:1/0 are exactly the same address. Because a word contains sixteen bits, word 0 contains B3/0 to B3/15. The very next bit is B3/16 which is in the next word of memory.

Other examples: B3/96 is bit 0 in word 6, so therefore the long address would be B3:6/0. B3/159 is bit 15 in word 9, so therefore the long address would be B3:9/15. If you multiply word 9 times 16 bits, then add the bit number (15), you will find the short address to be B3/159.

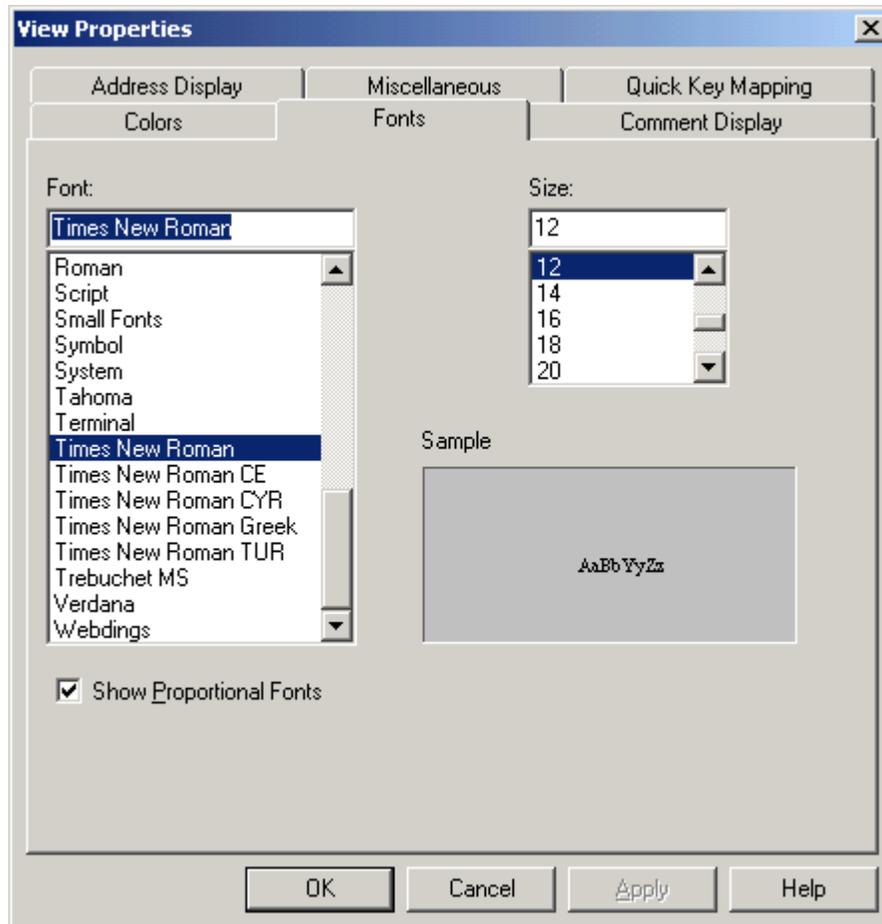
- 4) On the Miscellaneous tab, **Full Drag** mode lets you see the address or symbol when dragging an instruction in the ladder view. **3D instructions** put shadows behind instructions to give your logic a 3 dimensional look. **Page headers** display information about the ladder under the title bar in the ladder view. **Rung Wrapping** will wrap rungs to the next line if they are too large to fit on the current page. If rung wrapping is shut off, you must scroll to the right if a rung is too large to fit in the ladder view. **Auto Describe new instruction** will bring up a dialog screen for you to enter a description and symbol when new addresses are used in the project.



5) The Colors tab lets you change the colors of items in the ladder view. Click the item on the left you wish to set colors for, and then choose a text color and background color to the right. You will notice the Sample Text will change so you can see how the colors will look in the ladder. You also have a button for default colors if you would like to revert to the original color settings for RSLogix.



- 6) Below is an example of the Fonts tab. Usually the default font is fine, but occasionally, on some systems, you may want to change the font type and increase or decrease the font size.

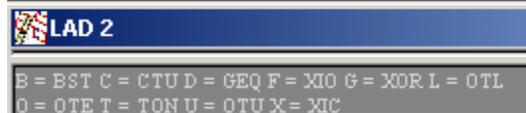


- 7) Under the Edit Menu, quick key mode can be turned on or off. Quick key mode allows a user to add instructions to ladder logic with the stroke of one key. When quick key mode is on, a header appears below the title bar indicating how the quick keys are mapped. In this example, if quick key mode was on, the user could press the letter O when editing ladder logic to insert the OTE (output to energize) instruction.

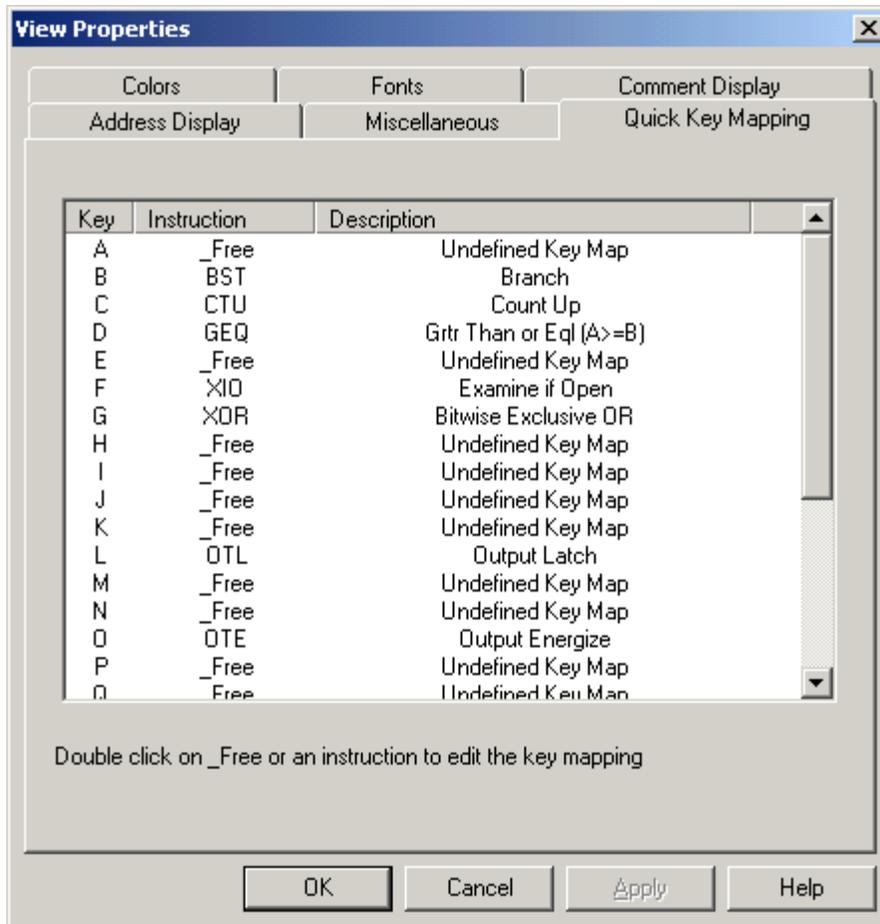
From the Edit Menu, Quick Key mode can be activated:



While in quick key mode, a header below the title bar indicates mapping:



Under View|Properties, Other keys can be mapped:



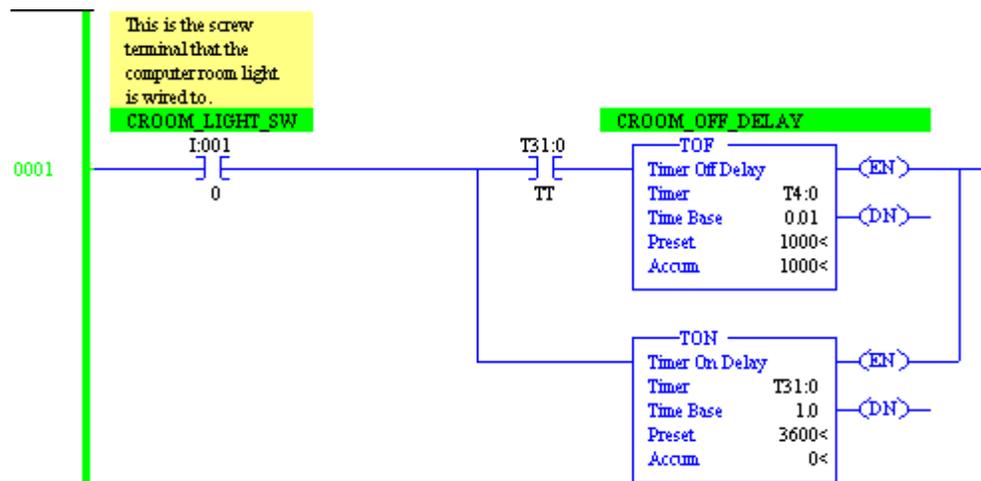
## On line Editing

There are five basic steps in performing an edit on line. 1) Start Edits, 2) Make Changes, 3) Accept edits, 4) Test Edits, and 5) Assemble edits. Although these steps seem very simple there are a few rules to watch out for.

- You cannot expand or create a data table on line in the SLC, and you must be in program mode (or off line) to expand a data table on line in the PLC-5.
- You cannot make an on line edit if the key switch is in Run Mode.
- In the PLC-5, if backplane switch 8 is on, you cannot make an on line edit.
- You do not need to perform an on line edit to directly change a value in the data table such as the preset of a timer or counter.
- If the processor is in program mode, you do not need to test and assemble after accepting.
- If the processor is in program mode, and a rung is deleted, there is no warning.

Let's walk through the 5 step procedure:

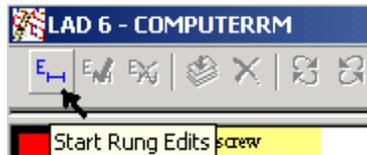
Look at the rung below. Our objective is to transfer control of the output to I:001/1. If you click on bit 0 and attempt to make a change, nothing happens.



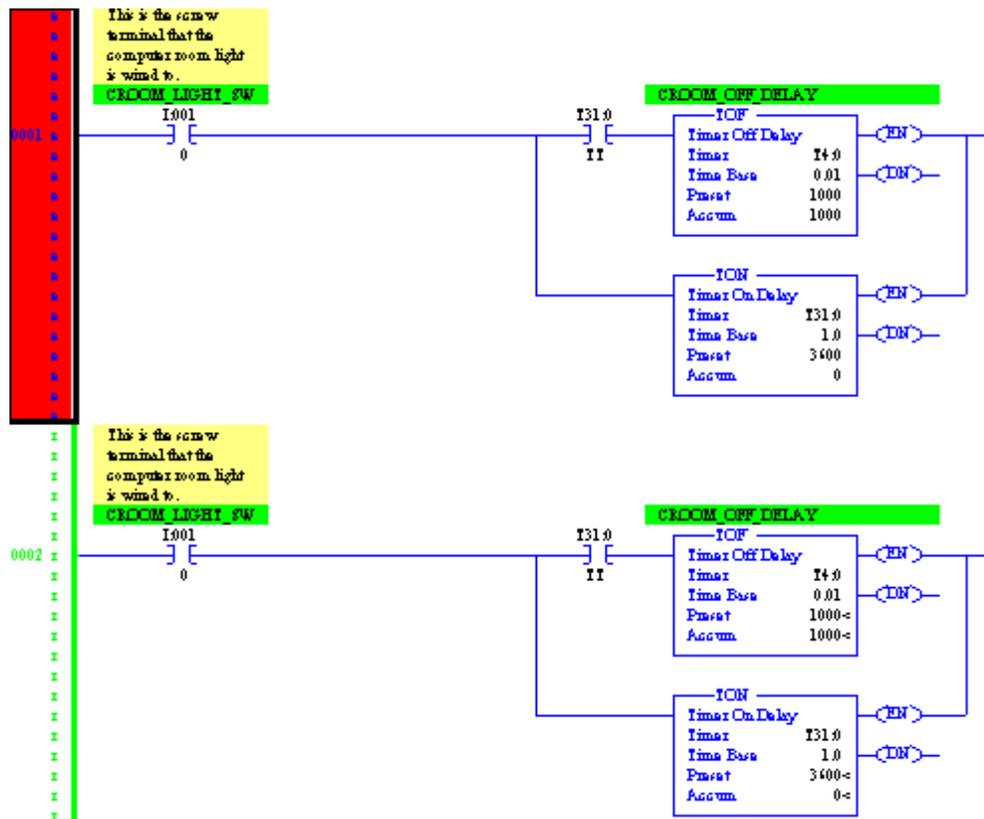
## Step 1) Start Rung Edits

The first step is to put the rung into edit mode. There are several ways this can be done:

- Double click the rung number
- Right click the rung number and start rung edits
- From 'Edit' on the menu bar, click rung edits, then start rung edits
- Click the start rung edit icon in the on line editing tool bar just above the ladder view



Notice that RSLogix made a copy of the rung for us to work with. By looking at the power rails, you can see the bottom rung is being executed by the processor, and the top rung is the one you need to make edits to. You will also notice the e (edit) and r (replace) in the margin are lower case. This means the edits are not in the processor yet. If you are adding new logic instead of modifying existing logic, this is the step where you add a new rung.



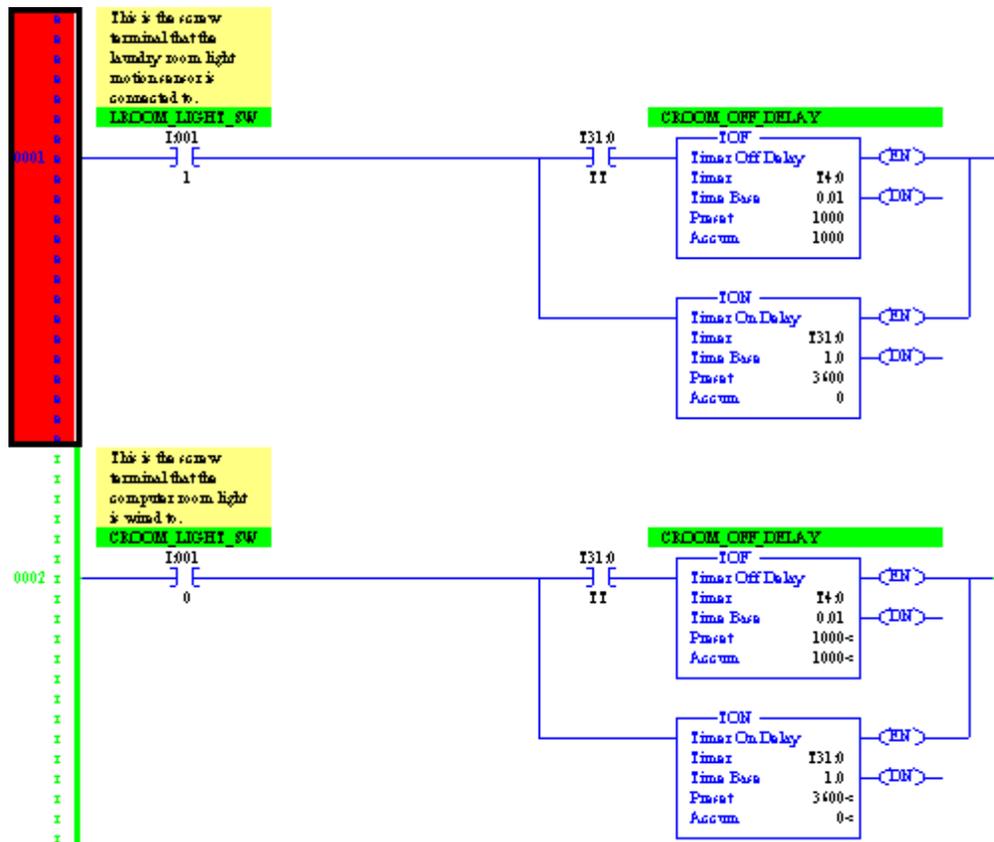
## Step 2) Make Changes

Now that the rung is in edit mode, changes can be made.

If you were adding a new rung, you can now add the new logic to the rung.

Be careful not to add any logic that will fault the processor or cause damage to personnel or equipment. In the next step, the changes we make will be sent to the processor.

In this example, bit 0 is being changed to bit 1 on the input. If you are using an SLC, the addressing will be slightly different.



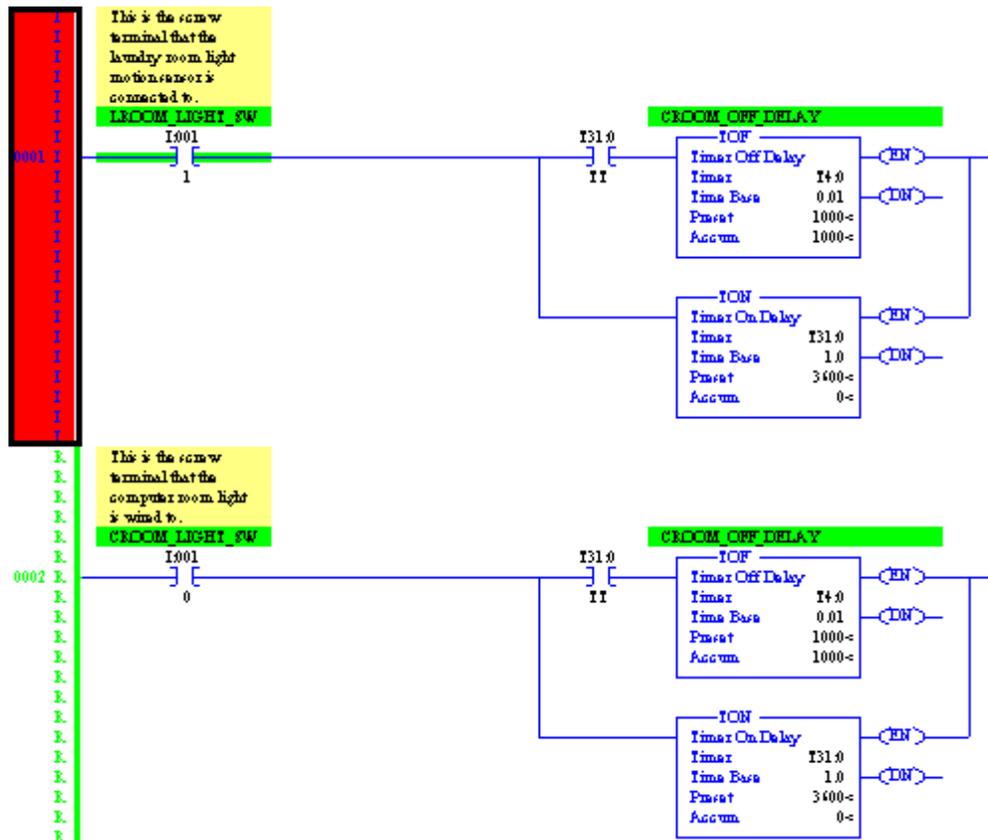
### Step 3) Accept Edits

Now that your rung is set up as you need it, it's time to send the edits to the processor. There are several ways to perform the next three steps.

- Right click the rung number, and accept edits
- Click Edit | Rung Edits | Accept rung from the menu bar
- Click the Accept Edits icon in the on line editing tool bar as shown below



If you need to accept multiple rung edits, you can hold down the CTRL key and click on the rungs you need to accept, or if you have many edits all together, you can click the first rung number, hold shift, then click the last rung number of the range you wish to accept. Notice in the margin rung 1 is marked for insertion, and rung 2 is marked for removal. The I's and R's are capital because the edits are now in the processor. Look at the power rails. You can see the old rung is still being executed by the processor.



## Step 4) Test Edits

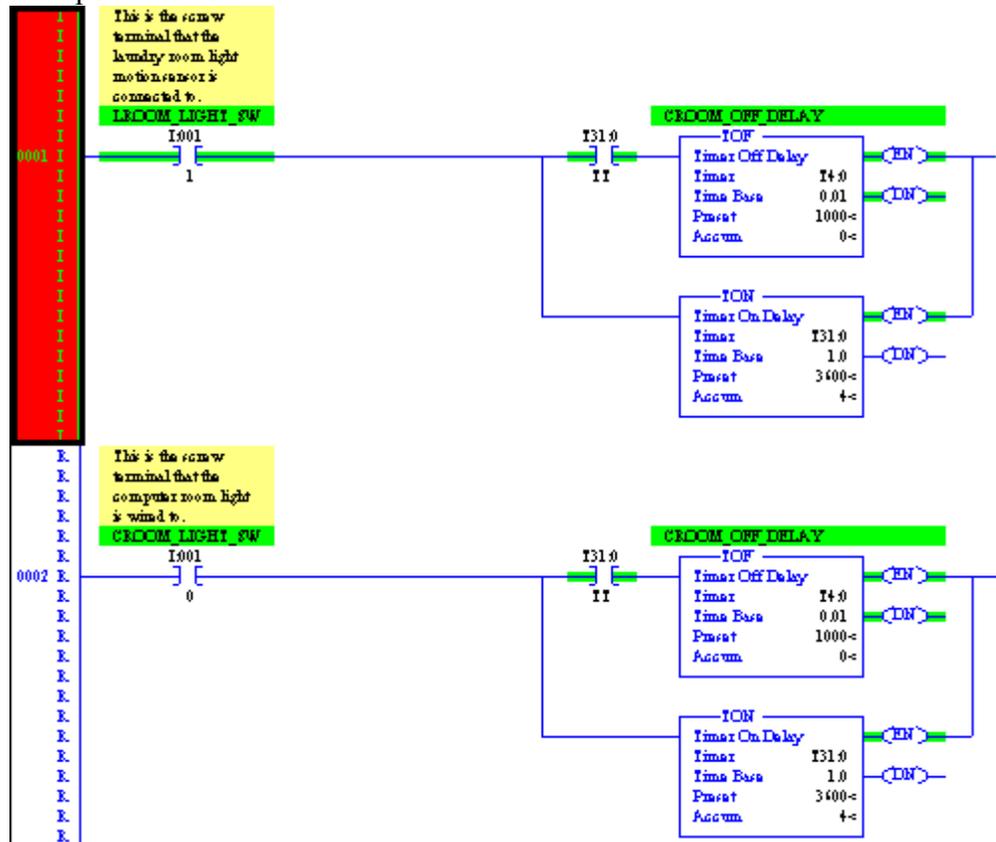
When you test edits, the new or modified rungs will become active. The old rungs will be left in the processor until we are sure our new rungs are working properly. Be aware that if you change an output address, there might no longer be logic writing to that address. This means that you could abandon a bit in the ON state.

If you are modifying an input type address you should also be careful. If the rung was previously true, you may want to make sure your new logic is also going to be true at the moment you accept, or the the output may shut off.

Let's test the edits, and you will notice the new rung(s) are active. If the edits do not work the way you anticipated, you can untest to revert to the old rung while you make other changes to the new rung.



Notice the power rails:

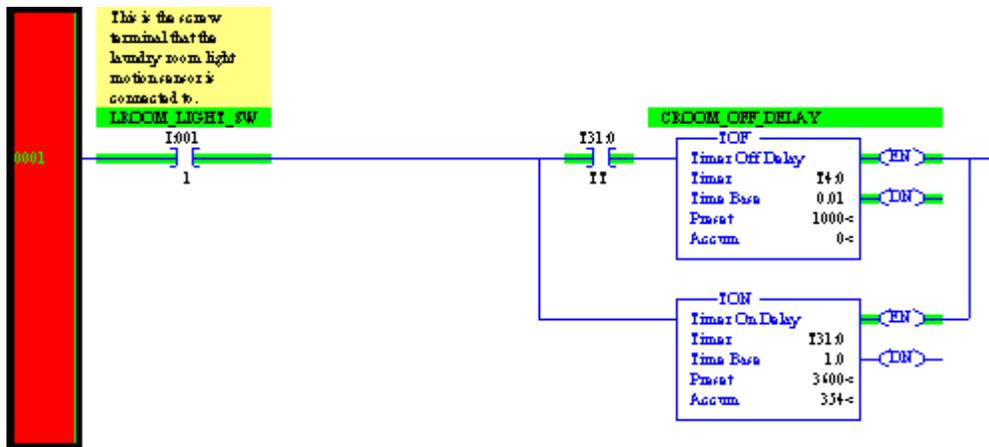


## Step 5) Assemble Edits

If your logic is working properly, go ahead and assemble the edits. Assembling removes the old rung, and the edit zone markers. After Assembling, you may want to save your work to the hard drive.



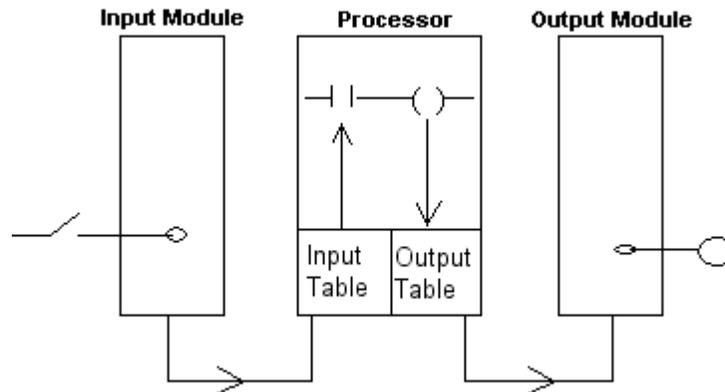
Notice the Logic now appears to be normal:



## ***Forcing I/O***

Forcing can be used for troubleshooting, and to some extent simulates real world jumpers. Leaving forces in the processor, or depending on forced I/O to make your equipment run is considered bad practice.

Look at the diagram below:



Under normal circumstances, the following events take place:

1. The switch is shut
2. A 1 appears in the input data table
3. The XIC instruction goes true
4. The OTE is enabled
5. A 1 is written to the output data table
6. The light will energize on the output module

Forcing the input:

If you place a jumper across the switch, you would have the same effect as the switch always being shut. A 1 would always be in the data table, the logic would be true, and the light would energize. The same effect applies to forcing. Forcing the input on would result in a 1 in the input data table for the switch, and all logic would be executed as if the switch was shut. The opposite applies to an OFF force. An Off force would be similar to cutting a lead on the switch. A zero would result in the input data table.

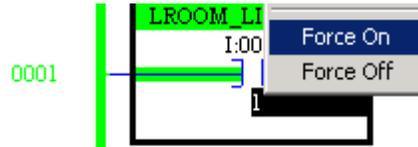
Forcing the output:

If you place a jumper to the output, the output table would still be a zero if the logic is false. Information does not flow from the output device to the output data table. Therefore, any XIC instruction that is looking at the output bit would also be false. The same applies to forcing. If you force an output device, the output data table will still be controlled by the ladder logic.

*Note: Even though forcing an output does not directly effect the data table, The field device itself could feed an input back into the processor causing other things to happen in logic. Know your system before using the force feature.*

There are several ways to force an output. Forcing can be applied from ladder logic, from the force table itself, or in the I/O configuration on a PLC-5.

In this example, we will force an input directly from ladder logic. Right click on the input address, and choose 'Force On'.



Notice the force light on the processor begins to flash indicating that forces are installed, but not enabled. The force can be enabled from the on line tool bar as shown below:



The force light on your processor will now be solid amber indicating that installed forces have been enabled. If we go to the data table, you will see that the input bit is on, and it is red indicating that a force has been enabled on the input.

Offset	17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0
I:000	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
I:001	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1



Forcing can also be performed directly from the force table. Scroll down toward the bottom of the project tree, and you will see the input and output force files. If we open the input force file, we can see what bit has been forced.

Offset	17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0
I:000	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
I:001	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.

The value of 1 indicates a bit has been forced on. The value of 0 indicates an off force, and a period indicates no force is installed on a particular bit. Forcing can be done directly from this table. If you right click on the table, forces can be enabled without using the on line tool bar. You can also use the Enable/Disable buttons at the bottom of the force table.

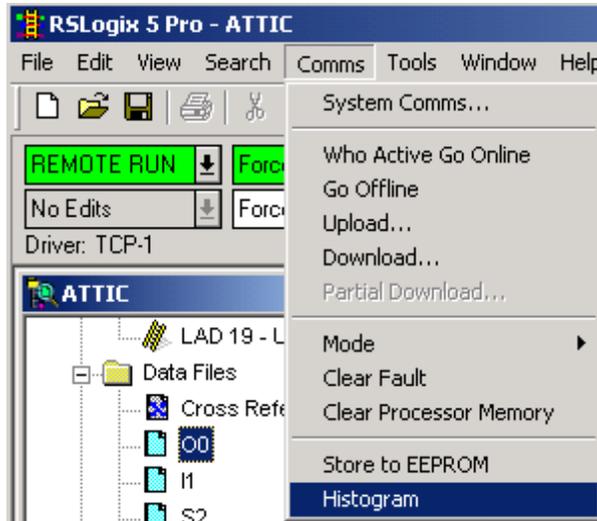
Radix:

Symbol:  Columns:

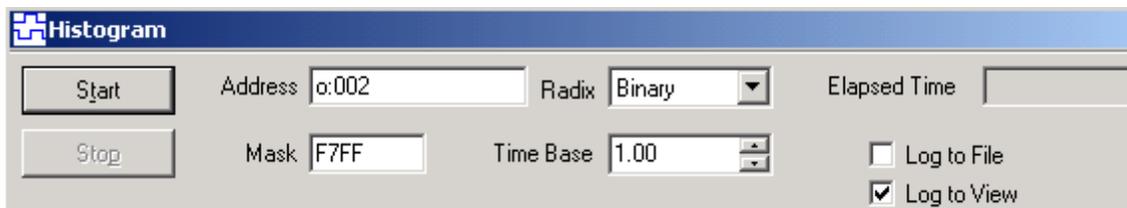
Desc:

## Using the Histogram

- 1) To use the Histogram, you must be on line with the processor, then click Comms | Histogram.

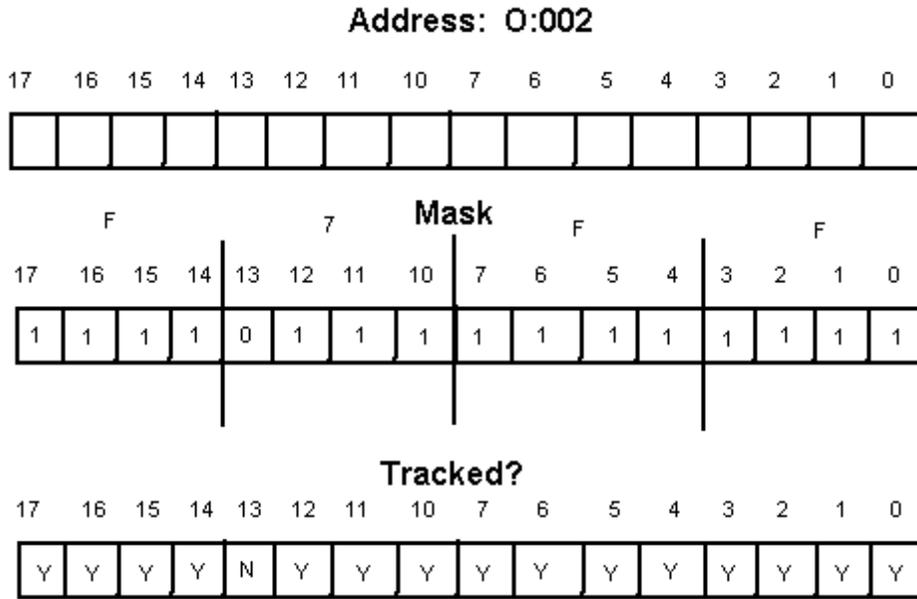


- 2) Next, you will be setting up the histogram.
  1. Enter the address of the word you wish to run the histogram on. A histogram can be run on most any 16 bit word. If you enter the address at the bit level, the mask will change so the histogram is only run on the bit you selected. This will be explained later.
  2. The Radix indicates the numbering scheme in which you wish to view the data. Options are: Binary, Octal, Hex, or Decimal.
  3. The mask is a hexadecimal number indicating which bits will be monitored in the address you entered. If you use a scientific calculator, specify a Hex radix, then enter a mask value. Convert to binary, and you will see that F7FF is equivalent to four 1's, a zero, and eleven more 1's. This means that all bits will be monitored except bit 13. (Look at the example on the next page).
  4. The time base is the amount of time between grid lines on the timing chart.
  5. If you choose 'Log to File', the histogram data will be stored in a file on your hard drive which can be viewed using a standard text viewer such as Microsoft Notepad. The location of this log file can be set in the histogram properties (right click on the histogram window).
  6. Log to view allows you to view the histogram after you start the chart.

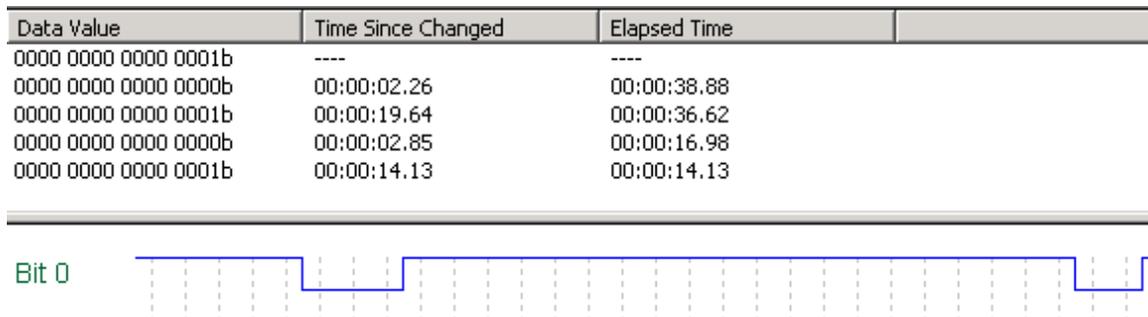


## Understanding the Mask:

This example shows what bits of the address you entered will be monitored with a mask value of F7FF:



3) Press the start button to begin tracking. In this example you see four transitions: Two ON transitions, and two OFF transitions.

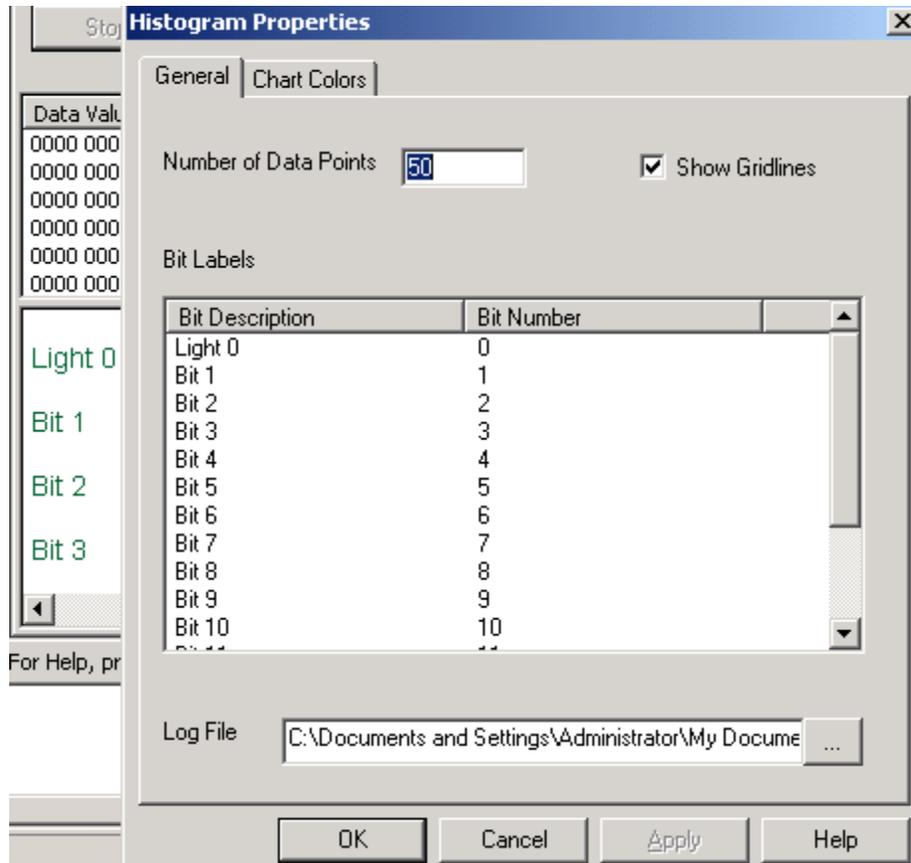


The histogram (top window) will catch most all transitions, however, the timing chart (lower window) works from a time base, and may not catch all transitions.

- 4) Special configuration for the histogram window can be accessed through the histogram properties. To access the properties, right click within the histogram window, and choose 'properties' from the window that appears.

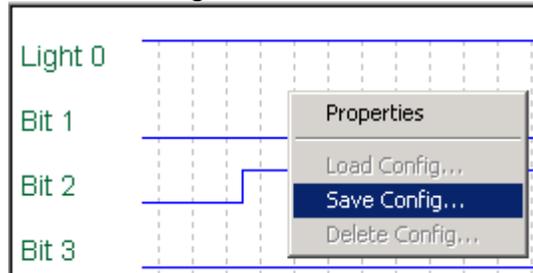


- 5) Look at the properties screen below:

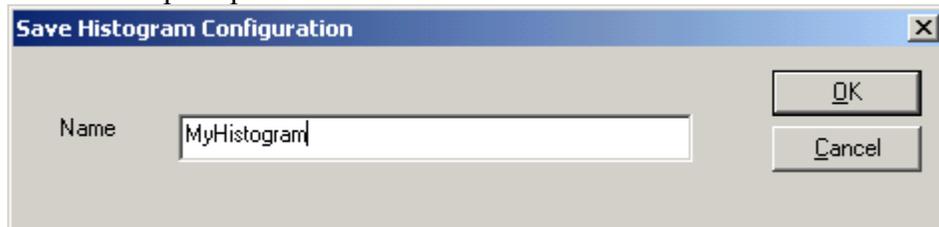


1. Select the number of data points you wish to log. The default value is 50. By choosing more data points, you will be able to stop the histogram, and scroll back through the timing chart.
2. The grid lines can be enabled or disabled on the timing chart.
3. To change a bit description, click on one of the bits such as bit0. Change the text, then press enter, and apply your changes. You will notice the text changes on the timing chart.
4. If you selected 'log to file' on the main histogram window, you see or change where this file is located on the system.
5. Click the chart colors tab to change the look of the main histogram window.

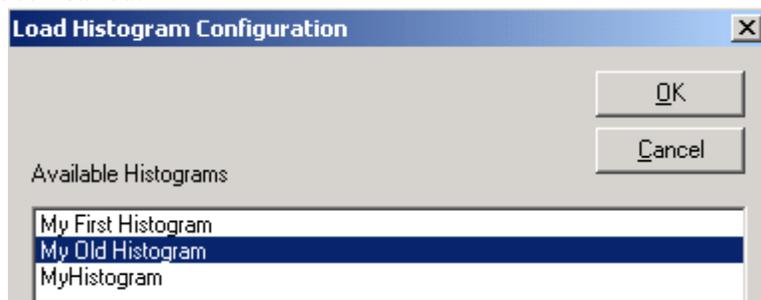
- 6) After you are finished with the histogram, you can save the configuration. Right click on the chart, and choose 'save config'.



- 7) You will then be prompted for a name:



- 8) You can also revert to other saved histograms if you right click the chart, and choose 'Load Config'. A window will then appear showing histogram configurations that have previously been saved.

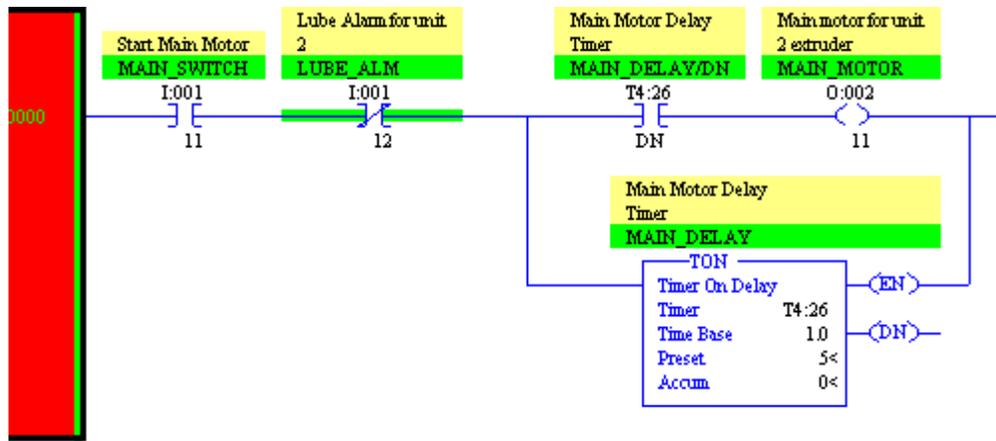


## Setting up a Custom Data Monitor

The Custom Data Monitor can significantly reduce downtime by allowing the user of RSLogix to create their own troubleshooting tool. If a pump quits running, a user could bring up the custom data monitor for that particular pump to see what input has not been made, or to see what alarm is locking the pump out just by looking down a list.

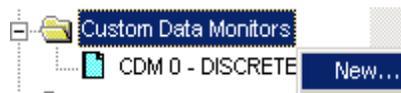
In this example, we will build a custom data monitor which will walk a user through troubleshooting a motor failure.

Look at the simple example below:

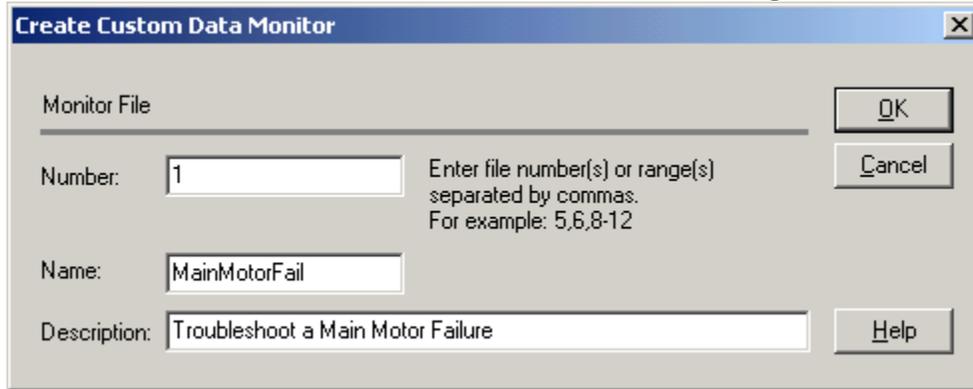


The main motor will only start if the main switch is on, the lube alarm is not on, and then we must wait 10 seconds. Let's build a CDM so we don't have to troubleshoot using ladder logic.

1) In the project tree, right click on the custom data monitor folder, and select 'New'.



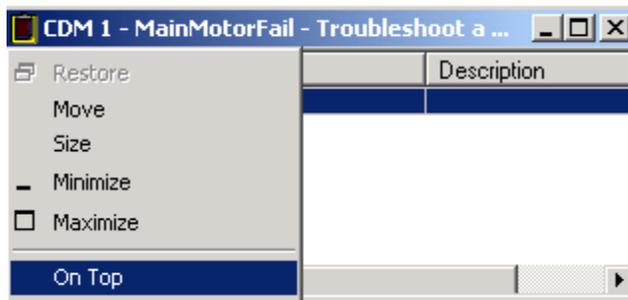
2) Let's name and describe the data monitor as shown below, then press OK.



By using descriptors, you are helping the troubleshooter determine the purpose of the data monitor when he hovers his mouse over the CDM in the project tree. You can create up to 255 data monitors, so descriptions can become very useful.

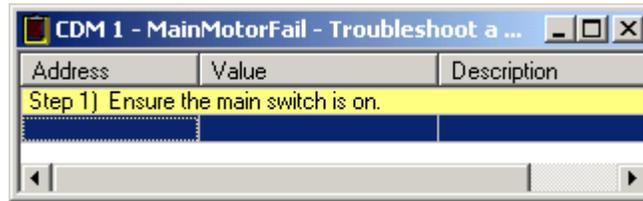


4) Next, double click your new custom data monitor. This will open the CDM, but if a user clicks on the logic window, part of the logic window will cover up the CDM. Usually, we want the CDM to be on top of other windows. To put your data monitor on top of all other windows, click the clipboard in the upper left corner of the CDM, and choose 'On Top'.

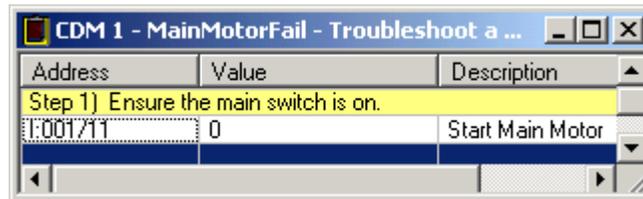


We can now set up the CDM!

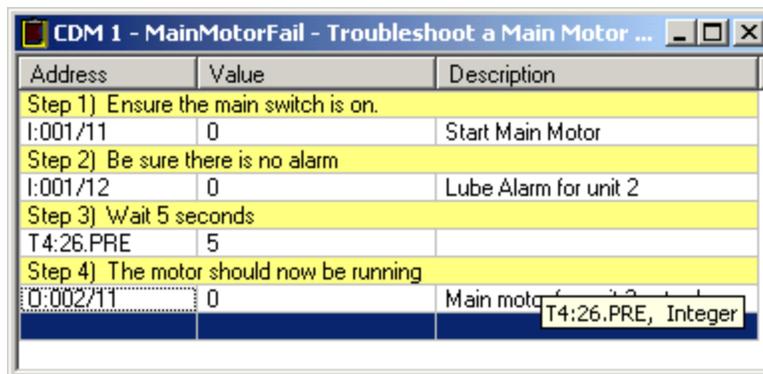
- 5) First, we will document the first step in troubleshooting. Highlight the first line of the CDM (so it's blue), and then begin to type your text as shown below:



- 6) Next, we will give the user an address so the value can be monitored. This address can be typed manually, or you can drag this address from the ladder. If you prefer the drag and drop method, be sure not to drop the address on top of your text. The address can be dropped anywhere in the bottom of the CDM (below all other entries), and it will assume the last available row.



- 7) Now the switch can be monitored in real time (as long as you are on line). Go ahead and set up the rest of the CDM as shown:



8) Notice the descriptions were brought in from the database. A user can also change descriptors within the CDM file, and those changes will be entered into the documentation database. Values can also be modified from the custom data monitor such as the timer's preset value. In the above example, just the preset is shown. If you were to drag the entire timer into the CDM, you would see a + next to the timer indicating the timer can be expanded to see the rest of the components. Look below:

Address	Value	Description
Step 1) Ensure the main switch is on.		
I:001/11	0	Start Main Motor
Step 2) Be sure there is no alarm		
I:001/12	0	Lube Alarm for unit 2
Step 3) Wait 5 seconds		
T4:26	(...)	Main Motor Delay Timer
- DN	0	
- TT	0	
- EN	0	
- PRE	5	
- ACC	0	
Step 4) The motor should now be running		
O:002/11	0	Main motor for unit 2 extruder

9) Addresses such as I:001/11 O:002/11 really don't mean a lot to us. The CDM can be set up to show symbols instead of addresses. To change this setting, right click on the CDM, and select 'Show Symbols'.

Address	Value	Description
Step 1) Ensure the main switch is on.		
I:001/11	0	Start Main Motor
Step 2) Be sure there is no alarm		
I:001/12	0	Lube Alarm for unit 2
Step 3) Wait 5 seconds		
T4:26	(...)	Main Motor Delay Timer
- DN	0	
- TT	0	
- EN	0	
- PRE	5	
- ACC	0	
Step 4) The motor should now be running		
O:002/11	0	Main motor for unit 2 extruder

10) Now that you are viewing the symbols, you can move your mouse over any of the symbols, and a tool tip will display the address for the symbol. To go back to address mode, right click on the CDM, and choose 'Show Addresses'.

Address	Value	Description
Step 1) Ensure the main switch is on.		
MAIN_SWITCH	0	Start Main Motor
Step 2) Be sure there is no alarm.		
LUBE_AI:001/11 -> MAIN_SWITCH, Bool	0	Lube Alarm for unit 2
Step 3) Start Main Motor		
MAIN_DELAY {...}	0	Main Motor Delay Timer
- DN	0	
- TT	0	
- EN	0	
- PRE	5	
- ACC	0	
Step 4) The motor should now be running		
MAIN_MOTOR	0	Main motor for unit 2 extruder

### On your own!

In RSLogix, write logic similar to the example shown above. Be sure to use descriptions and symbols to document your project. Create your own CDM for a user to quickly find a problem when the output is not on.

## Setting up a Trend Chart

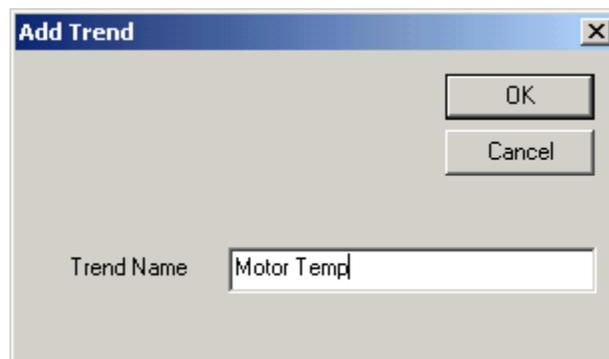
The trend chart allows you to track data over time, as a software 'chart recorder'. The trending chart is good for analog signals, but will work with bits as well (although not as good as the histogram for tracking bit transitions).

In this example, we are going to track and address, N7:7 over time.

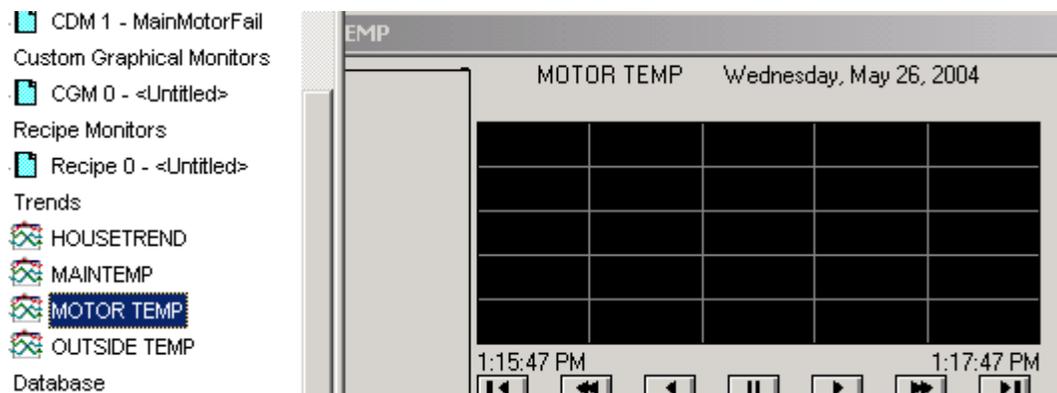
1) Right click on Trends in the project tree, and select 'New'.



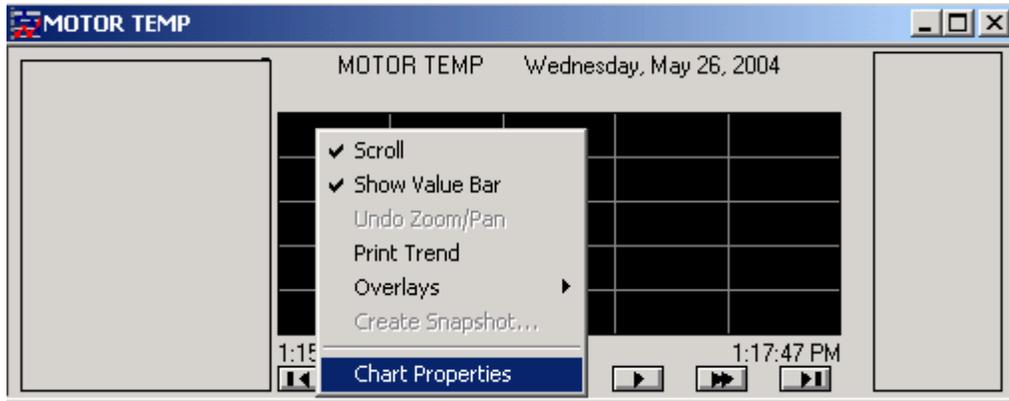
2) Next, you will name your trend. For this example, we will call it motor temp, then press OK.



3) Under the trends folder, double click to open the Motor Temp trend chart.



- 4) Next, a pen must be added to the chart. Right click on the chart to access the chart properties.



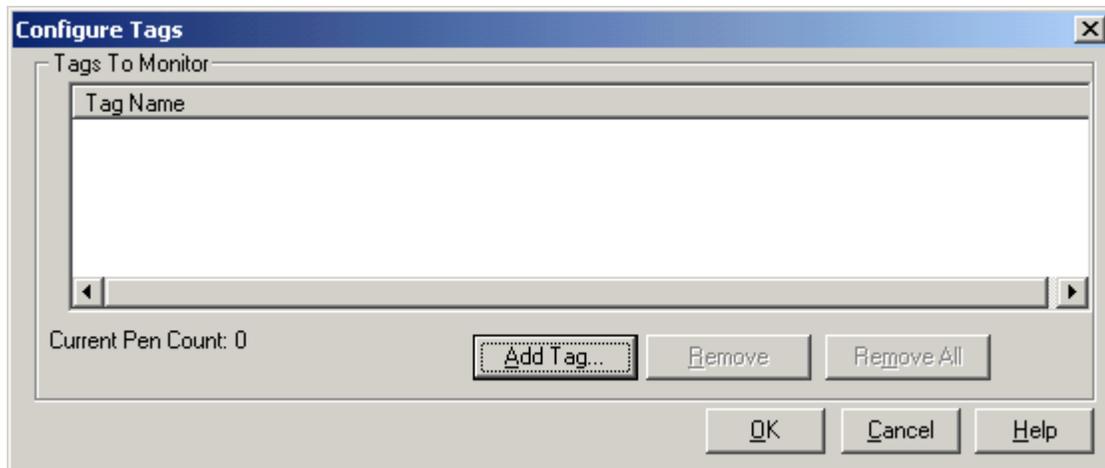
- 5) Click on the 'Pens' tab in the properties window.



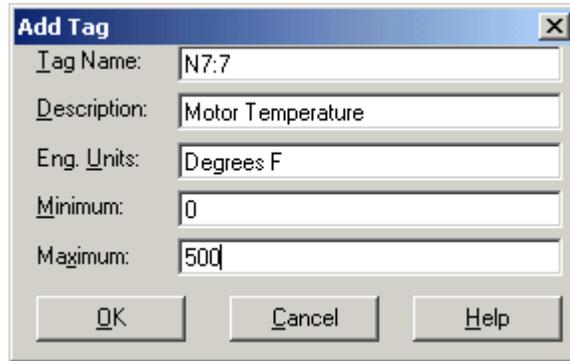
- 6) Click the Add/Configure Tags button (in the middle of the properties window).



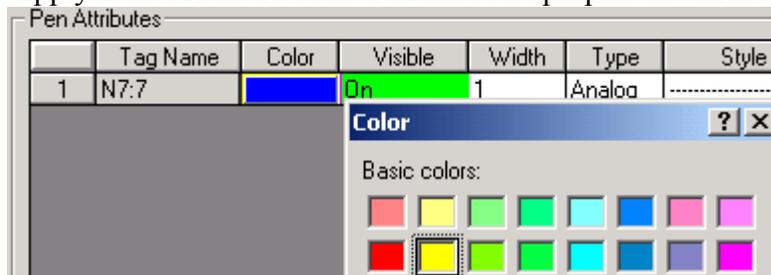
- 7) Next, the 'Configure Tags' Dialog screen appears. Click 'Add Tag'.



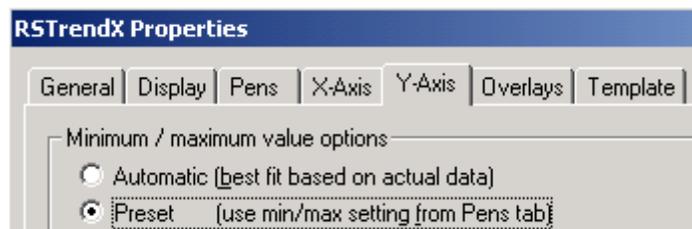
- 8) The 'Add Tag' Window will be set up as follows for this example. The Tag Name is the actual address you need to track. Also set up the Minimum and Maximum values you would expect at this address. Then we will press OK when finished, then press OK on the Configure Tags window.



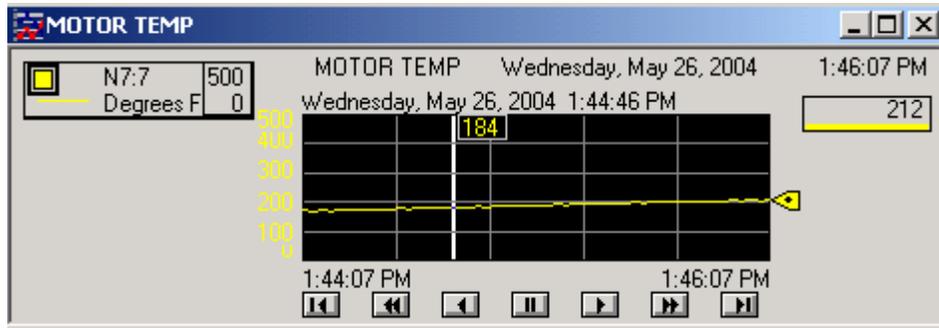
- 9) Next, set the color for your pen. The chart has a black background by default, so we can choose a bright color. To set the color, double click the color box in the pen attributes frame. A pallet will appear. Select a new color, then press OK on the pallet. Next, press Apply then OK at the bottom of the chart properties window.



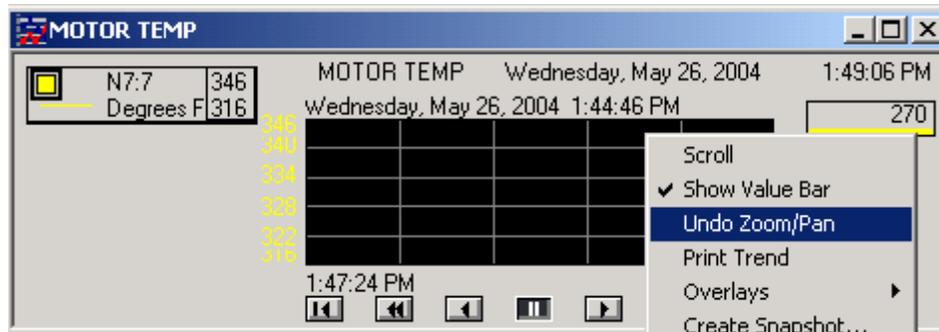
- 10)The chart is now tracking, but the first thing we notice is what appears to be wild fluctuations in temperature. If we look at the scale, however, the temperature is only fluctuating a few degrees. By default, the chart is in automatic mode, so the scale will automatically adjust to the lowest and highest value it sees. To change this scale to the preset values we entered when setting up the tag, right click the chart, and choose chart properties. On the Y Axis tab, and choose 'Preset'. Apply your changes, then press OK.



11) You can now see the pen is tracking your data. The motor temperature is slowly increasing over time. To see what the motor temperature was at any point along the graph, click one time on the chart, and a value bar will appear indicating what the value of the pen was at that moment.



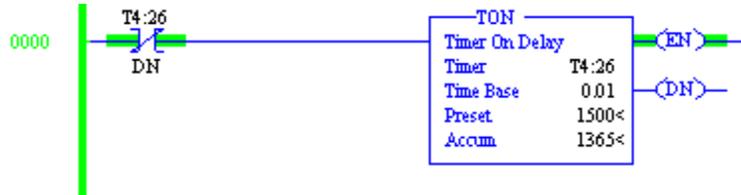
12) Be careful. If you accidentally move your mouse while clicking, you will draw a box around a small portion of the chart. When you let up on your mouse button, the chart will zoom in on the area where you clicked. You will notice this by a magnifying glass on the mouse pointer. To get out of this mode, right click on the chart, and Undo Zoom/Pan. Then press the || (pause/scroll) button at the bottom of the chart to begin tracking again.



13) By default the amount of time shown from left to right is 2 minutes. If you wish to change this value, right click on the chart, and choose chart properties. Then click on the X axis tab. The time span can then be modified. When finished, apply your changes, then press OK.

## On Your Own!

Create a trend chart based on any analog signal in your program. If you don't have an analog signal to track, create a self running timer as in the example shown below:



The Tag Name for this example would be T4:26.ACC (The accumulated value of timer 26).

Explore the other tabs in the trending chart. There are many other features that can be customized. Once you get your pen set up, try to change the color of the background under chart properties.

## *Custom Help Screen*

For each application, one customized help screen can be created. This help screen can be in many formats including text, html, and word document format.

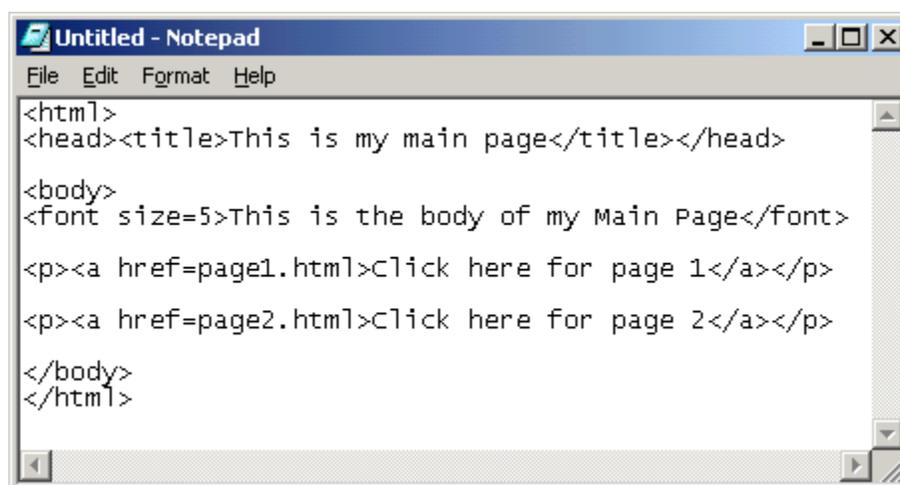
Since there is only one custom help screen available per project, this example will use the HTML format. Using HTML, we can create a local web page with links to other documents. This makes this feature very useful.

For this example, we will create 3 web pages. A Main page called main.html, and two other pages called page1.html, and page2.html. The user will call up the main page from RSLogix, and from the main page, the user can bring up page 1 or page 2.

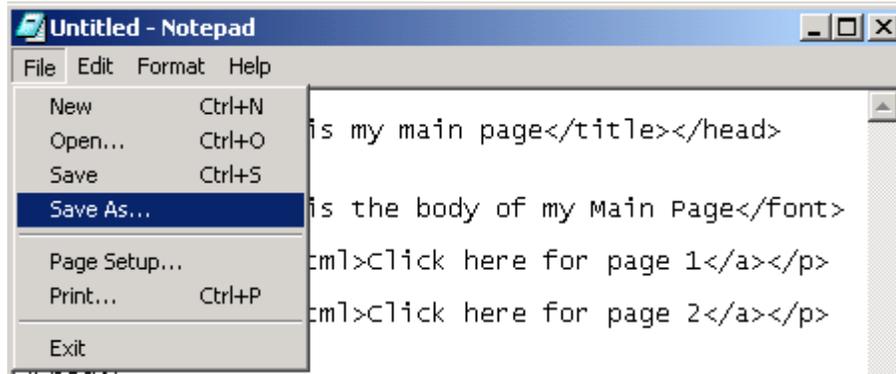
- 1) The first step is to bring up a standard text editor such as windows notepad. This is under Start | Programs | Accessories.
- 2) Now, you can copy and paste the following code into windows notepad:

```
<html>
<head><title>This is my main page</title></head>
<body>
<font size=5>This is the body of my Main Page</font>
<p><a href=page1.html>Click here for page 1</a></p>
<p><a href=page2.html>Click here for page 2</a></p>
</body>
</html>
```

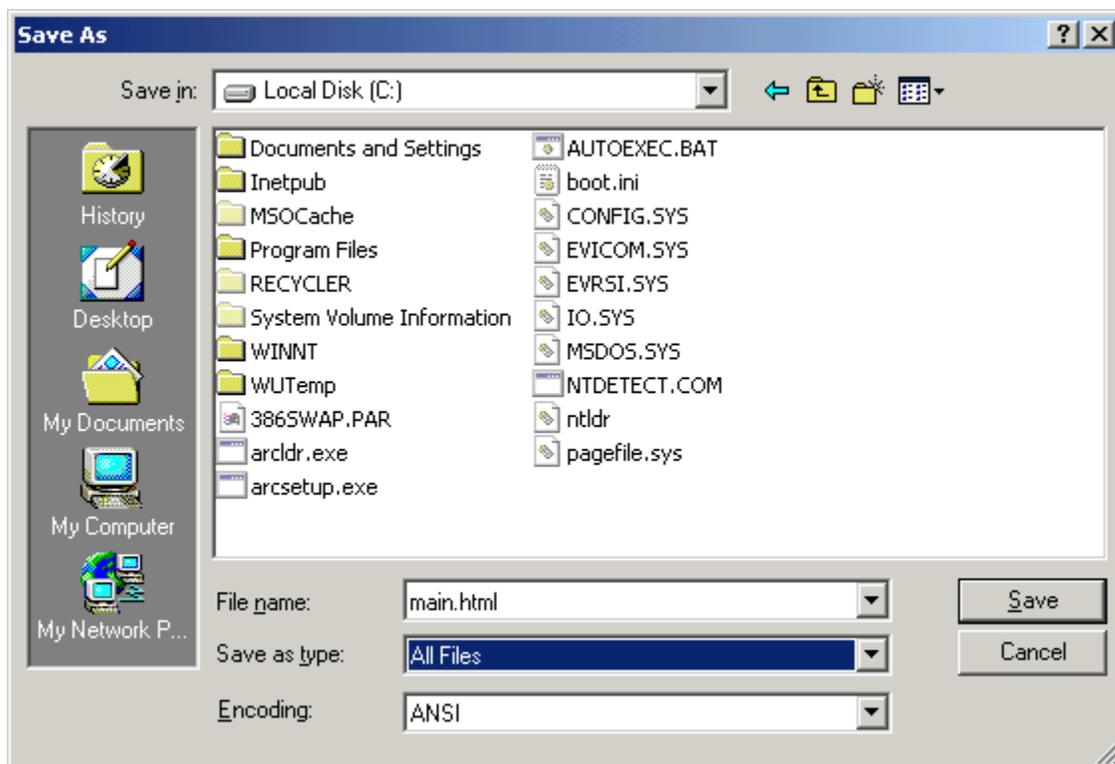
Your page will look like this:



3) Next click File | Save As from the menu bar in notepad.



4) The file will be called main.html on the main C: drive as shown. Be sure to set the file type as All Files, then save your work.

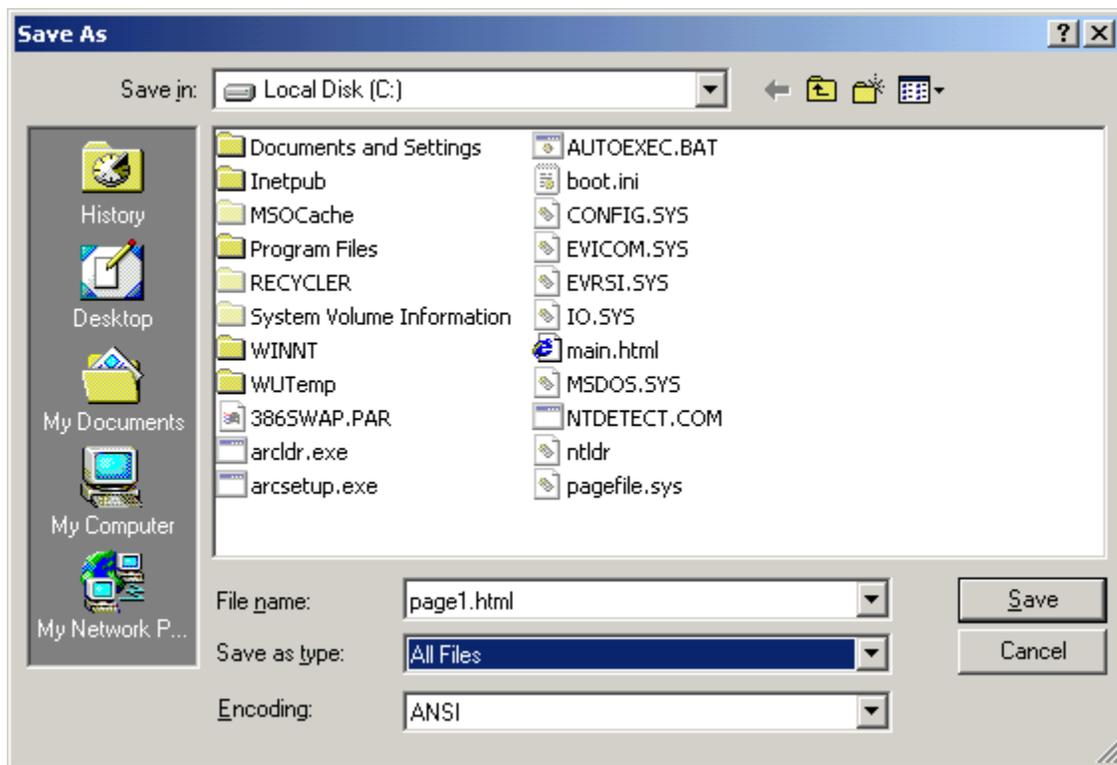


5) A similar procedure will be followed for Page 1 and Page 2. Click File | New in notepad to create a new document.

6) Paste the following code into notepad:

```
<html>
<head><title>This is Page 1</title></head>
<body>
<font size=5>This is the body of Page 1</font>
<p><a href=main.html>Click here to go back to main</a></p>
<p><a href=page2.html>Click here for page 2</a></p>
</body>
</html>
```

7) Click File | Save As, to save page1.html to the C: drive. Again be sure to select files off All Types as shown below, and then click the save button.

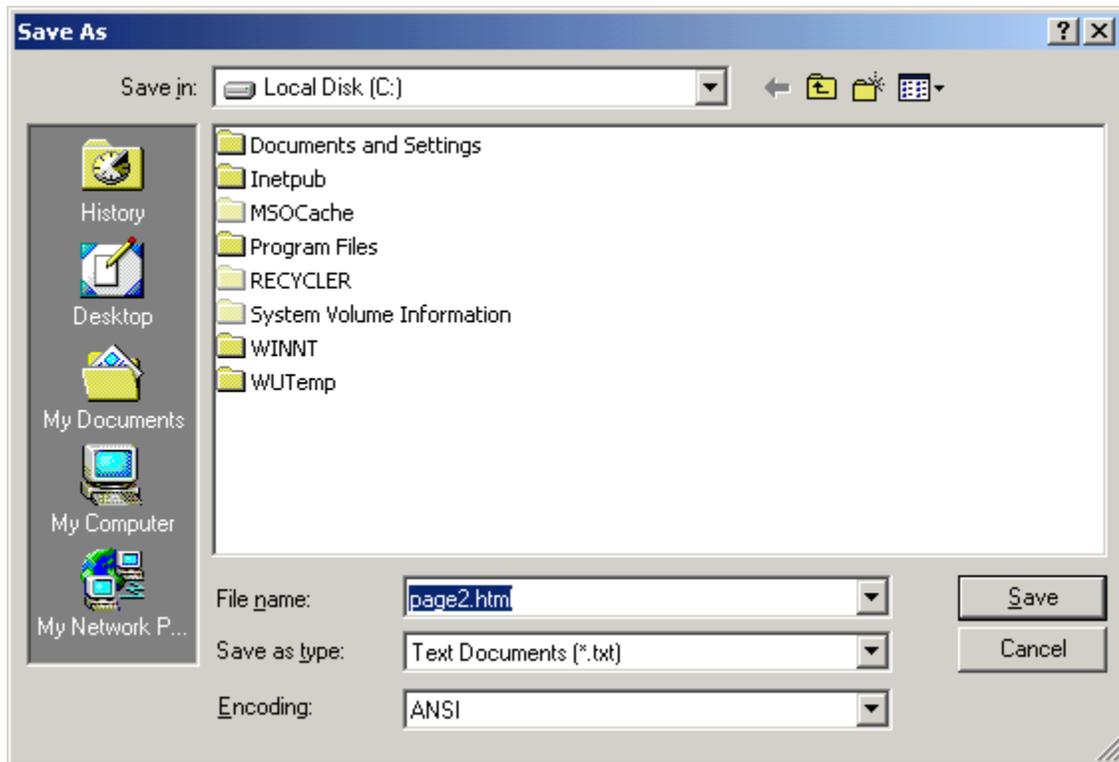


8) Again, click File | New in notepad

9) Paste the following code into notepad:

```
<html>
<head><title>This is Page 2</title></head>
<body>
<font size=5>This is the body of Page 2</font>
<p><a href=main.html>Click here to go back to main</a></p>
<p><a href=page1.html>Click here for page 1</a></p>
</body>
</html>
```

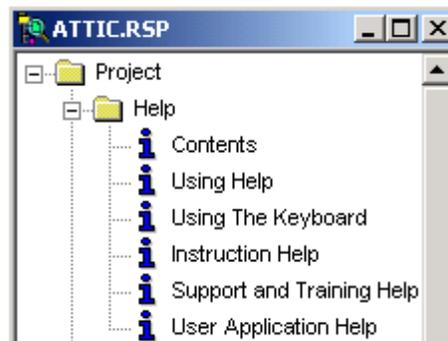
10) Click File | Save As, to save page2.html to the C: drive. Again be sure to select files off All Types as shown below, and then click the save button.



11) You can now go back and modify the content of the pages, or even add pages so they fit the need of your application.

12) Next, RSLogix needs to be configured so the user help feature knows what file to look for. Open RSLogix, and then open the project (.rsp or .rss file) that you wish to link this help information to.

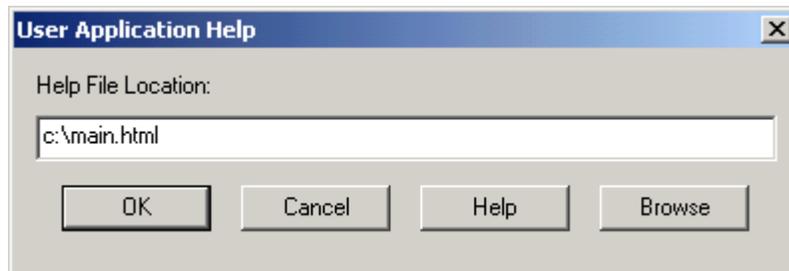
13) In the project tree, expand the help folder as shown:



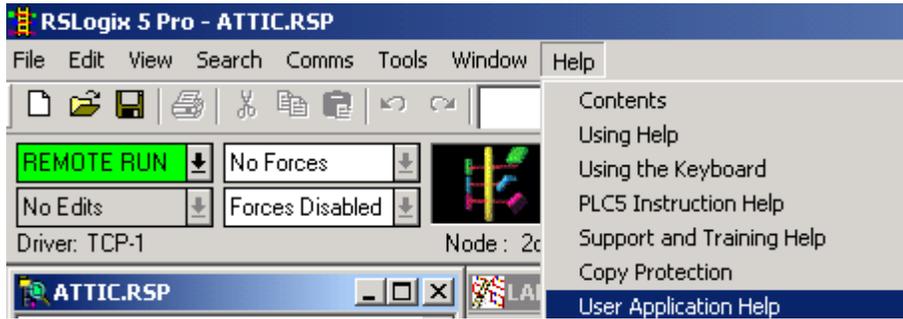
14) Right click on User Application Help.



15) You are then asked for the location of the help file. This will be [c:\main.html](#). You can also browse for the help file. Click OK when the help file location is correct.



16) Now you can select user application help either from the project tree, or from the help menu on the menu bar.



17) Your default browser is now launched, and navigation should work correctly.

This is the body of my Main Page

[Click here for page 1](#)

[Click here for page 2](#)

## ***Hide and Unhide Program and Data Files***

Program and Data files that are seldom accessed can be hidden from view. They can also become inadvertently hidden from view.

### **Hiding a Data or Program file:**

To hide a file from view, Right click the file, and choose *Hide*:

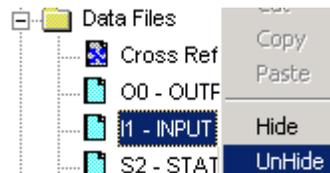


When a file is hidden it becomes invisible from view of the user:

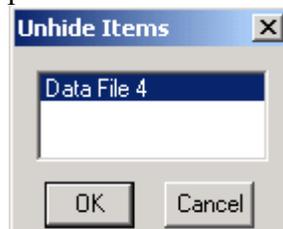


### **Unhiding a Data or Program File:**

To unhide a data or program file, right click on any data or program file (or data/program file folder), and choose *Unhide*.



A dialog box will appear containing a list of the hidden data and program files. Choose the files you wish to unhide, then press OK.



You will then see that your program or data files are again shown in the project tree.



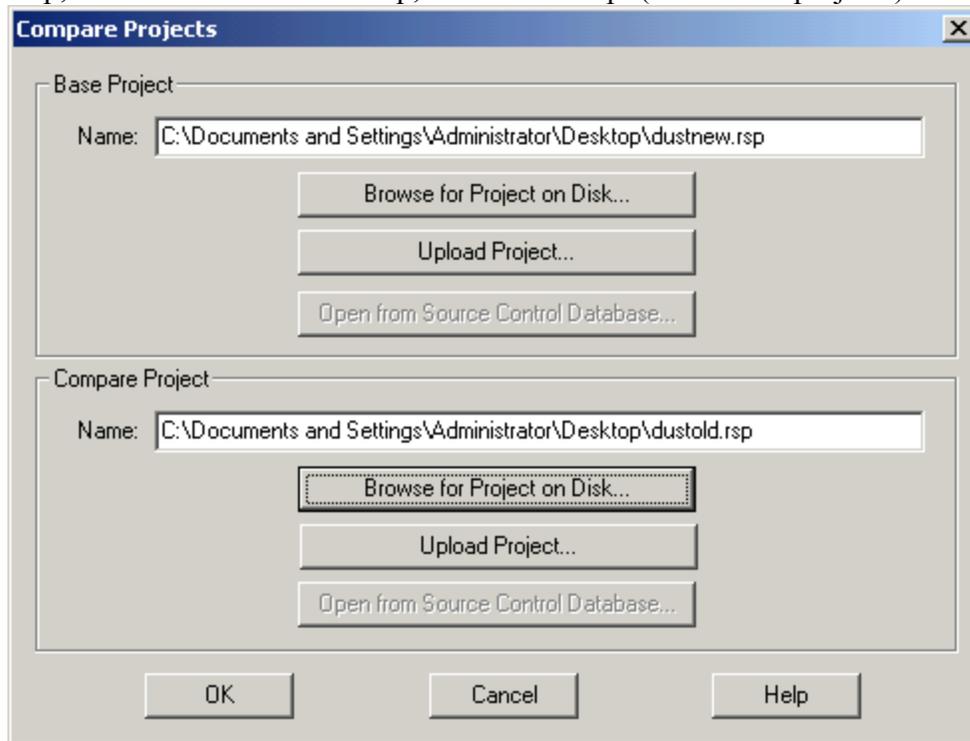
## Using the Project Compare Utility

Have you ever wondered if the project on your computer is the same as the project currently running in the PLC? Or have you ever wondered what the differences are between two PLC programs? The Project Compare Utility will help you determine the differences between 1) Two projects on the hard drive, 2) A project on the hard drive against a project in the PLC, or 3) Two projects currently running in different processors.

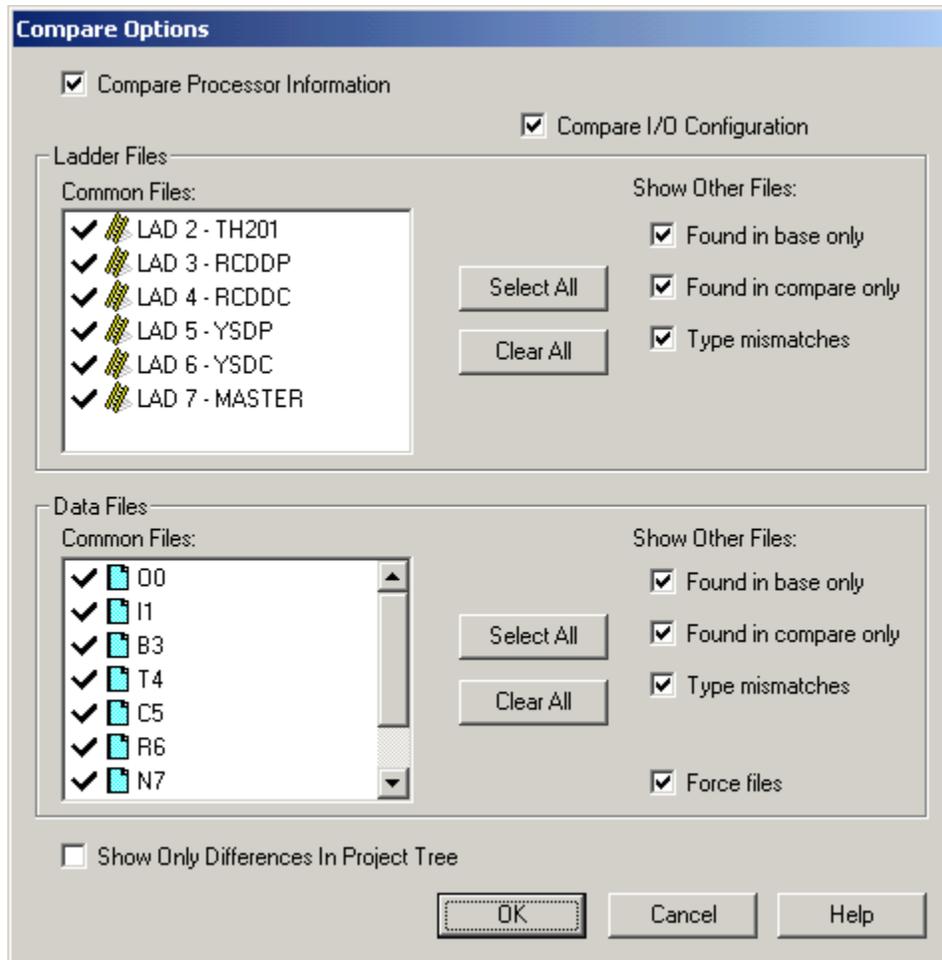
- 1) Click Tools | Compare from the menu bar to access the project compare utility. You do not need to have a project open, and any open projects will be closed.



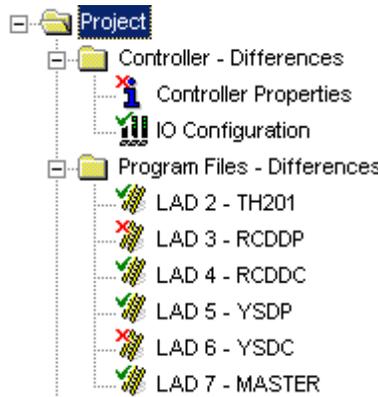
- 2) Next, you can browse for a project on the hard disk, or upload a project from the PLC for both the Base and Compare projects. For this example, we will compare two files on the hard drive against each other to find the differences. These two files are on the desktop, and are called dustnew.rsp, and dustold.rsp (browse for projects)



3) Next, the compare utility needs to know what you want to compare. For this example, we will leave this at default to compare everything. You can also check a box at the bottom of the Compare Options screen so only the differences are shown in the results. It is normal for data files to differ between projects, because the state of the I/O, Timers and Counters could be different depending on what part of the machine cycle the program was uploaded. On this screen just press OK.



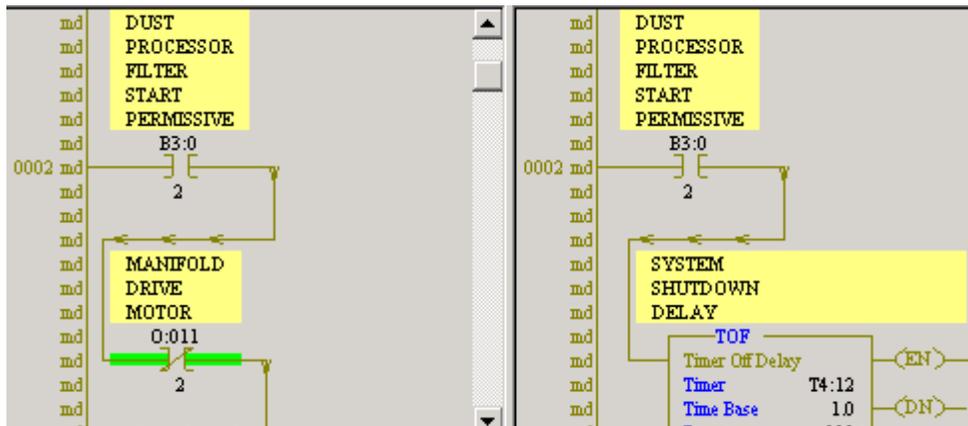
- 5) In the image below, you can clearly see what ladders have discrepancies between the two projects. Ladder 3 and ladder 6. There are also some differences in the controller properties. Let's double click ladder 3.



- 6) The ladders are then placed side by side. Click the blue down arrow to navigate to the first discrepancy.



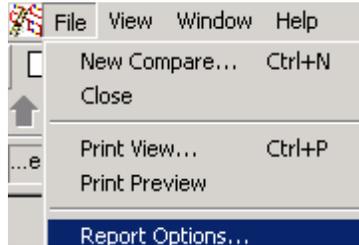
- 7) Look in the margin, and you will see the letters md. A legend at the top of the window tells us that md has been modified. Can you find the difference between the two rungs? An extra bit has been added to the new project.



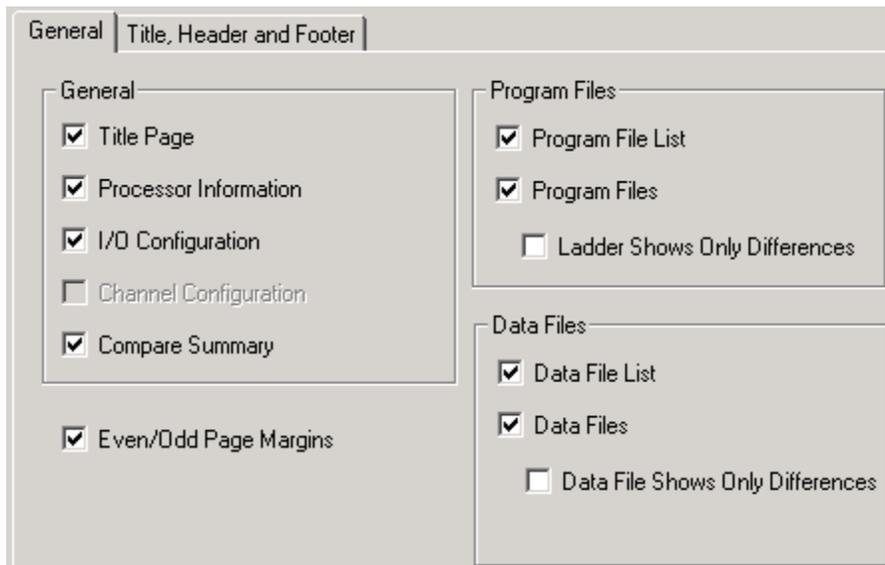
- 8) To navigate up or down through discrepancies between the two ladders, continue to click the blue up or down arrows. To look at another ladder, close out of the window that is open within the compare utility. You will be taken back to the project tree.

## Generating Compare Reports

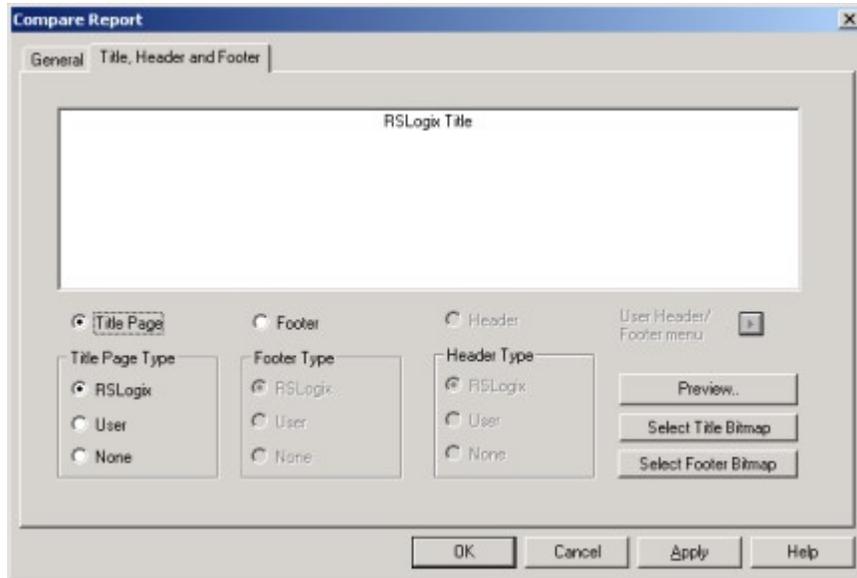
- 9) Compare reports can be generated in a similar way that project reports are generated. To access the report options, click File | Report options from the menu bar.



- 10) On the General tab, select the reports you would like to print.



11) Click on the Title, Header, and Footer tab. On this screen you can configure your text and bitmaps for the report. Remember, if you need to customize one of these options, be sure to select User mode.



12) Click the preview to see what your final report will look like.

## Printing Reports

Reports can be a useful troubleshooting tool for those who do not have access to RSLogix, or those who prefer to troubleshoot from a hard copy of the program. Before reports can be printed, the user must first tell RSLogix what reports are to be printed (such as ladder logic or data files). After the reports are selected, each report can be given special options (such as what ladders or rungs to print in a program file). Be aware that even the smallest project can use a lot of paper if all reports are selected.

*Note: If you are just wanting to print the current ladder, or a current data file, a report does not have to be configured. Just use the print view feature (File | Print View), or click the print ladder icon on the standard tool bar.*

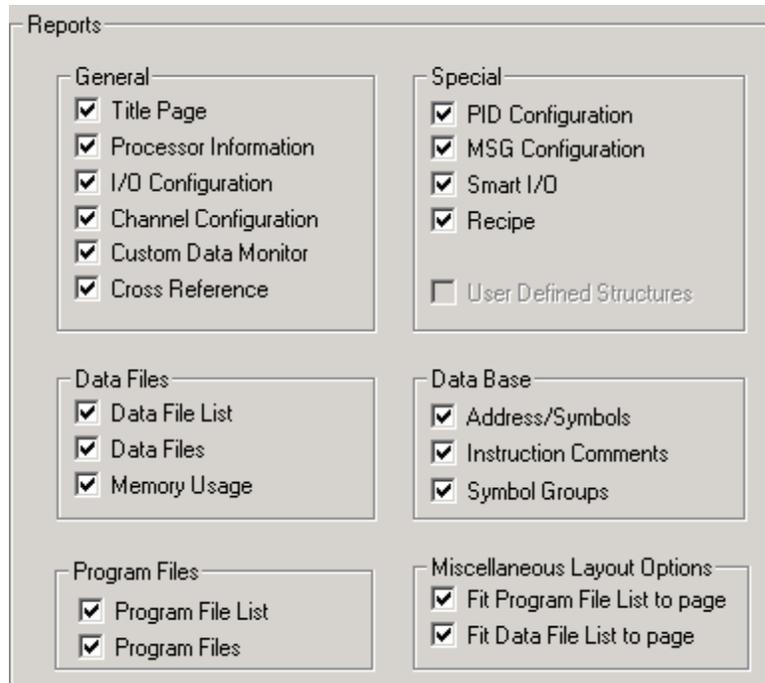


*Note: Reports may be configured while on line, but in order to print or preview a report, you must be off line so data table values are not changing.*

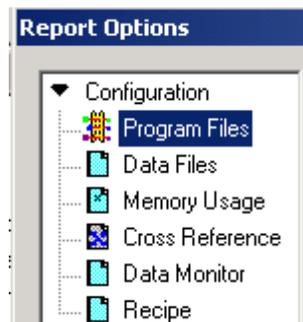
1) Click File | Report options from the menu bar.



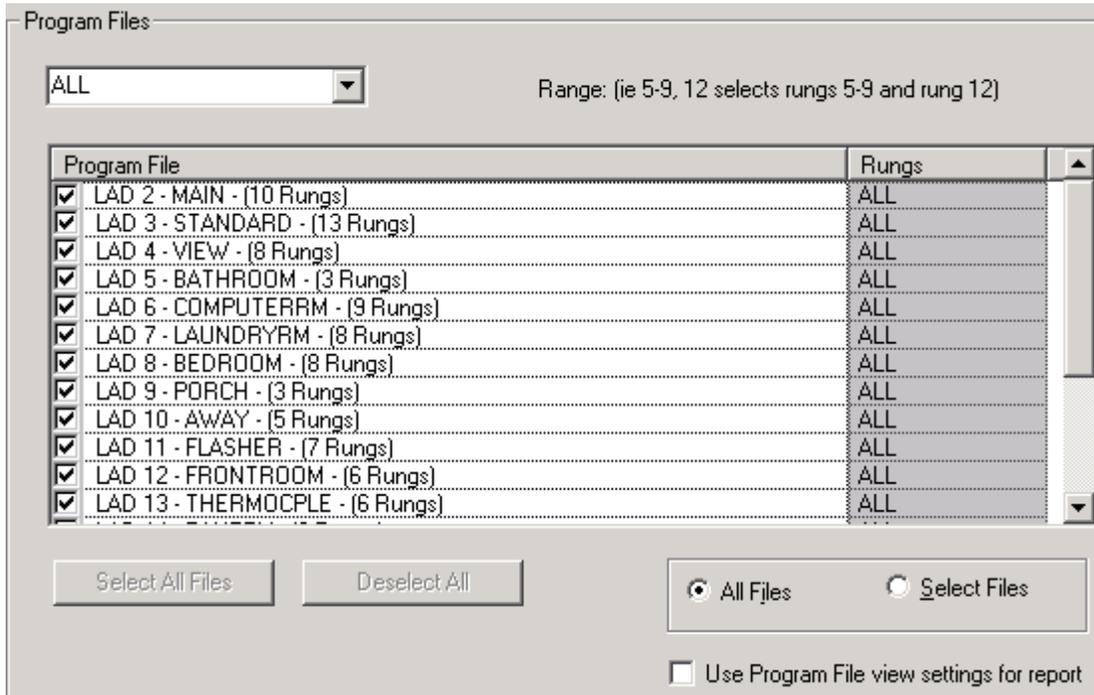
- 2) Next, select what reports you would like to print. For this example, select all reports.  
(We will just preview the reports instead of actually printing them.)



- 3) Now that all reports have been selected, let's look at the options that are available for each individual report. On the left side of report options, you will find a configuration tree. Click on Program files to see what options are available.



4) By default, all rungs on all ladders will be printed. If you wish to select only certain programs,



If you wish to print only certain ladders, you must choose the 'Select Files' button.



You can then select which ladders you wish to print. In the example below, Ladders 3 and 4 will be printed, and rungs 1,2,3,5,6, and 7 will be printed of Ladder 6.

Program File	Rungs
<input type="checkbox"/> LAD 2 - MAIN - (10 Rungs)	ALL
<input checked="" type="checkbox"/> LAD 3 - STANDARD - (13 Rungs)	ALL
<input checked="" type="checkbox"/> LAD 4 - VIEW - (8 Rungs)	ALL
<input type="checkbox"/> LAD 5 - BATHROOM - (3 Rungs)	ALL
<input checked="" type="checkbox"/> LAD 6 - COMPUTERRM - (9 Rungs)	1-3,5-7

You also have the ability to use the program's view settings for the report (found under View | properties). This feature allows you to change the way the ladder is displayed on your screen such as comments, colors, fonts, and bit display mode. Check this box if you want the printout to reflect your view settings.

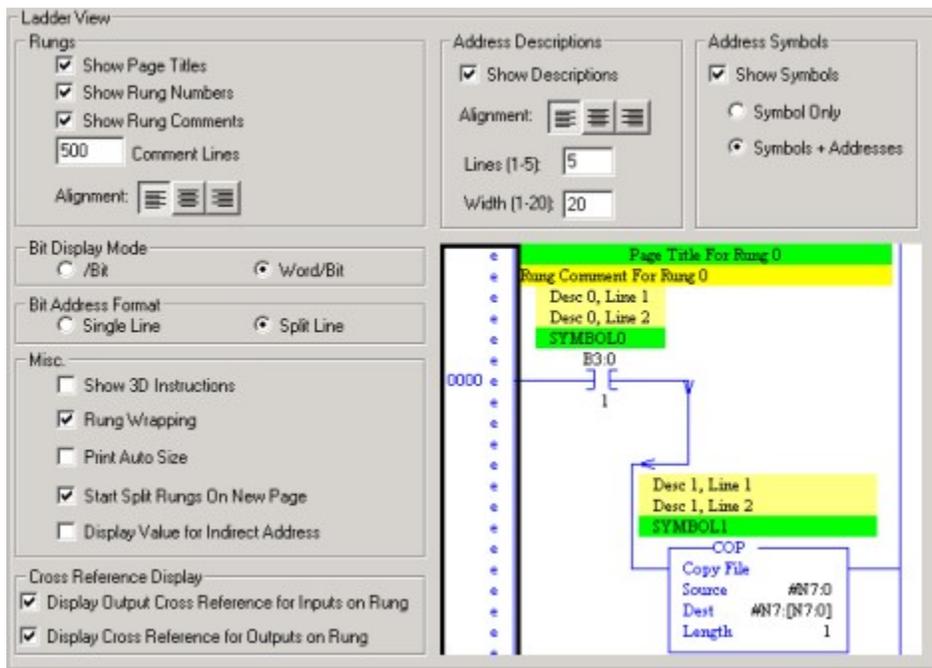


If this box is not checked, the report settings can be different than the view settings. If the colors in your view settings will not print well on a black and white printer, it might be a good idea to configure the print settings differently.

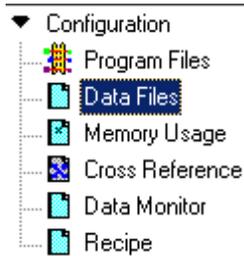
To change the ladder print settings, click 'Ladder Setup' in the configuration tree.



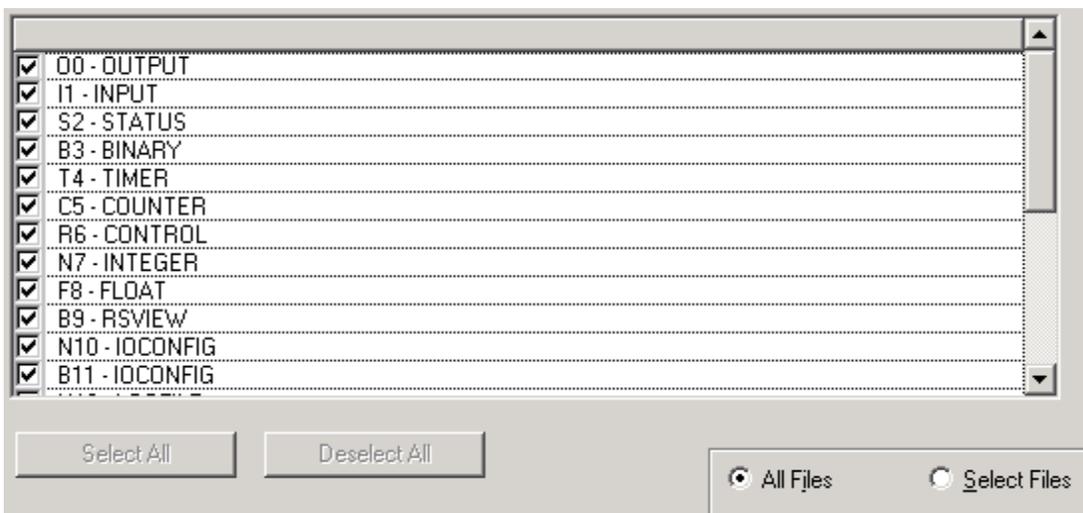
Now, configure how you want the ladder to print. The colors and fonts can be changed as well. Remember, this screen is only used if you are not using the view settings for the report.



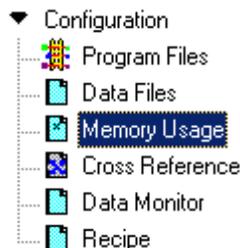
5) Next, Click 'Data Files' under Configuration.



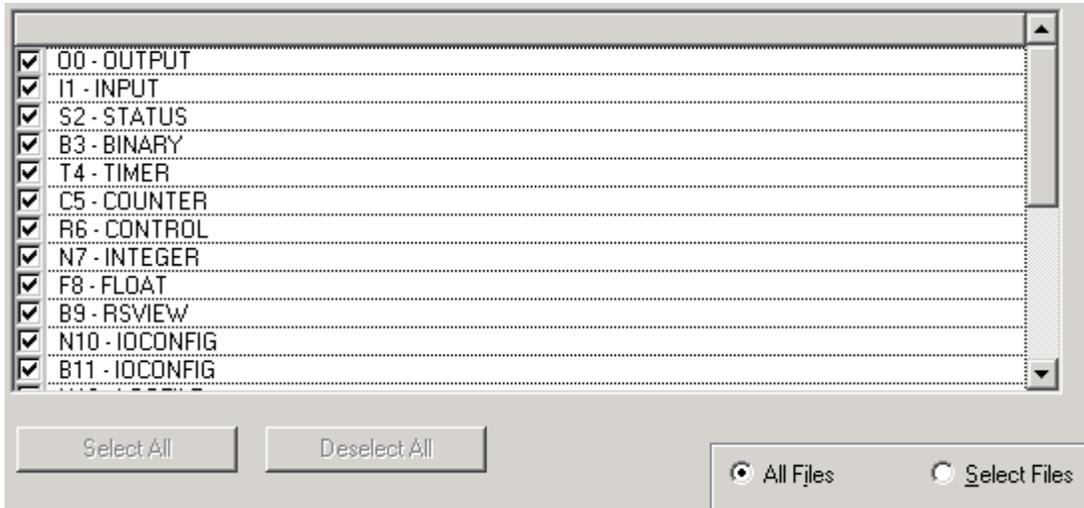
6) Configuration of the data files report is very similar to the program files. By default, all data files are printed. If you want only certain files to print, choose 'select files', and put a check mark next to the files you wish to print.



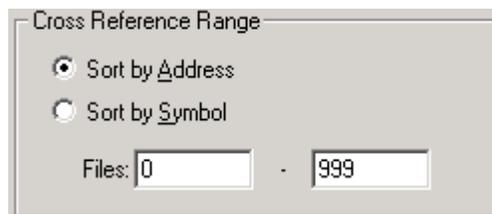
7) Next is the memory usage report. Occasionally, an RSLogix user will inadvertently specify a timer such as T4:999, when he actually meant to specify T4:99. RSLogix will automatically expand the T4 file to include a thousand timers. This is a waste of memory. The memory usage report will you locate files which are unnecessarily using a lot of memory.



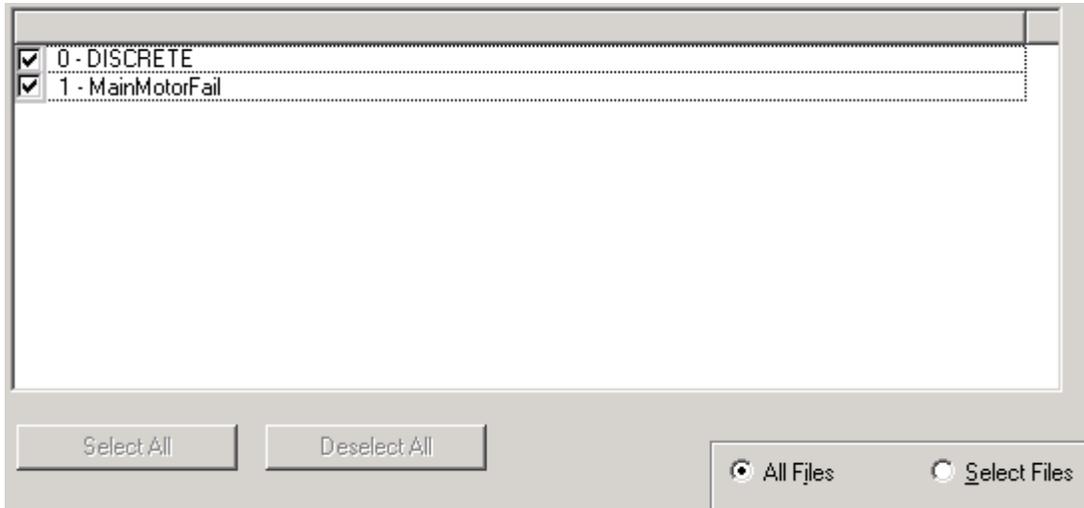
Configuration of the memory usage report is very similar to that of the data file report. By default, all files are selected. To select only certain files, choose 'select files', and put a checkmark next to the files you want included in the report.



8) For the cross-reference report, you can sort by symbols, or sort by addresses. If you aren't using many symbols in your project, you may want to leave this setting at default (sort by addresses). The cross reference report will help the user navigate through logic by showing every location in logic each address is used.



- 9) Custom Data Monitors can be used as troubleshooting tools when on line with the processor. Although a printout of a custom data monitor would have limited use, you do have the option to include CDM files in your report.



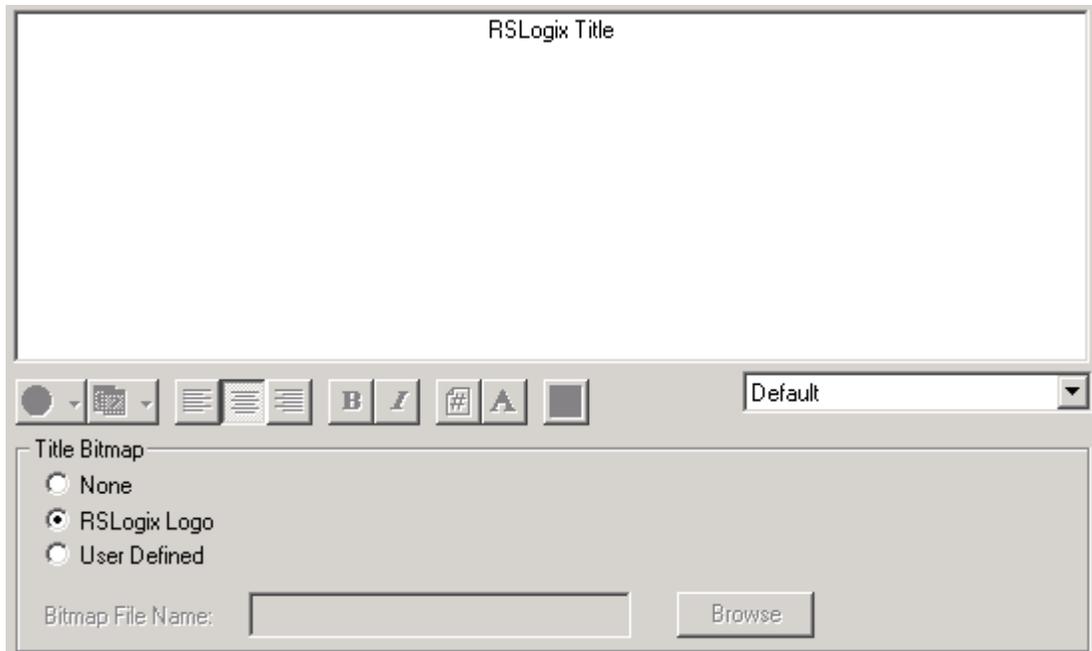
## Customizing the report

If you plan to keep the report, you want it to look professional. You can configure your own title page, and a header and footer for every page. Your report can contain your own text, and your own bitmap.

Since the configuration of all three options (title, header and footer) are the same, we will just take a look at the report's title page.

Double click 'Title' under Layout on the left side of your page.

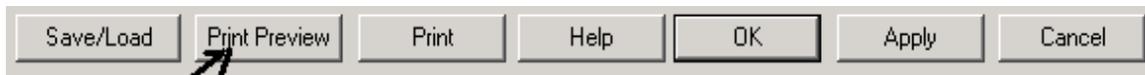




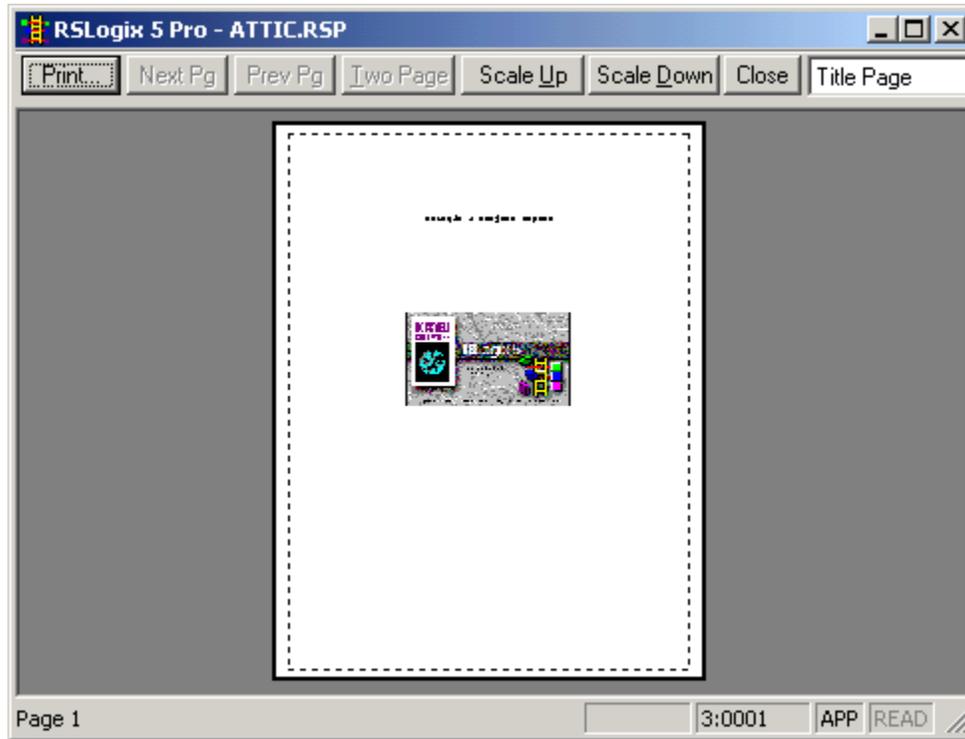
In the above example, you will notice that you don't have the ability to change the text from RSLogix Title to your own title. If you want to customize the text, you must hit the pull down tab, and change 'default' to 'user'. You can also define your own bitmap, and browse to the file on your system.

## Previewing your report

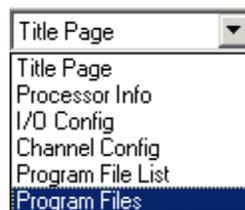
Since a report this large can use a lot of paper, let's do a preview to make sure the report is going to print how we would expect. Click the Print Preview button.



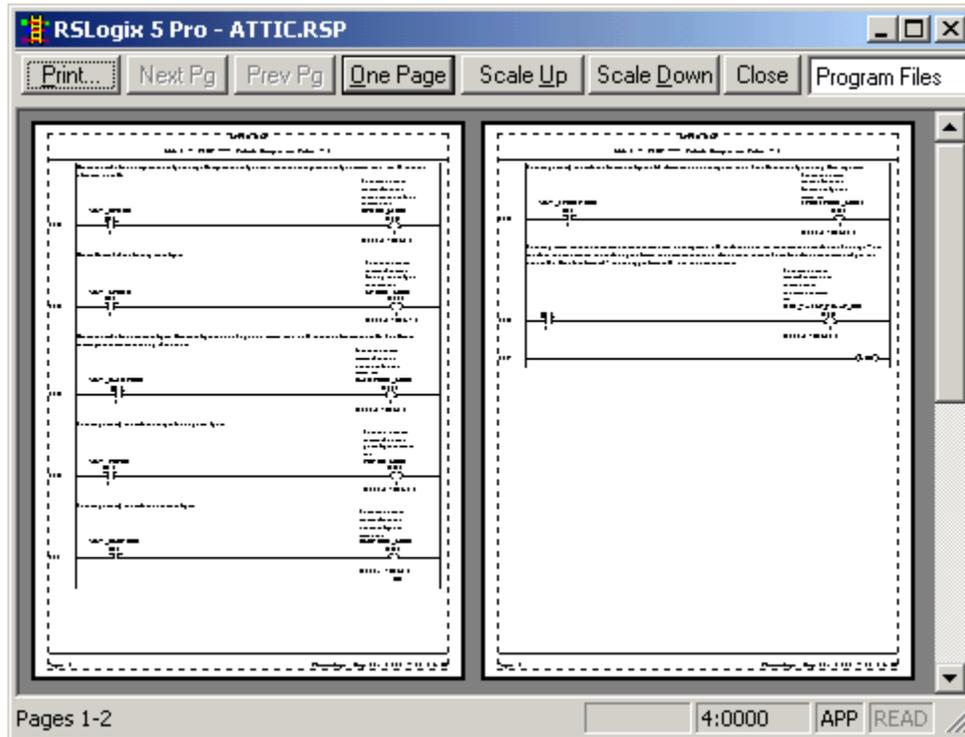
The report is generated. Notice that when you move your mouse over the preview, your mouse pointer becomes a magnifying glass. Click on the report to zoom in. Click again... And again... Notice there are 3 different zoom levels. The last time you clicked the report went back to standard size.



We are currently looking at the title page. From the pull down menu, choose 'Program Files'.



You see the JSR's in ladder 2. Click the 'Next File' to get to ladder 4. Remember, we didn't include ladder 3 in the report. While within ladder 4, you can click next page and previous page to view the entire ladder, or you can press the 'Two Page' button to view both pages at the same time. You must be in the lowest zoom level for this to be an option.



The scale up and scale down buttons make the content of each page larger or smaller (not the page itself). Take a look at some of the other pages you generated. See if the preview is what you expected when the report was being set up.

## On Your Own!

Open a project that is currently in use at your location. Configure the reports you would like to see printed, and run a print preview.

## GNU Free Documentation License

### GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

#### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous

- versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
  - **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
  - **D.** Preserve all the copyright notices of the Document.
  - **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
  - **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
  - **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
  - **H.** Include an unaltered copy of this License.
  - **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
  - **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
  - **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
  - **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
  - **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
  - **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
  - **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## **10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

### How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover  
Texts. A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.