Modbus TCP Master Driver for DeltaV Virtual I/O Module

for both Simplex and Redundant Modbus TCP Versions 3.6.x

USER MANUAL

April 2007

Disclaimers

While this information is presented in good faith and believed to be accurate, Mynah Technologies does not guarantee satisfactory results from reliance upon such information. Nothing contained herein is to be construed as a warranty or guarantee, express or implied, regarding the performance, merchantability, fitness or any other matter with respect to the products, nor as a recommendation to use any product or process in conflict with any patent. Mynah Technologies reserves the right, without notice, to alter or improve the designs or specifications of the products described herein. All sales are governed by Mynah Technologies' terms and conditions, which are available on request.

Table of Contents

Table of Figures and Tables

1.0 Introduction

1.1 Scope

This document is the User Manual for the Virtual I/O Module (VIM) with the ModbusTCP master driver firmware for the Emerson Process Management (EPM) DeltaV Control System. It provides the information required to install, configure, and maintain the driver firmware on the VIM. The reader should be familiar with EPM's DeltaV Programmable Serial Interface Cards (PSIC), Modbus protocol, and connected field devices (supporting the ModbusTCP protocol).

The section *Document Format* briefly describes the contents of each section of this manual. System Specifications outlines hardware and software requirements for the ModbusTCP Driver firmware.

1.2 Document Format

This document is organized as follows:

Introduction	Describes the scope and purpose of this document.
Theory of Operation	Provides a general functional overview of the Modbus TCP Driver.
Firmware Flash Upgrade	Describes procedures to upgrade the Modbus TCP driver firmware in the VIM.
DeltaV serial card Configuration	Describes procedures and guidelines for configuring the DeltaV serial cards residing in the VIM.
VIM network configuration	Describes Modbus TCP network device configuration.
Operational Check	Provides tips and assistance to ensure the VIM is properly setup and configured.
Technical Support	Describes who to call if you need assistance.

1.3 System Specifications

The following table lists the minimum system requirements for the Modbus TCP Driver:

Firmware	Modbus TCP Driver Firmware
VIMNet Utility	Windows PC resident VIMNet Plug and Play Server Utility.
Protocol Compatibility	ModbusTCP protocol conforming to “MODBUS Messaging on TCP/IP, Implementation Guide”, Rev 1.0, May 2002. This document is available from MODBUS.ORG
Software Requirements	DeltaV System Software (Release 6.3 or later) installed on a hardware-appropriate Windows workstation configured as a ProfessionalPlus for DeltaV Serial Interface Port License (VE4102). One license is required for each serial port used. The VIM has a maximum of 8 available serial ports. MYNAH VIM driver firmware IOD-4102
Minimum DeltaV Hardware Requirements	DeltaV M3, M5, M5+ or MD Controller 1 standard 2 wide controller carrier 1 standard Power Supply
VIM Hardware Requirements	MYNAH VIM part no. MIM-4207 For Simplex installation: 1 standard 2-wide controller carrier (Model Number VE3051C0) and 1 standard Power Supply (Model Number VE5008) For Redundant installation: 2 standard 2-wide controller carrier (Model Number VE3051C0) and 2 standard Power Supply (Model Number VE5008)
Network Hardware Requirements	Multiport 10/100BaseT Switch not shared with DeltaV Control Network. Two network switches are required for redundant communication.

Table 1: Modbus TCP Driver System Specifications

2.0 Theory of Operation

The DeltaV Virtual I/O Module (VIM) provides a native DeltaV I/O interface to open plant Ethernet networks and devices that use ModbusTCP (RTU TCP, RTU via TCP and RTU via UDP) protocol. DeltaV controllers can read and write signals from the plant floor devices that use these Ethernet networks such as PLCs, Motor Control Centers, and Weigh Scales. As such, the VIM is a Network Gateway between DeltaV controllers and field devices supporting network communications. This connectivity is illustrated below:

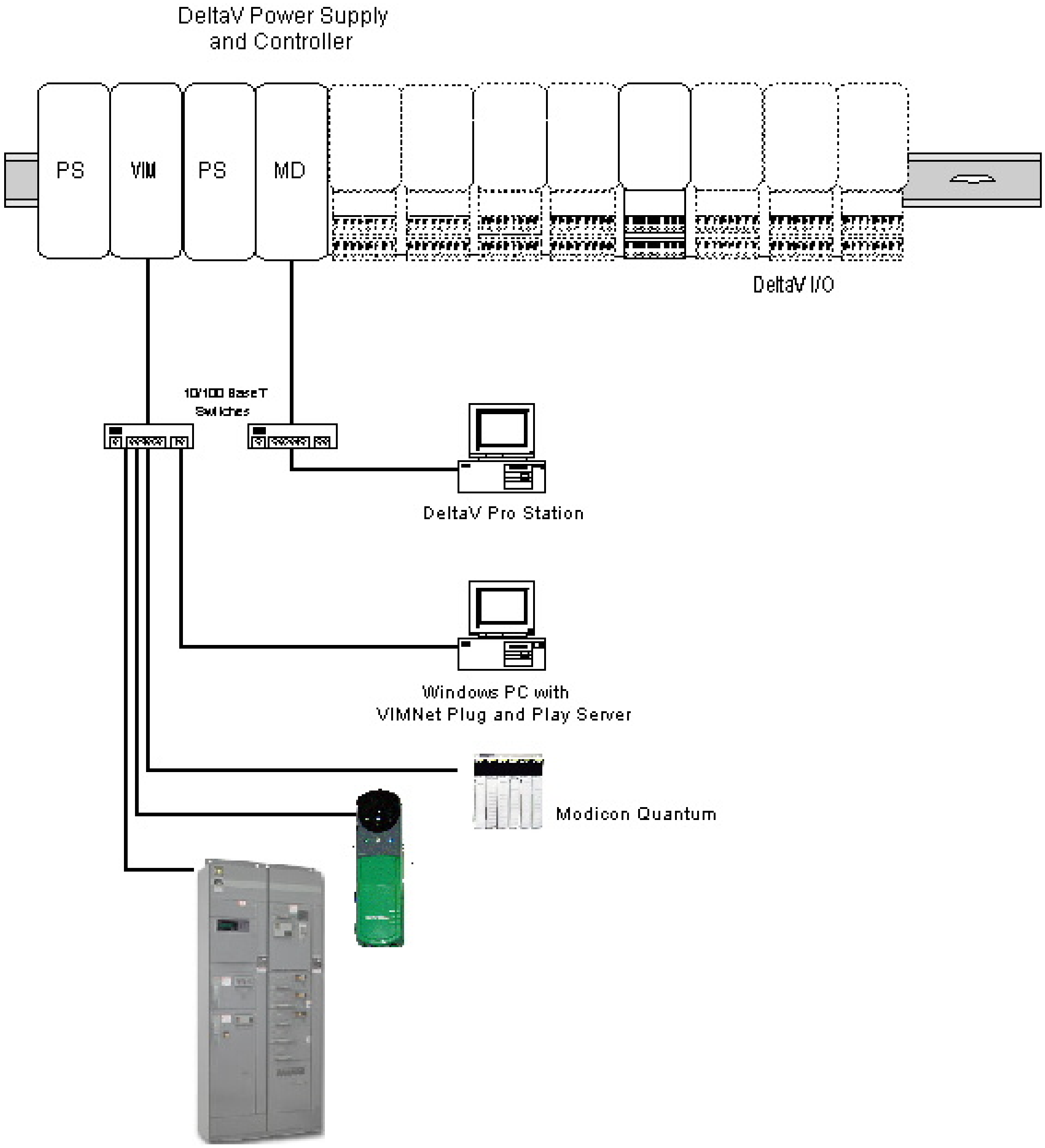


Figure 1: Simplex Modbus TCP Network

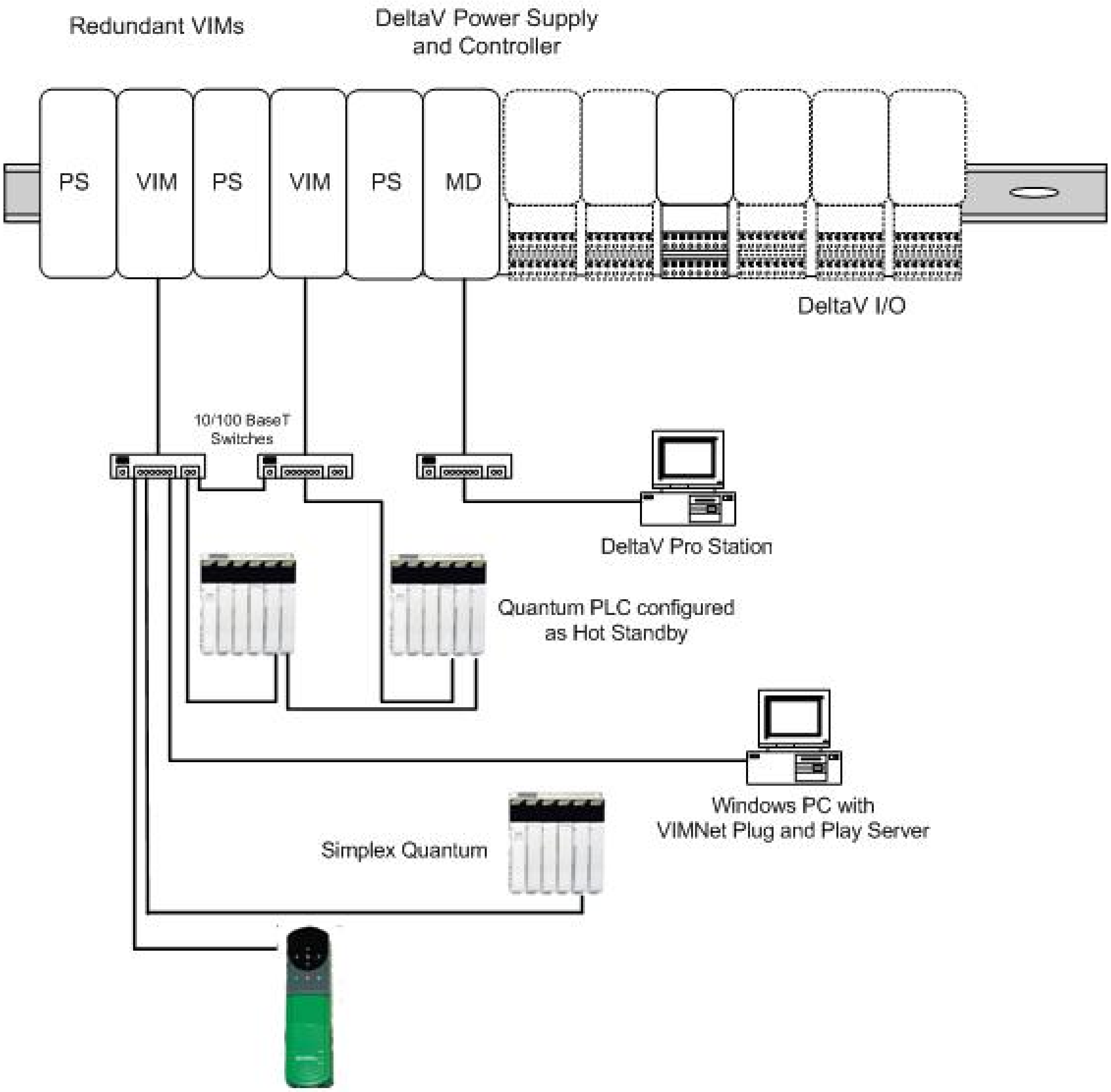


Figure 2: Redundant Modbus TCP Network

2.1 DeltaV Native I/O

The VIM provides a native DeltaV I/O interface by emulating four Programmable Serial Interface Cards (PSIC). By design, the VIM acquires the last 8-wide I/O carrier of a DeltaV system, emulating cards 57-60 or 61-64 as a single, simplex unit. Installing 2 simplex VIMs side-by-side provides emulation of all 8 serial I/O cards 57-64. The configuration of card group 57-60 or 61-64, and network properties of connected field devices is done in the VIMNet Plug and Play Server described in Section 3.

For redundancy support, the appropriate firmware (v 3.6.0 or later) must be flashed into the VIM. Four redundant PSICs are emulated when 2 VIMs are installed side-by-side and configured as a redundant pair. One VIM emulates all odd numbered serial cards, while the other VIM emulates all even numbered serial cards. The emulated serial cards behave as redundant pairs, i.e., 57/58, 59/60, etc. However, when redundancy switchover occurs, all cards behave as a bank and switch in unison. For example, if there is a communication error on card 57 that requires a switchover, the VIM will switch to its partner and cards 58, 60, 62 and 64 will become active.

The emulated serial cards appear to DeltaV as real serial I/O. The configuration of data tables to be read and written is done at the DeltaV Explorer level, in the same manner as required for a serial PLC device. This allows communications with any PLC or non-PLC device that supports the Modbus TCP messaging.

Each PSIC has 2 ports configured under it. There are 16 datasets under each port. Consequently, the VIM has the capacity of 128 datasets. One dataset is equivalent to 100 16-bit registers, or 50 floating point (32 bit) registers. These 128 datasets are user mapped to PLC devices as required for your application.

2.2 Modbus Devices

The Modbus device address is considered unique in the serial cards port domain. Specifically, within a serial port, all configured Modbus devices are unique. You can, however, configure the same device with the same address under another port. For a device address configured more than once under more than one port, the IP address always remains unique.

The VIMNet Server configuration correlates each unique Modbus address with an IP address. At the simplest level, each Modbus device equates to an IP address. In some cases, a single IP address may also be mapped to more than one Modbus device, as is typically required when interfacing with Motor Control Centers. In this case, the IP address mapped belongs to a gateway device, which in turn acts as a data concentrator communicating with multiple actual Modbus devices, each with a unique address.

The gateway device typically communicates serially with its slaves. For example, a Modbus Ethernet Bridge manufactured by Schneider Electric (part # 174CEV30020) is such a device. This is a network device that communicates via ModbusTCP with the VIM. The message packets are converted to standard Modbus and serially transmitted over RS-232 or RS-485 to slave devices. The subsequent responses are converted and transmitted to the VIM via ModbusTCP. Because of the serial communications (maximum baud rate of 19.2k), you can expect message times of 1-2 seconds.

The VIM has the capacity to communicate with up to 16 network devices simultaneously. The communications tasks in the VIM are all active concurrently, each handling the messaging for the configured device (with its unique IP address). Of the 16 network devices, any mix of TCP and UDP devices can be configured

In general each Modbus device is sent read/write requests for one dataset at a time. Depending on CPU load on the Modbus device the turn around can be as low as 10 msec. or as high as 200 msec. per dataset.

To increase throughput some Modbus devices, e.g. Schnieider’s 140 NOE 77101 allow multiple connections, with each connection handling 16 messages simultaneously. The rules of Modicon CPU loading still apply, however, this mechanism allows for higher throughput. The VIM makes use of this by allowing you to configure the maximum messages per device. See section 3.4 for details.

2.3 Messaging Options

Devices configured in the VIM are configured to communicate using RTU TCP, RTU via TCP or RTU via UDP.

- RTU TCP comprises the Open ModbusTCP message structure. Message packets contain a 6-byte header as defined by the ModbusTCP standard. Messages are received and transmitted using TCP.

- RTU via TCP is simply Modbus messaging encapsulated and transmitted over the network using TCP. Messages are received and transmitted using TCP.

- RTU via UDP is similar to RTU via TCP. It is simply Modbus messaging transmitted over the network using UDP.

3.0 VIMNet Plug and Play Server

3.1 Installation of Simplex Virtual I/O Module (VIM) Hardware

Step 1 – You will need two 2-wide carriers, 2 power supplies, one DeltaV controller and one VIM. Mount a power supply on the left side and the DeltaV controller on the right side of one 2-wide carrier. Mount a power supply on the left side and the VIM on the right side of the second 2-wide carrier. Connect the second 2-wide carrier to the left edge of the Controller 2-wide carrier. Repeat this step for all simplex VIM installations. The final assembly should be as follows:

Figure 3: Simplex VIM Assembly

Step 2 – Connect a network cable from the VIM bottom port to a single isolated switch.

Note

Do not use the DeltaV Primary or Secondary switches for VIM field communications.

Step 3 – Connect the PC with the VIMNet software to isolated switch connected to the VIM. The DeltaV ProPlus PC may be used to host the VIMNet Server. However, a separate network card must be used for VIMNet communications.

3.2 Installation of Redundant Virtual I/O Module (VIM) Hardware

Step 1 – You will need three 2-wide carriers, 3 power supplies, one DeltaV controller and two VIMs. Mount a power supply on the left side and the DeltaV controller on the right side of one 2-wide carrier. Mount a power supply on the left side and the VIM on the right side of the other two 2-wide carriers. Connect the two 2-wide VIM carriers together and to the left edge of the Controller 2-wide carrier. Repeat this step for all redundant VIM installations. The final assembly should be as follows:

Figure 4: Redundant VIM Assembly

Step 2 – Connect a network cable from each VIM Ethernet port to a separate dedicated isolated switch.

Note

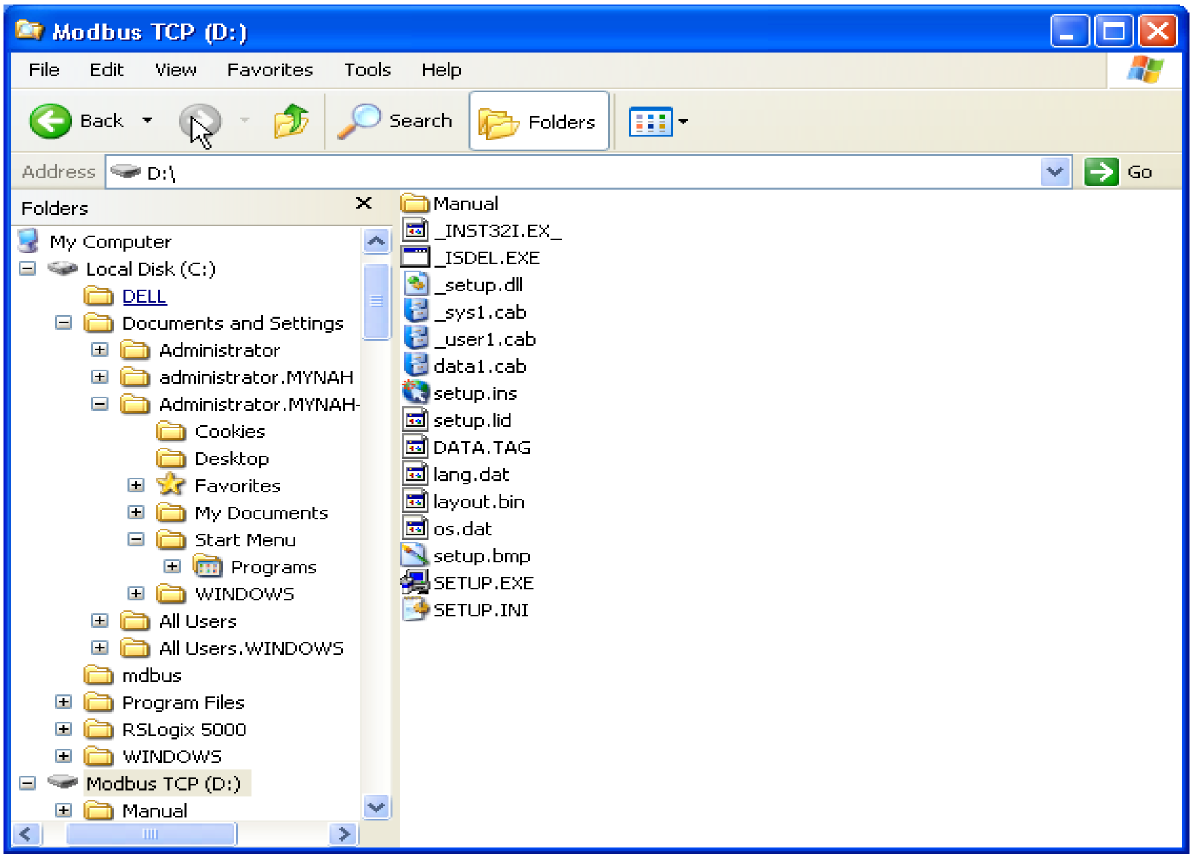
Do not use the DeltaV Primary or Secondary switches for VIM field communications.

Step 3 – Connect the PC with the VIMNet software to either one of the isolated switches connected to the VIMs. The DeltaV ProPlus PC may be used to host the VIMNet Server. However, a separate network card must be used for VIMNet communications.

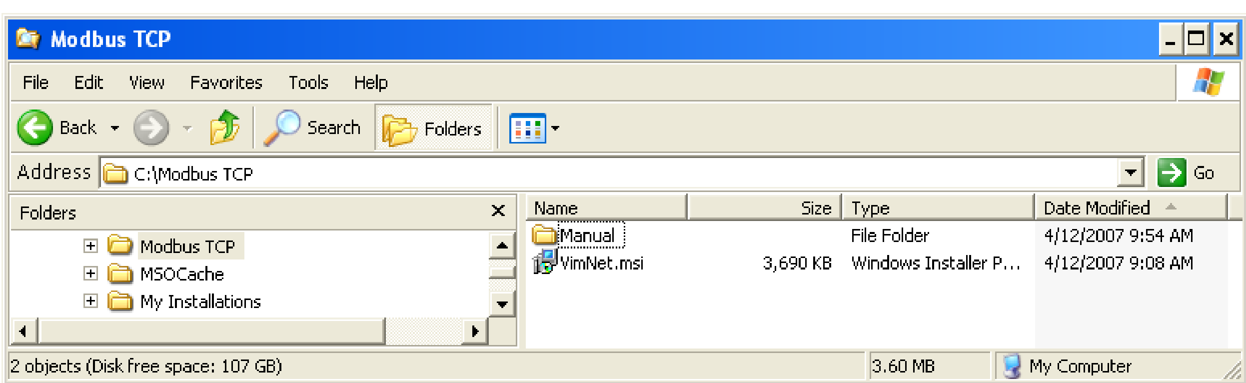
Step 4 – Connect the two switches together with a straight network cable. This is required so that the VIMs can communicate with each other. Configuration and status information is passed between the VIMs using this connection. If this cable is not installed, VIMs will not be visible to each other and DeltaV Diagnostics will show a Standby unavailable error message.

3.3 Installation of Software

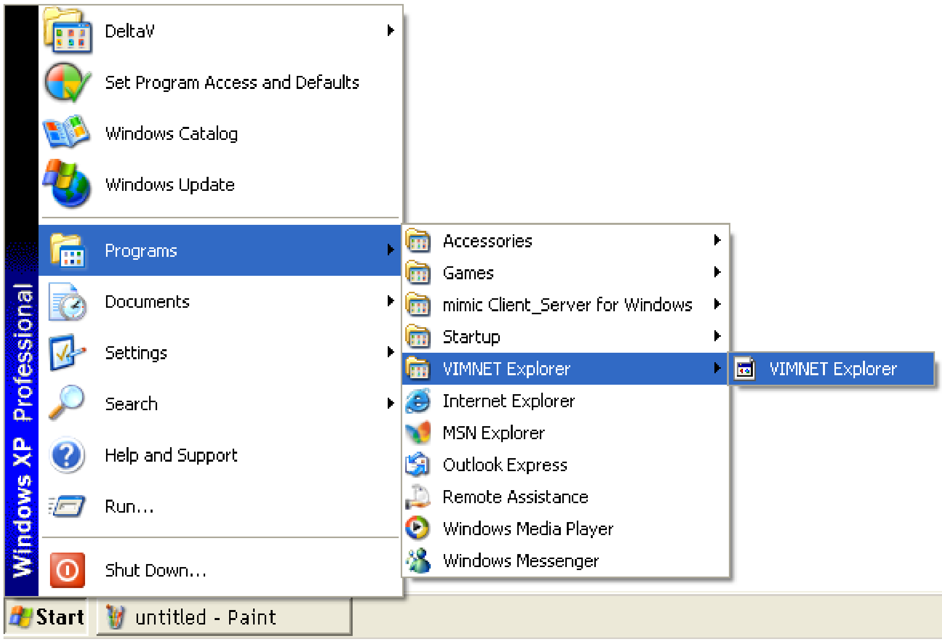
To install the software, insert the CD into the drive. Older distributions used the following installation files. Double Click on Setup.exe file to install VIMNet.



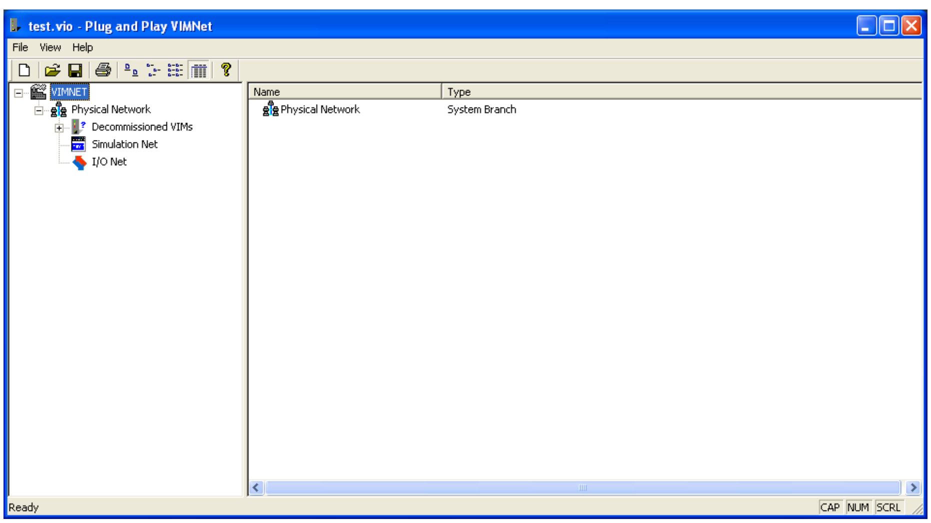
Newer distributions use a single MSI file called VimNet.MSI as shown below. Double Click on VimNet.MSI to install VIMNet.



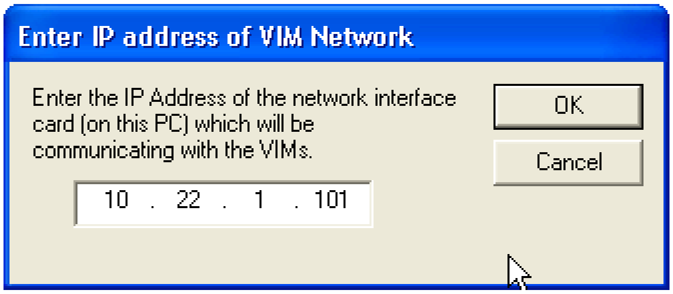
Step 1 – Launch the VIM Plug & Play Server by going to Start –> Programs –> VIMNet Explorer -> VIMNet Explorer.



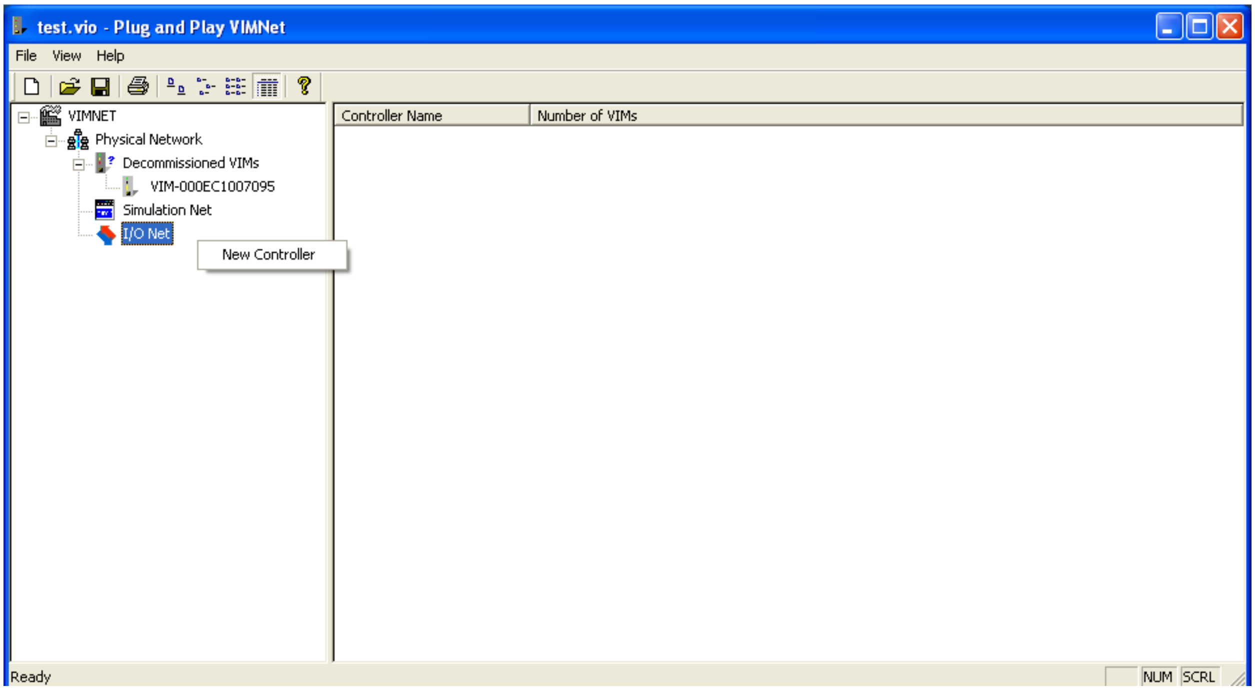
The following main VIMNet Explorer screen will be displayed:



Step 2 - Right click on Physical Network go to Properties. You will be prompted to enter the IP Address of the network communicating with the VIM. The IP address shown is a default. Change this to the IP address you are using, and then click OK.

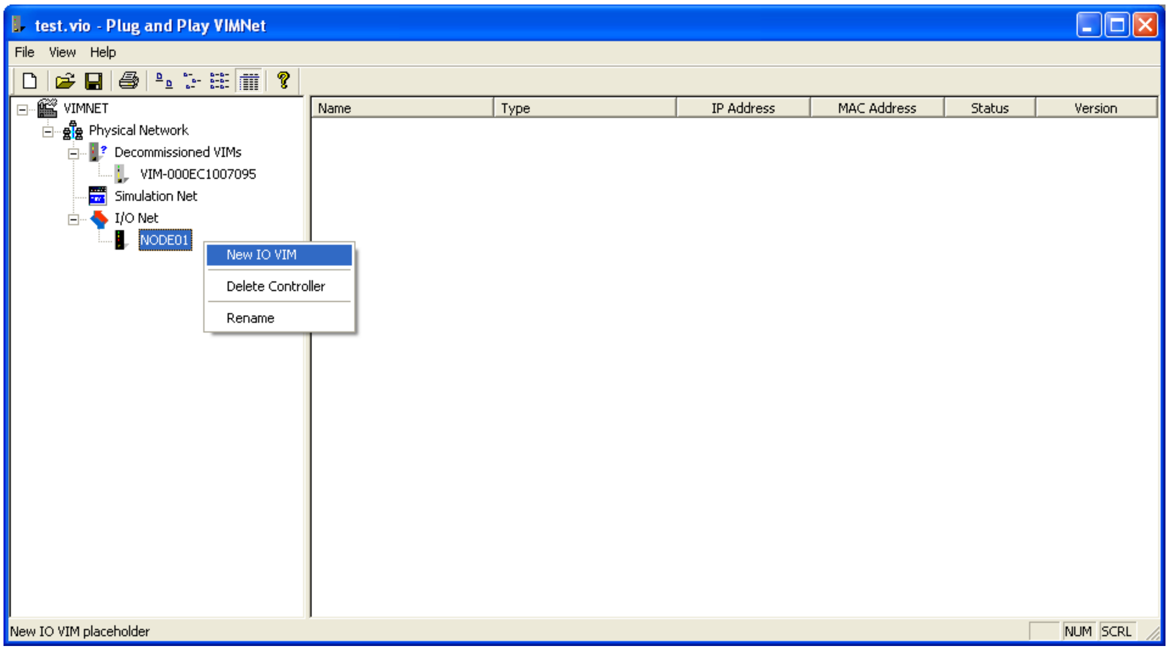


Step 3 – Right click on I/O Net and select New Controller menu option. This will create a controller object underneath I/O Net. The controller will have a default name, e.g., Node1. Rename the created controller to match the controller name in DeltaV.

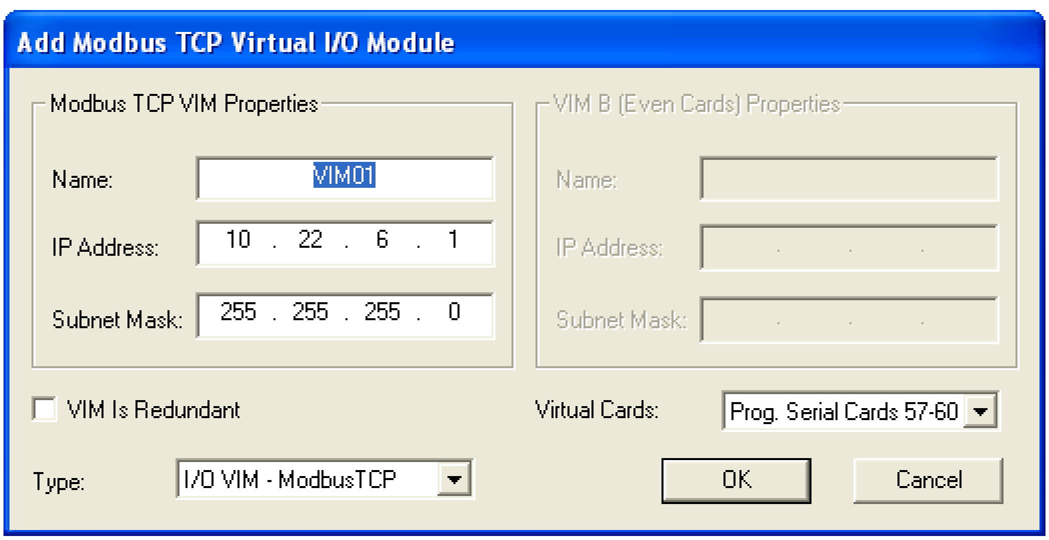


3.4 Configuring Simplex VIM

Step 1 – Right click on the Controller to Add Virtual I/O Module (VIM) placeholder.



A dialog box will appear to Add Virtual IO Module:



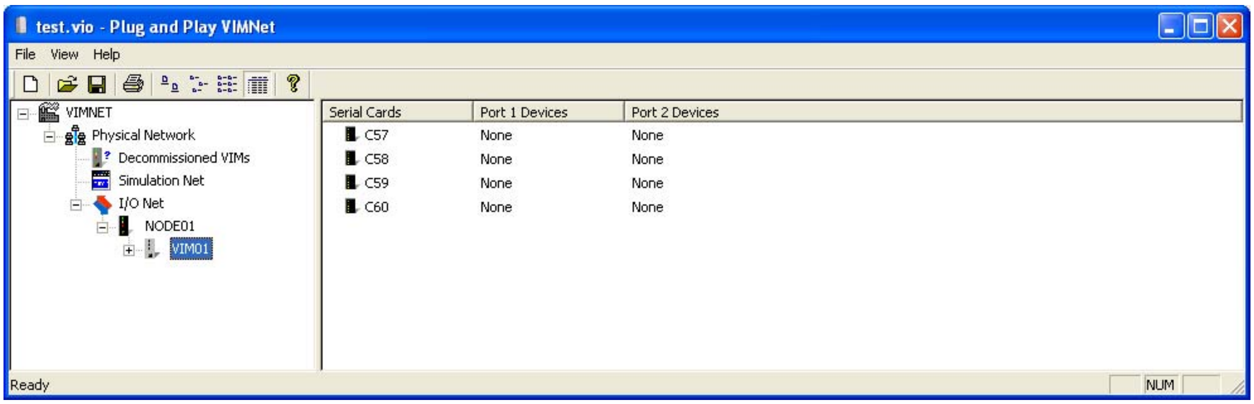
Fill in the parameters as follows:

a. Name – Unique 32 character VIM name
b. IP Address – an IP address in your network which is not currently being used
c. Subnet Mask – remains as default
d. Virtual Cards – select card group to be emulated by VIM, i.e., cards 57-60, or 61-64
e. Type – This is the VIM firmware type. Select Modbus TCP.

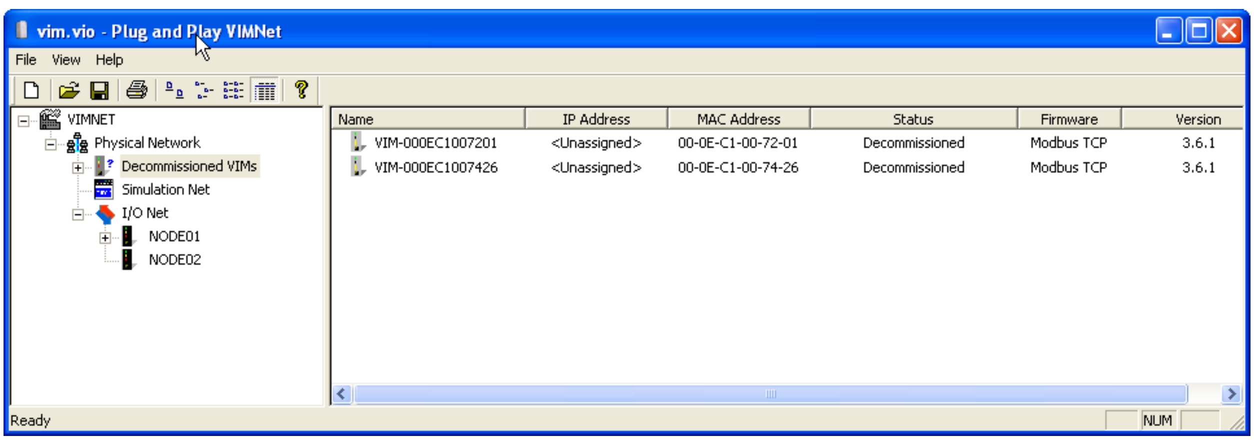
f. Redundancy – Leave the “VIM is Redundant” checkbox unchecked.

After filling out the parameters, select OK.

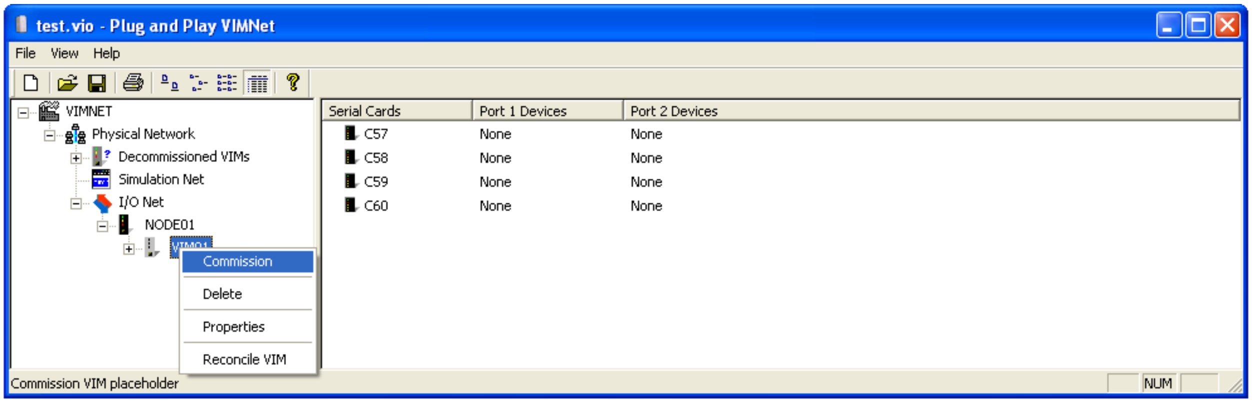
The Virtual I/O placeholder module has been added (note that the Virtual Cards appear and you are ready to commission.



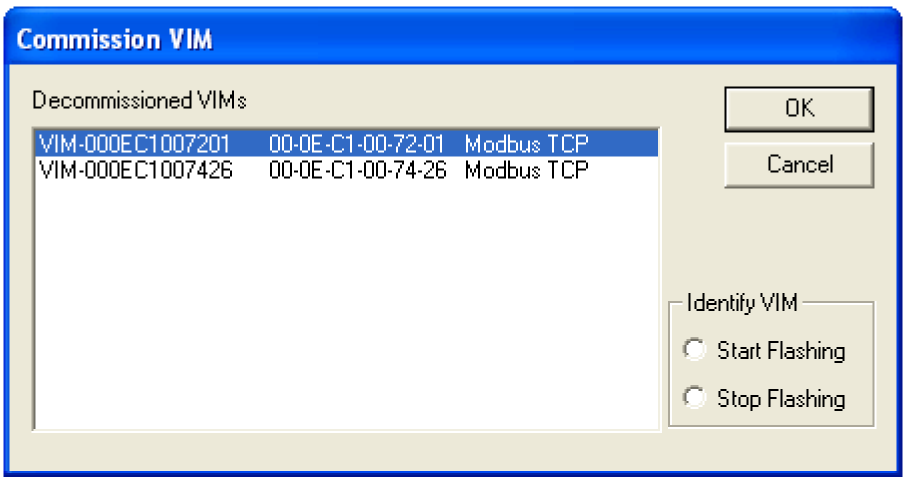
Step 2 – Click on Decommissioned VIMs to display all available decommissioned VIMs. This list is dynamically populated as VIMs are detected on the network. Click the Decommissioned VIMs object in the left pane and verify the VIMs appear in the list in the right pane.



Step 3 – Right click on the VIM placeholder under I/O Net and select the Commission menu option.



a. Select the VIM to be commissioned from the List of Decommissioned VIMs



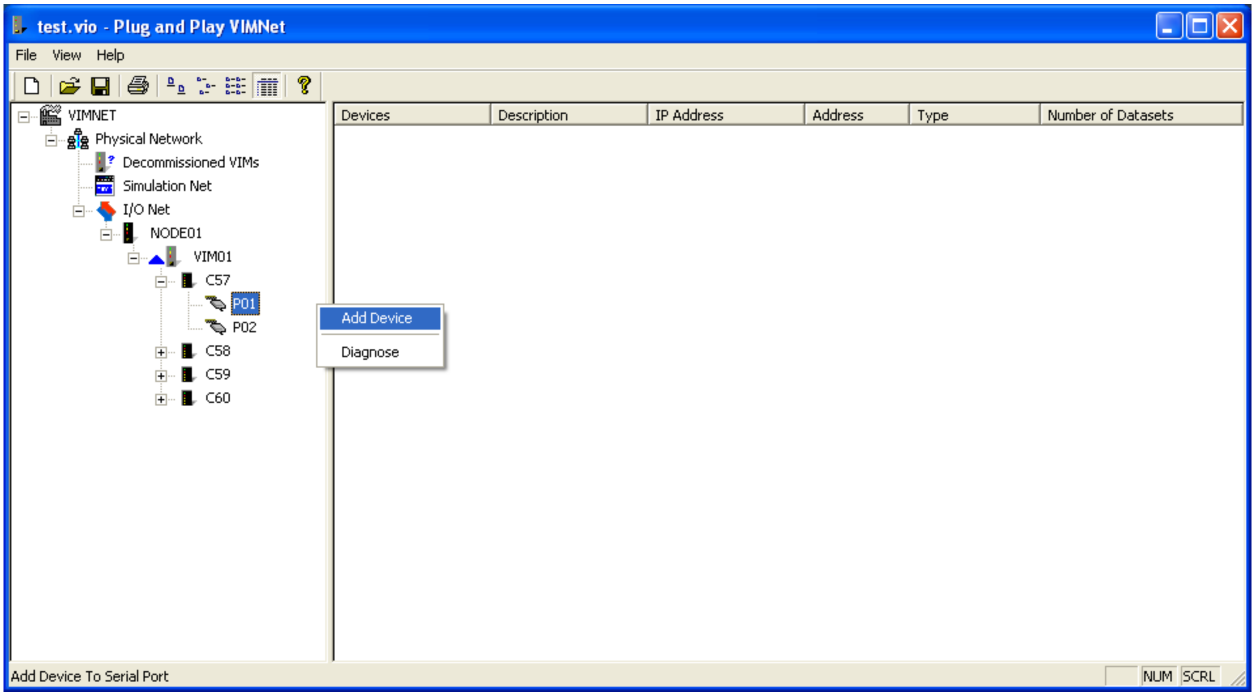
b. Select "Start Flashing" to identify the VIM you are commissioning. Once the correct VIM has been located, select Stop Flashing and then select OK.

If not located check the network connection and power supply. Cancel the dialog and repeat Step 6.

When commissioned, the Active LED will stay steady green and your state on the VIMNet Plug and Play Server will indicate that the commission is good. The Standby LED will remain off.

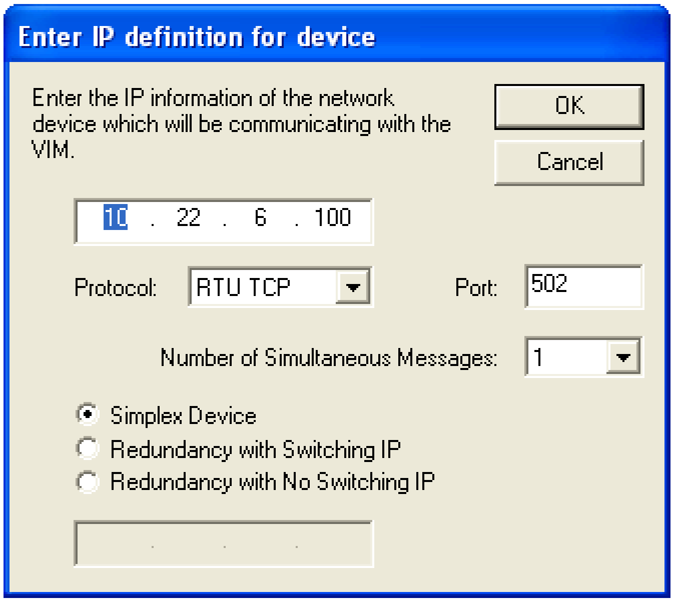
Step 4 – Repeat Steps 1, 2 and 3 for all VIMs.

Step 5 - To complete VIM configuration, network devices must be added to the virtual cards. Right click on the Serial Port and select Add Device.



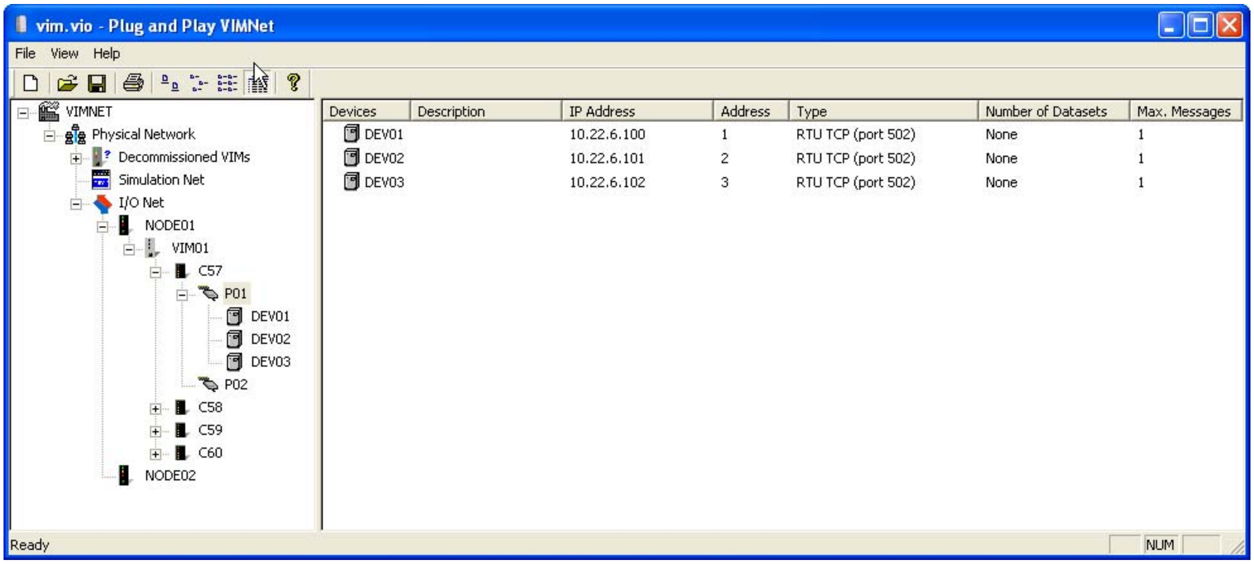
Fill in the parameters as follows:

a. Device Address – 1-255. This is the PLC address of the device, which must be the same as the device address configured in the DeltaV explorer for the serial card.
b. Description – up to 32 characters
c. Click Add to add a new IP address and specify its properties. The following dialog will appear. Note that unused IP addresses are automatically discarded from the list. All configured and available IP addresses are shown in the list. You can map a device to any available IP address. Furthermore, more than one device can be mapped to a single IP address.



a. Specify the IP address of the Modbus TCP PLC device.
b. Select the communication protocol to be used with the Device.
c. Enter 502 for the standard Modbus TCP Port Number. This can be modified as needed for the field device.
d. Adjust the Number of Simultaneous Messages (read/write requests) as needed. For example, Quantum PLCs allow a maximum of 16 requests.

Click OK after filling the parameters. The following window shows multiple devices configured.



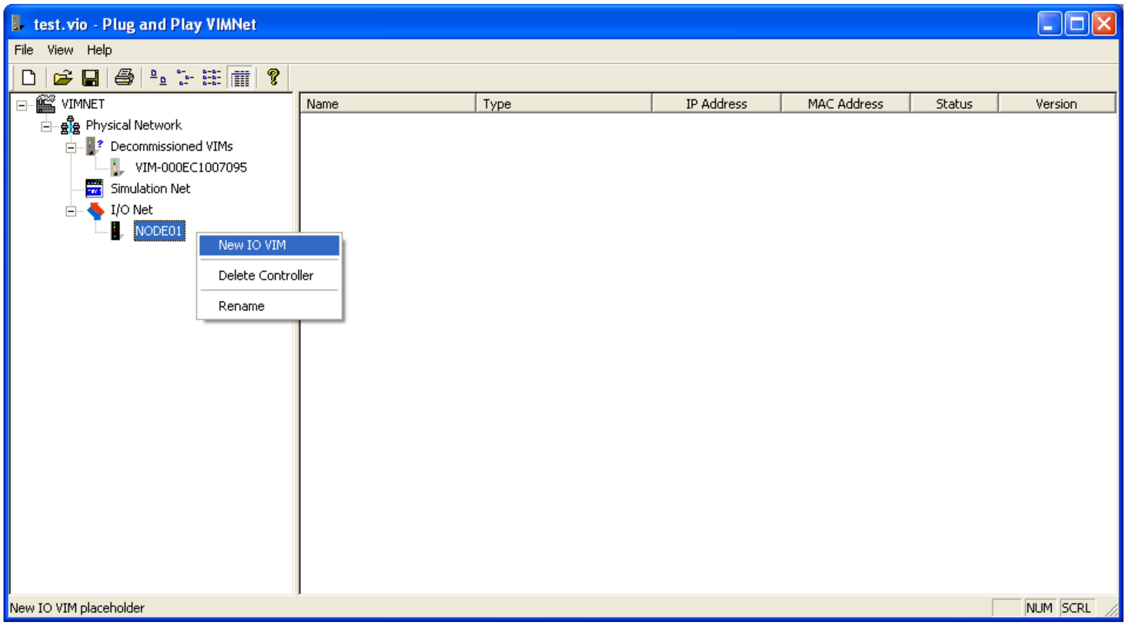
Note

The mapping of device address to IP address is the most critical part of the VIM configuration. Care must be exercised to ensure correctness.

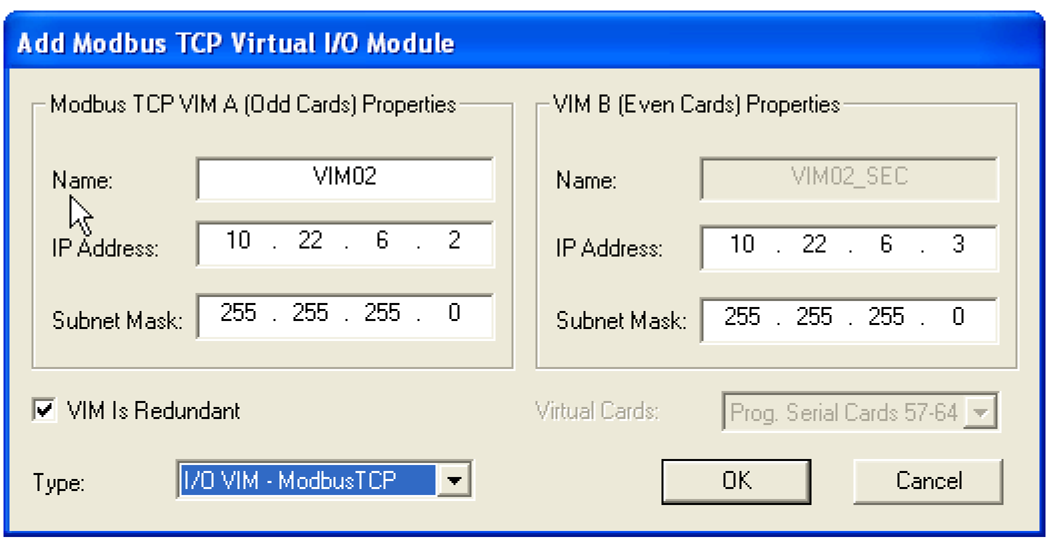
When you are finished configuring VIMs, continue to Section 3.6. To configure Redundant VIMs, continue to Section 3.5.

3.5 Configuring Redundant VIM

Step 1 – Right click on the Controller to Add Virtual I/O Module (VIM) placeholder.



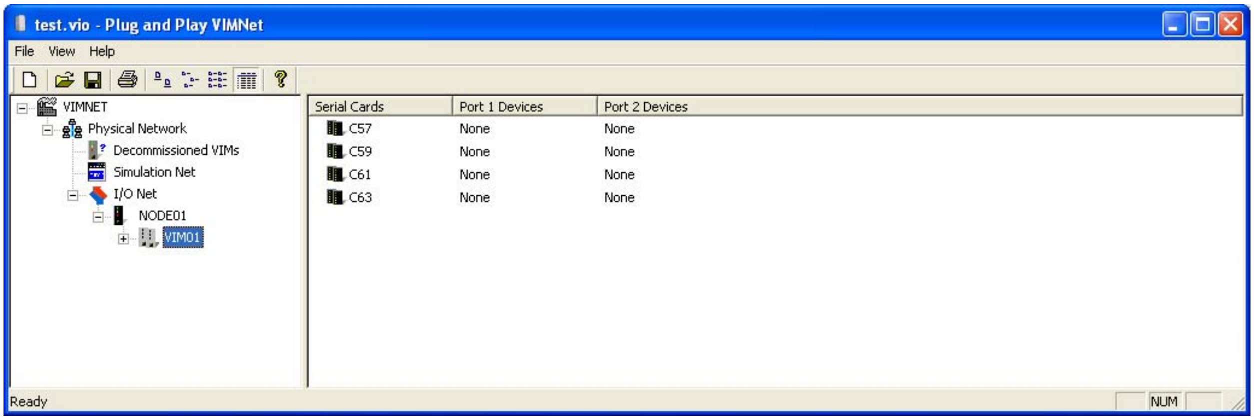
A dialog box will appear to Add Modbus TCP Virtual I/O Module



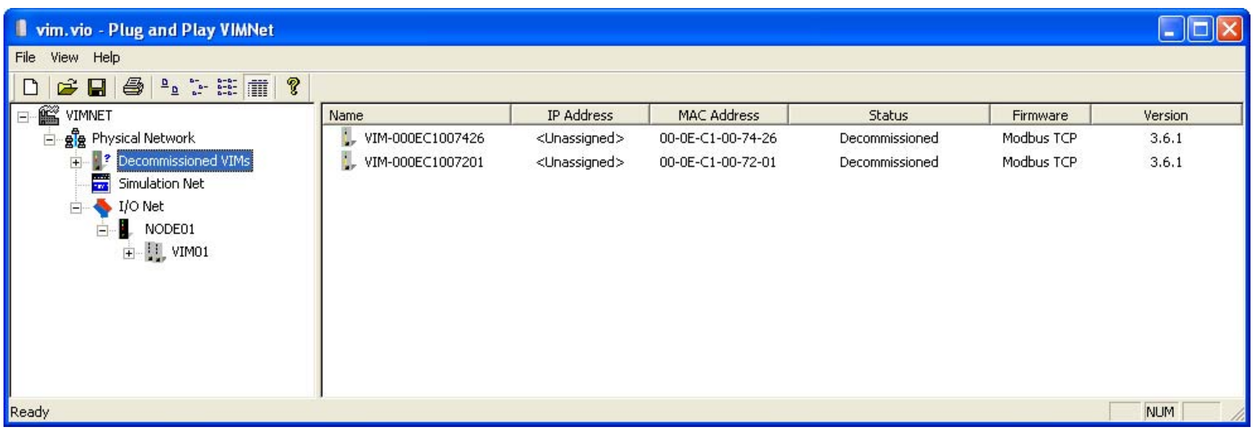
Fill in the parameters as follows:

a. Name – Unique 32 character VIM name.
b. IP Address – IP addresses in your network that are not currently being used.
c. Subnet Mask – as required by your network architecture.
d. Select the "VIM is Redundant" checkbox.
e. Virtual Cards – All eight serial cards will be allocated as four redundant pairs.
f. Type – This is the VIM firmware type. Select Modbus TCP.

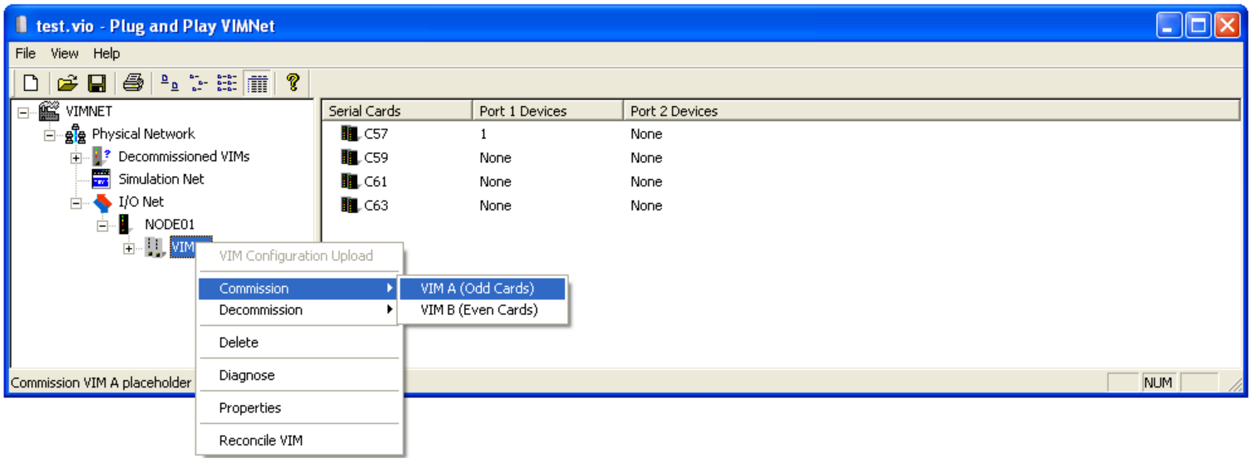
After filling out the parameters, select OK. The Virtual I/O placeholder module has been added. (Note that the Virtual Cards appear and you are ready to commission.)



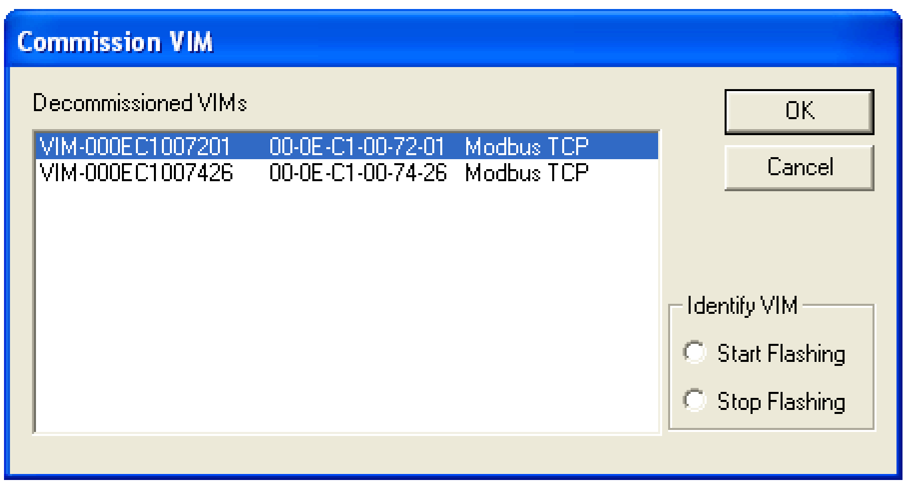
Step 2 – Click on Decommissioned VIMs to display all available decommissioned VIMs. This list is dynamically populated as VIMs are detected on the network. Click the Decommissioned VIMs object in the left pane and verify the VIMs appear in the list in the right pane.



Step 3 – Right click on the VIM placeholder under I/O Net and select the Commission menu option. You must commission both VIMs separately as VIM A and VIM B.



Select the VIM to be commissioned from the list of Decommissioned VIMs.



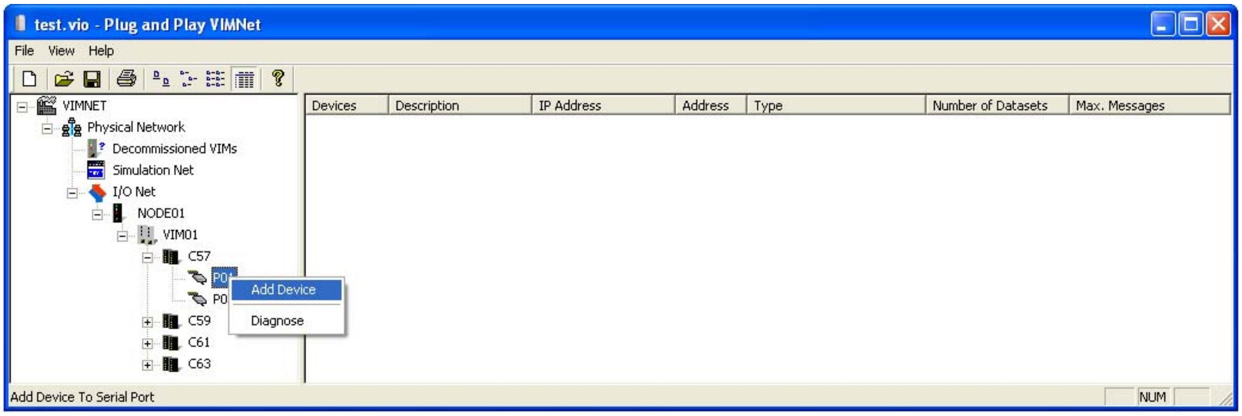
Select the Start Flashing Radio Button to identify the VIM you are commissioning. Once the correct VIM has been located, select Stop Flashing and then select OK.

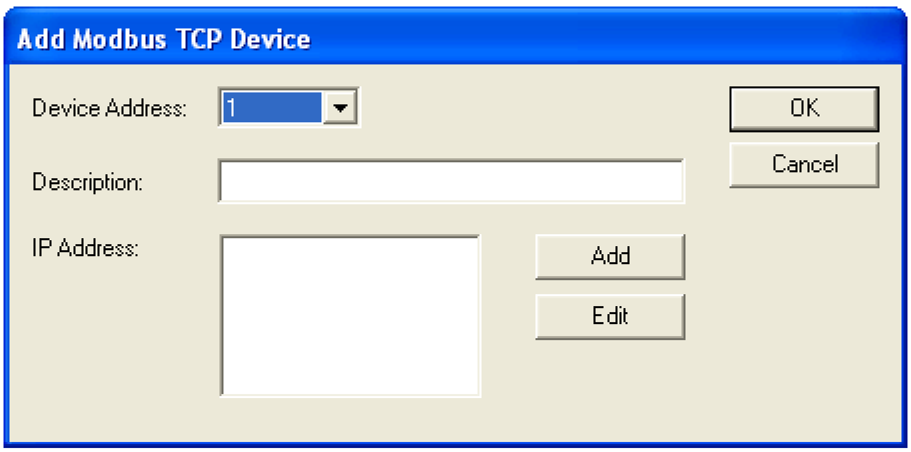
If the VIM cannot be located, check the network connection and power supply. Cancel the dialog and repeat Step 3.

When commissioned the Active LED will stay steady green and your state on the VIMNet Plug and Play Server will indicate commission good. The Standby LED state will change based on redundancy role.

Step 4 – Repeat Step 3 for the partner VIM.

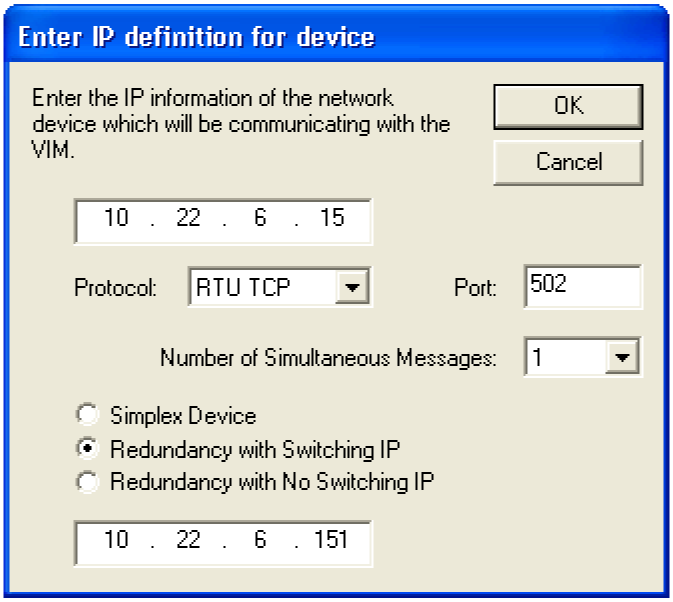
Step 5 - To complete VIM configuration, Network devices must be added to the Virtual Cards. Right Click on the Serial Port and select the Add Device menu option.





Fill in the parameters as follows:

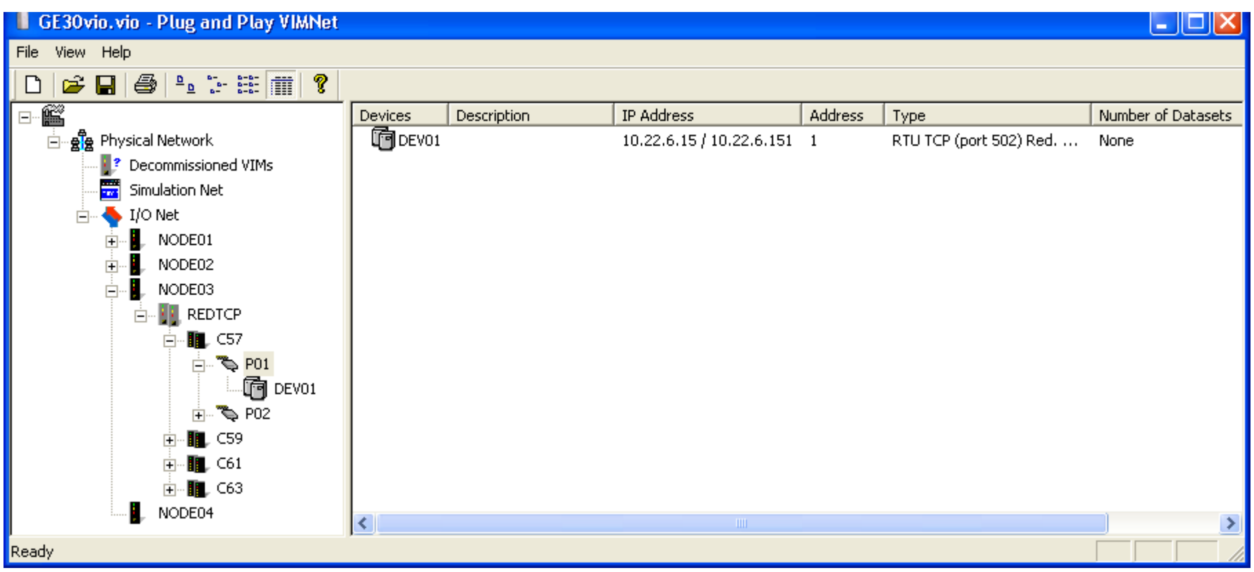
a. Device Address – 1-255. This is the PLC address of the device, which must be the same as the device address configured in the DeltaV explorer for the serial card.
b. Description – up to 32 characters
c. Click Add to add a new IP address and specify its properties. The following dialog will appear. Note that unused IP addresses are automatically discarded from the list. All configured and available IP addresses are shown in the list. You can map a device to any available IP address. Furthermore, more than one device can be mapped to a single IP address.



In this dialog:

a. Specify the IP address of the Modbus TCP PLC device.
b. Select the type of device redundancy being used. Device redundancy is described in Section 7.
c. Enter 502 for the standard Modbus TCP Port Number. This can be modified as needed for the field device.
d. Adjust the Number of Simultaneous Messages (read/write requests) as needed. For example, Quantum PLCs allow a maximum of 16 requests.

Click OK after filling the parameters. The following window shows multiple devices configured.



Note

The mapping of device address to IP address is the most critical part of the VIM configuration. Care must be exercised to ensure correctness.

3.6 Uploading a VIM Configuration

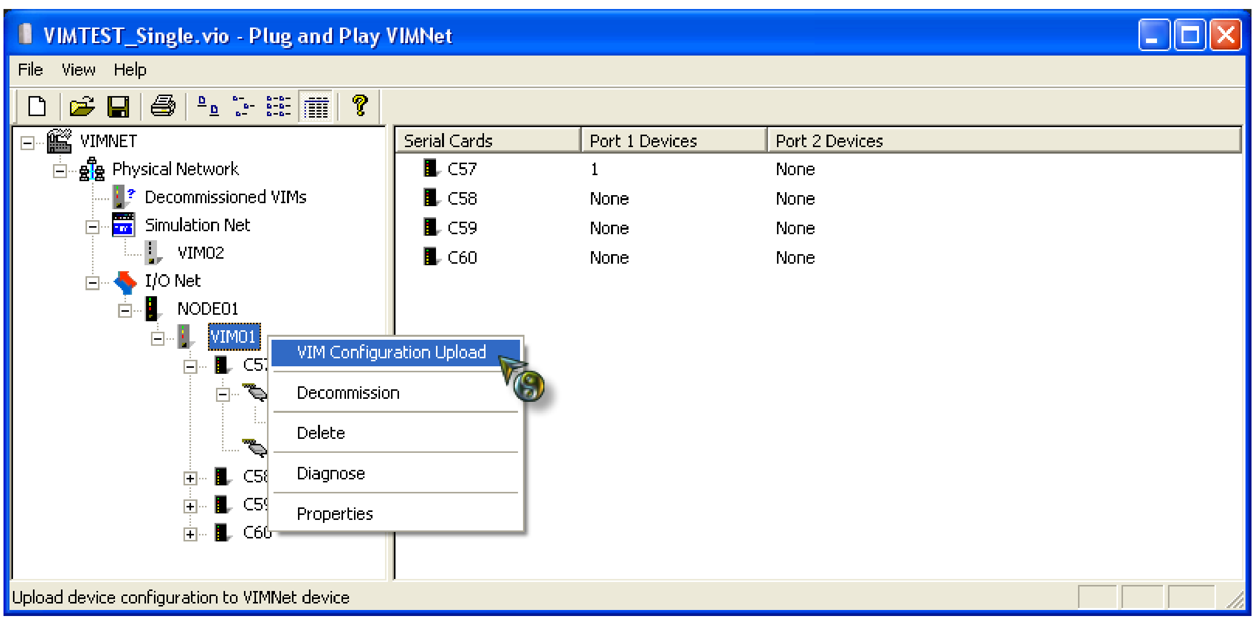
VIMNet configuration creates a mapping between PLC device addresses and IP addresses. This mapping must be uploaded into the VIM for proper communications. A configuration that has not been uploaded to the VIM is indicated with a blue triangle next to the VIM icon. To upload a configuration, the VIM must first be commissioned.

Caution

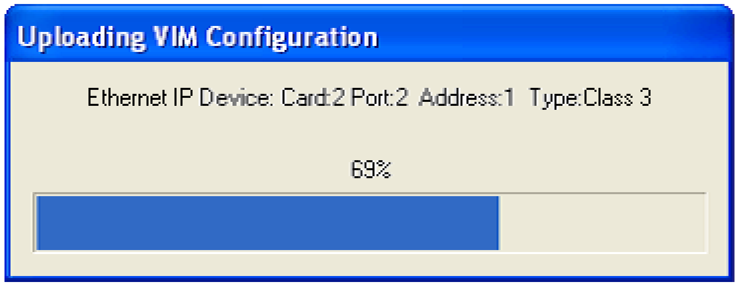
Uploading a new configuration into the VIM will cause all field communications to terminate. After upload completion, the VIM will automatically reboot.

VIM upload must be done with the process in safe mode.

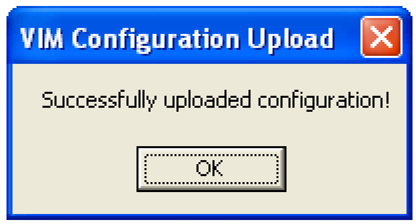
Right click on the VIM and select VIM Configuration Upload.



The Uploading VIM Configuration progress bar will indicate the status of the upload:



Upon successful completion of the VIM Configuration Upload, click OK.



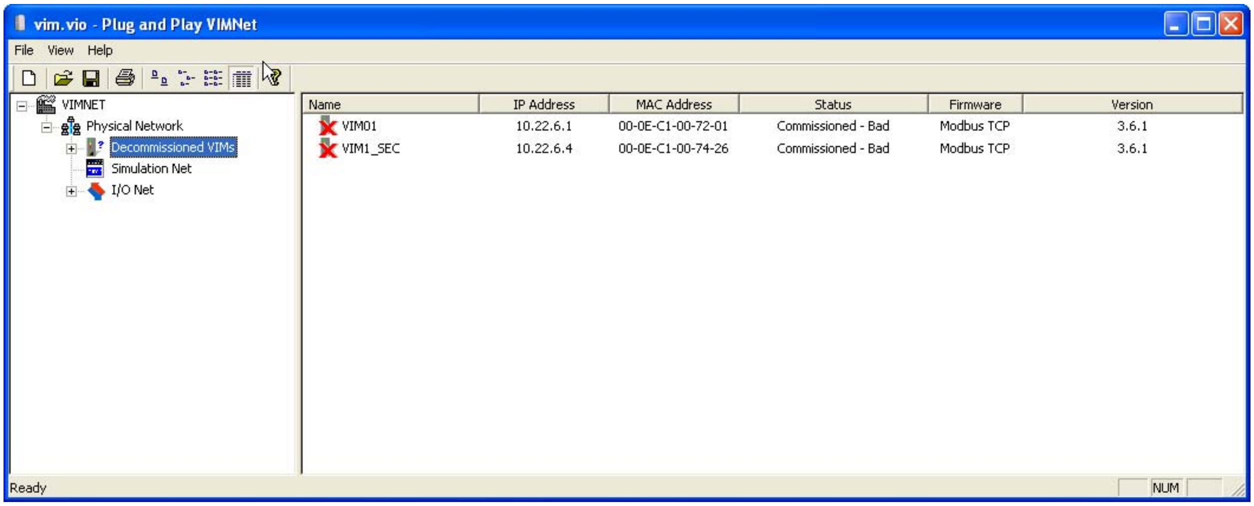
The upload process terminates all communications with DeltaV over the railbus. Upon upload completion, the VIM automatically reboots and goes online. Click OK to terminate the dialog.

If your upload is unsuccessful, you will need to decommission and re-commission the VIM and try again. Contact MYNAH Support if you are not successful in uploading.

3.7 Saving the VIM Configuration

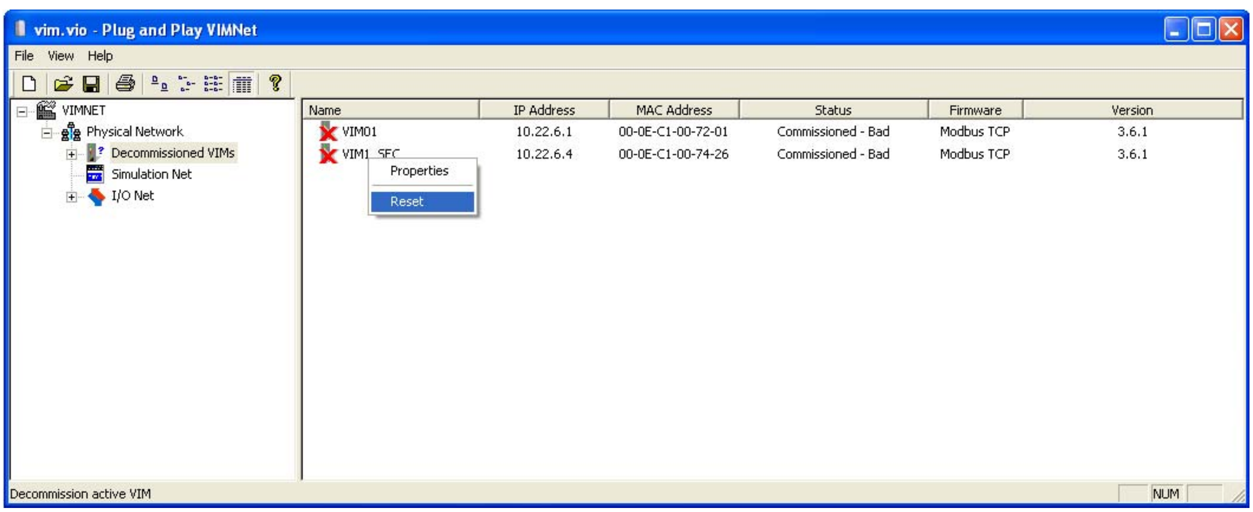
VIMNet configuration is saved in a file with a VIO extension. This file can be located anywhere in the PC local or network folder. The current state of commissioned VIMs, as well as VIM network device configurations is contained in this file. The VIMNet Explorer does not have to be online all the time. However, if it is restarted, this file should be reopened so that the current state of VIMs does not show as error.

When the VIMNet Explorer is restarted, it will start scanning for VIMs on the network, and display what is found. Commissioned VIMs found on the network will be compared with configured placeholders and there current state displayed. Mismatched VIMs, i.e., those which do not exist as placeholders, or mismatches in MAC address or IP address will be displayed in the Decommissioned list as errors. The following shows VIMs in error.



If the original configuration file is not available, the VIMs in error must be manually cleared. The options are to either Reset the VIM in the Decommissioned list, or to Reconcile the mismatched VIM with a configured placeholder in the I/O Net.

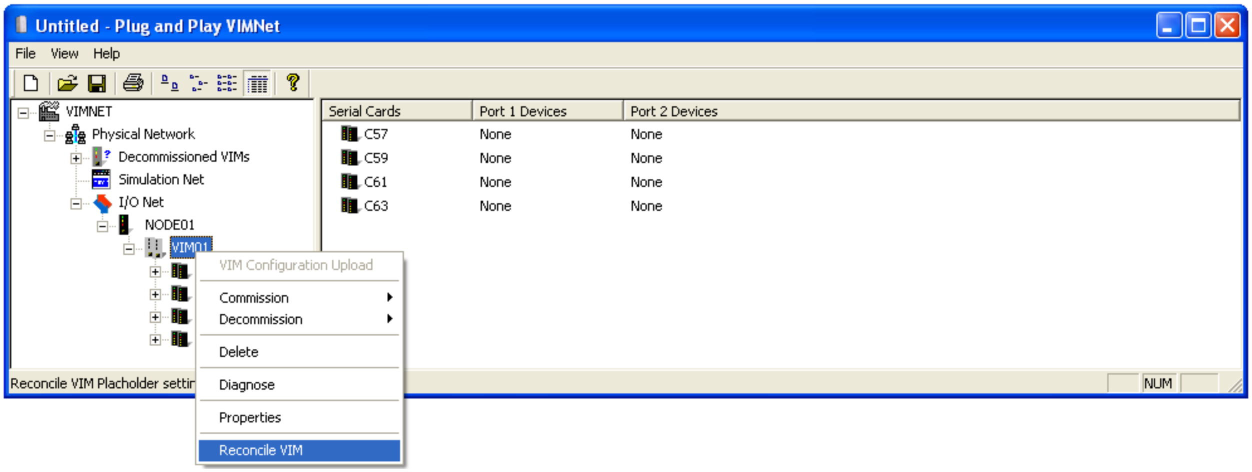
To Reset a VIM, right click on the VIM in the Decommissioned list to get the context menu. Then select Reset as shown below. The VIMNet Explorer will send a Decommission command over the network, and clear the VIM from its list. It is anticipated that the Decommission command will be accepted by the VIM resulting in a decommissioned VIM. The VIM will then appear as an unconfigured, decommissioned VIM in the VIMNet Explorer list.



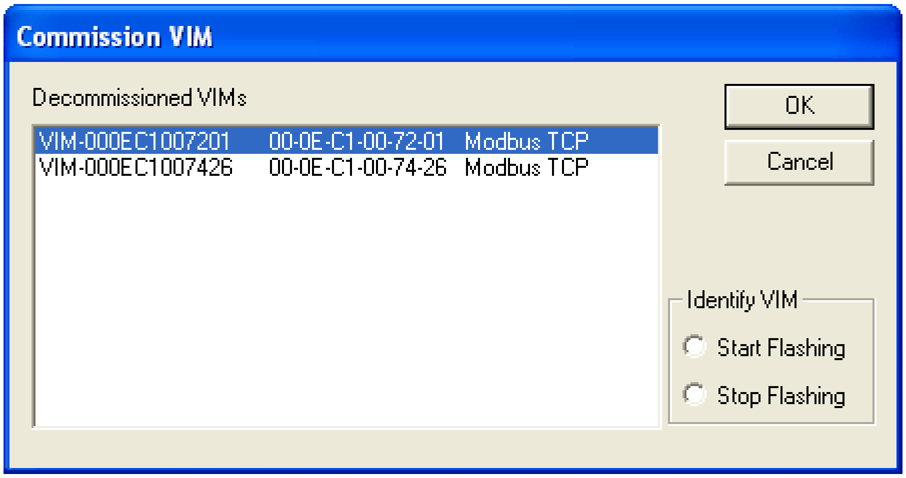
Performing a Reset will decommission a VIM. This will terminate all field communications.

The process of reconciling a detected, commissioned VIM, with an unassigned placeholder allows you to reconstruct a configuration file without decommissioning and then recommissioning the VIM. To

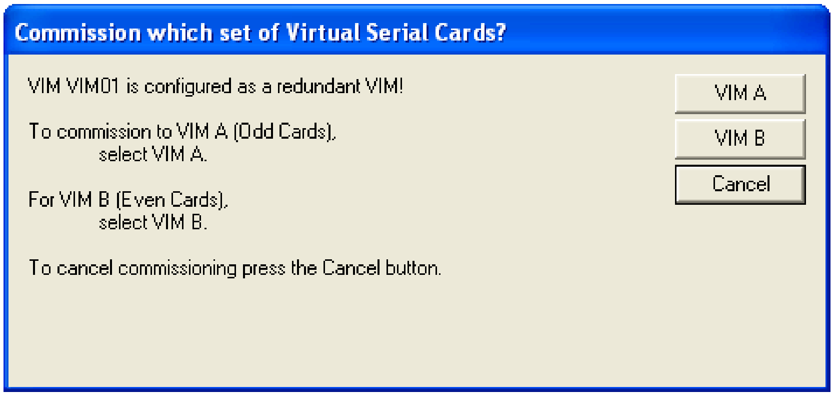
Reconcile a VIM, right click on the VIM in the I/O Net to get the context menu, then select Reconcile VIM menu option as shown below.



This will launch a dialog as follows, showing all the detected, commissioned, and unattached VIMS.



Select a VIM in the list and click OK. If the VIM placeholder is redundant and both VIMs are unattached, a dialog will be displayed as follows where you can select VIM A or VIM B.



If the VIM placeholder is simplex or if only one VIM out of a redundant pair is unattached, the reconcile process with immediately create the link without further prompts.

The reconciled VIM will appear as normal and commissioned, and the decommissioned list will be cleared. Note that if you are creating a new configuration file, you must recreate the field device network definitions and then upload to the VIM.

3.8 Flash Upgrade of the VIM

For VIM functionality changes, MYNAH Technologies will issue firmware upgrade files as required. The new firmware files must be flashed into the VIM.

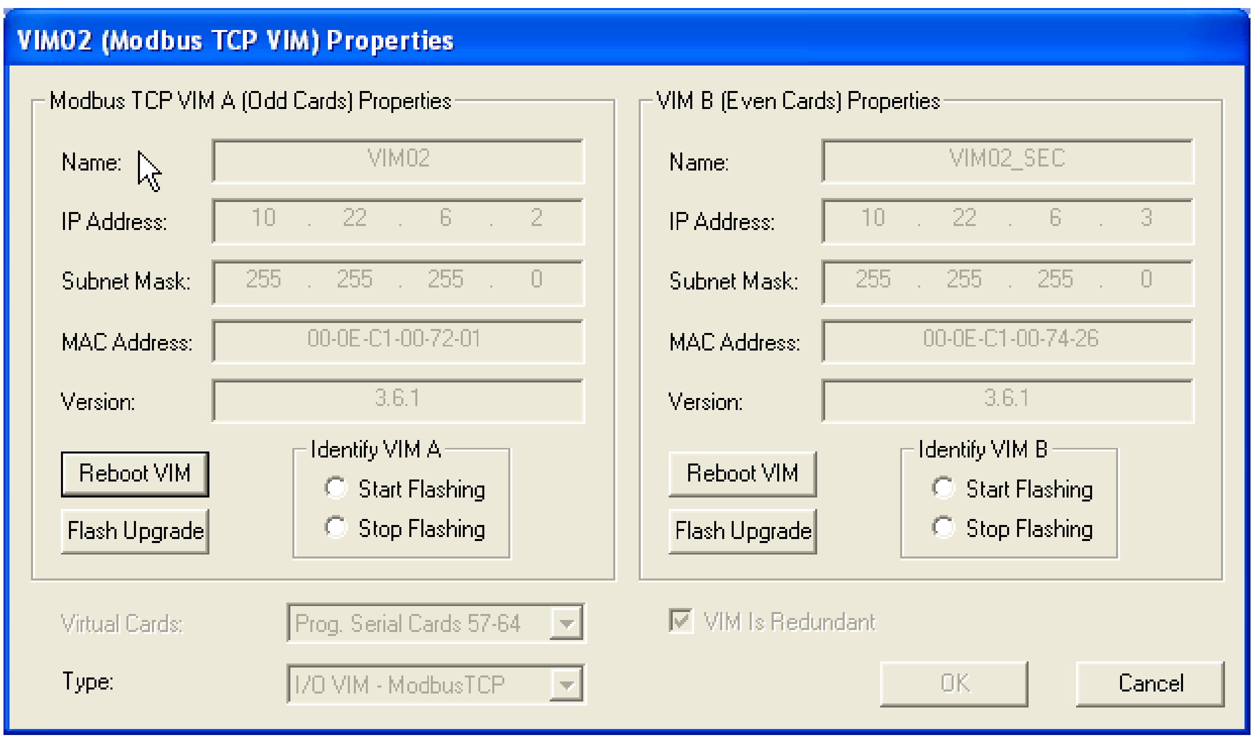
If your current operating firmware version is v3.5.7 or earlier, please contact Mynah technical support for instructions on how to upgrade to the latest system.

Caution

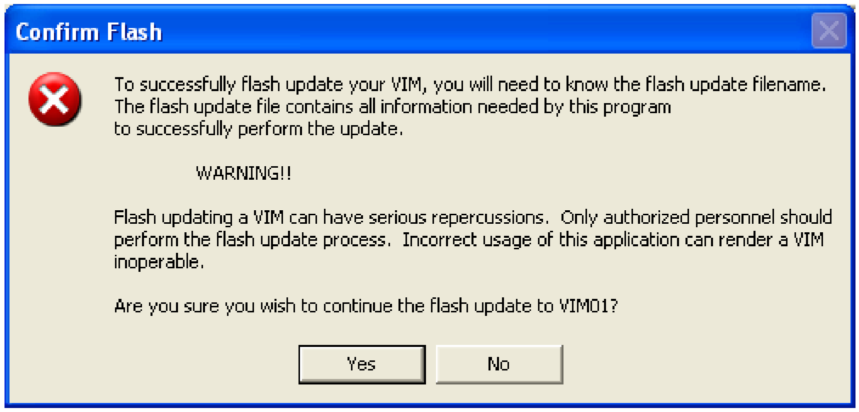
Flashing VIM (Simplex or Redundant) with new firmware will cause all field communications to terminate. Upon flash completion, the VIM will automatically reboot.

VIM flash must be done with the process in safe mode.

To do this, right click on the target VIM object and select Properties. The following dialog box will appear:



Click Flash Upgrade. A warning will appear as follows:

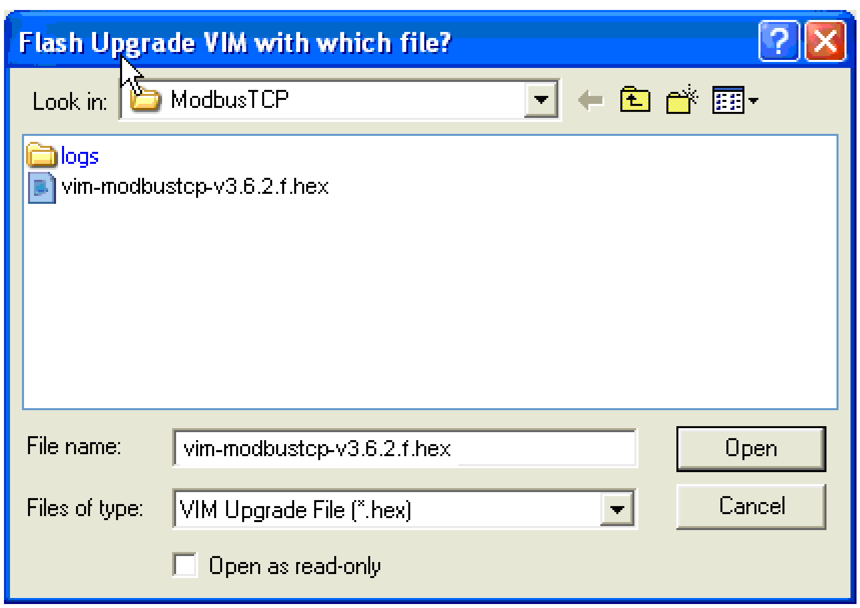


Click Yes to start the flash process. Note that while flashing the VIM, all communications with DeltaV Controller are terminated.

Browse to select the firmware file. Firmware files have a .HEX extension. Please contact Mynah Technical Support for the correct file to use.

Warning

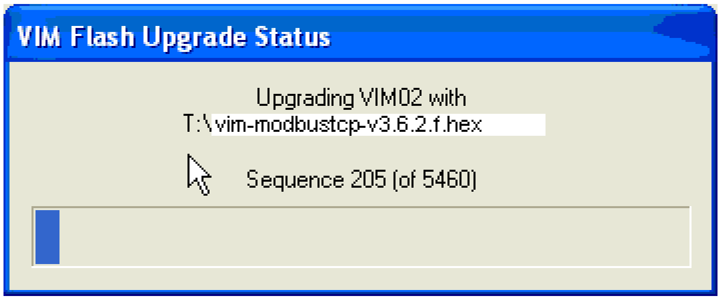
Using an incorrect firmware file may render the VIM inoperable.



Select the file to continue the flash upgrade process.

Note that the file format for the VIM should be:
vim-modbustcp-vmajor version.minor version.maintenance build.**p**artial or **f**ull.hex.

Once the file has been selected, a connection is opened to the VIM and the flash system is downloaded. During the download, a progress bar will display as follows:

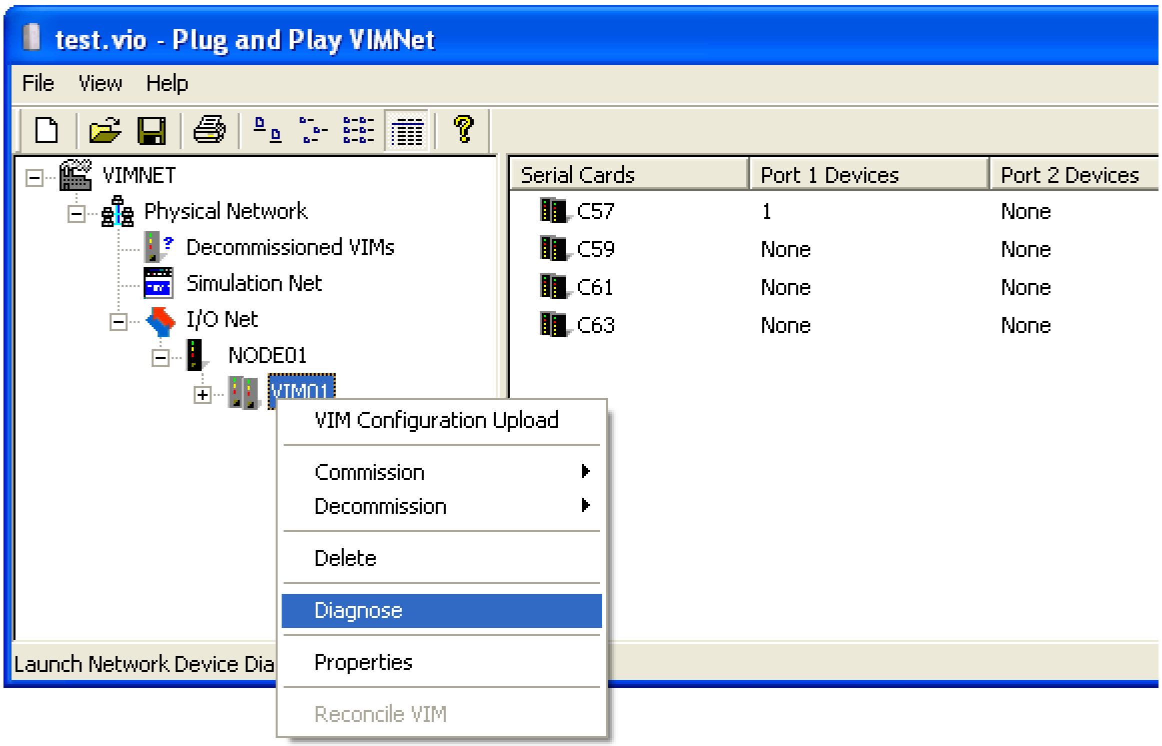


Upon completion, the VIM will reboot and go online.

In case of redundant VIMs, both must be flashed separately to the same firmware revision.

4.0 VIMNet Diagnostics

VIMNet Diagnostics are provided to assist you in troubleshooting abnormal situations, and to view network communications statistics. VIMNet Diagnostics can be launched multiple times, once for each active VIM in the network. Or a single instance of Diagnostics can be used to view all active VIMs. Launch the Diagnostics application by right clicking on the commissioned VIM in the VIMNet Explorer as follows:



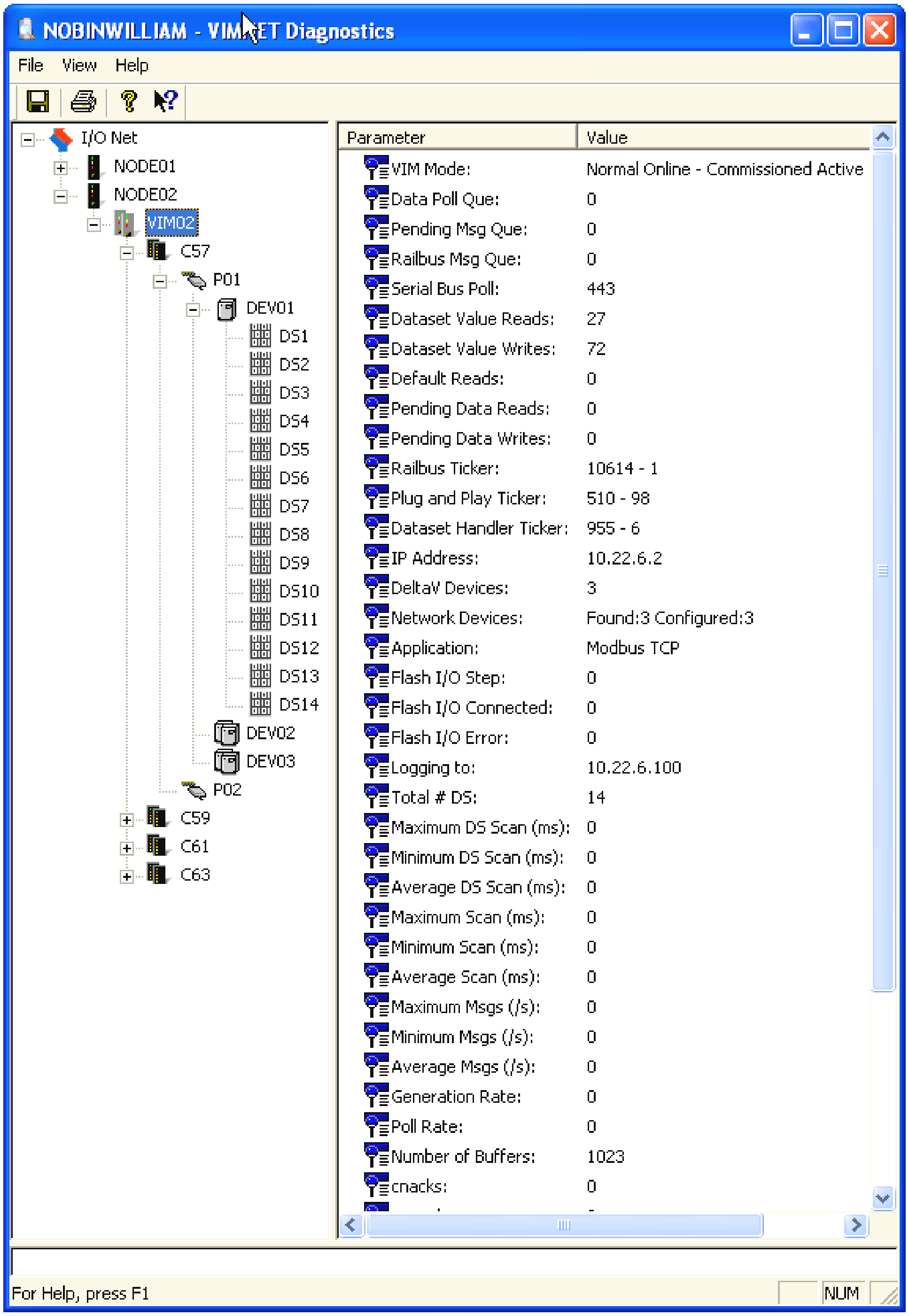
Note that diagnostics for simplex and redundant VIMs are identical.

When the diagnostics application is launched, it opens a network connection with the VIM specifically to read diagnostic information. The information is continuously scanned and displayed in the window. You can select the scan rate. However, the default rate is 1 second.

Diagnostic information is displayed at each level of the VIM architecture. You can drill down to the dataset level, which is the lowest level. The following screens show diagnostic information at each level, starting with the VIM level.

4.1 VIM Level Diagnostics

The first screen after launch is as follows:



The information displayed in this window is explained in Table 2 below:

Diagnostic Item	Description
VIM Mode	Shows current mode: Commissioned, Failsafe, etc.
Data Poll Queue	Number of messages waiting to be sent to DeltaV
Pending Message Queue	Number of waiting diagnostics message responses to be sent to DeltaV
Railbus Message Queue	Number of waiting Railbus messages received from DeltaV to be processed
Serial Bus Poll	Counter of poll requests received from Controller
Dataset Value Reads	Counter of dataset value read requests received from Controller
Dataset Value Writes	Counter of dataset value write requests received from Controller
Default Reads	Counter of default read requests received from Controller
Pending Data Reads	Counter of pending data read requests received from Controller
Pending Data Writes	Counter of pending data write requests received from Controller
Railbus Ticker	Ticker of process handling Railbus messages
Plug and Play Ticker	Ticker of process handling Plug/Play messages
Dataset Handler Ticker	Ticker of process handling dataset updates
IP Address	IP address of VIM
DeltaV Devices	Number of DeltaV devices in configuration from Controller
Network Devices	Number of devices configured/found
Application	Application type: Modbus TCP or ModbusTCP
Flash I/O Step	Reserved for Flash evaluation
Flash I/O Connected	Reserved for Flash evaluation
Flash I/O Error	Reserved for Flash evaluation
Logging to	IP address of PC if message logging is turned on
Total # DS	Total number of datasets in this configuration
Maximum DS Scan (ms)	Maximum scan time (ms) for single dataset based on 16 simultaneous messages
Minimum DS Scan (ms)	Minimum scan time (ms) for single dataset based on 16 simultaneous messages
Average DS Scan (ms)	Average scan time (ms) for single dataset based on 16 simultaneous messages
Maximum Scan (ms)	Maximum scan time (ms) for all datasets
Minimum Scan (ms)	Minimum scan time (ms) for all datasets
Average Scan (ms)	Average scan time (ms) for all datasets
Maximum Msgs(/s)	Maximum messages per second
Minimum Msgs (/s)	Minimum messages per second
Average Msgs (/s)	Average messages per second
Generation Rate (/s)	Rate of DS changes detected in field data
Poll Rate (/s)	Rate of DS Polls received from DeltaV controller

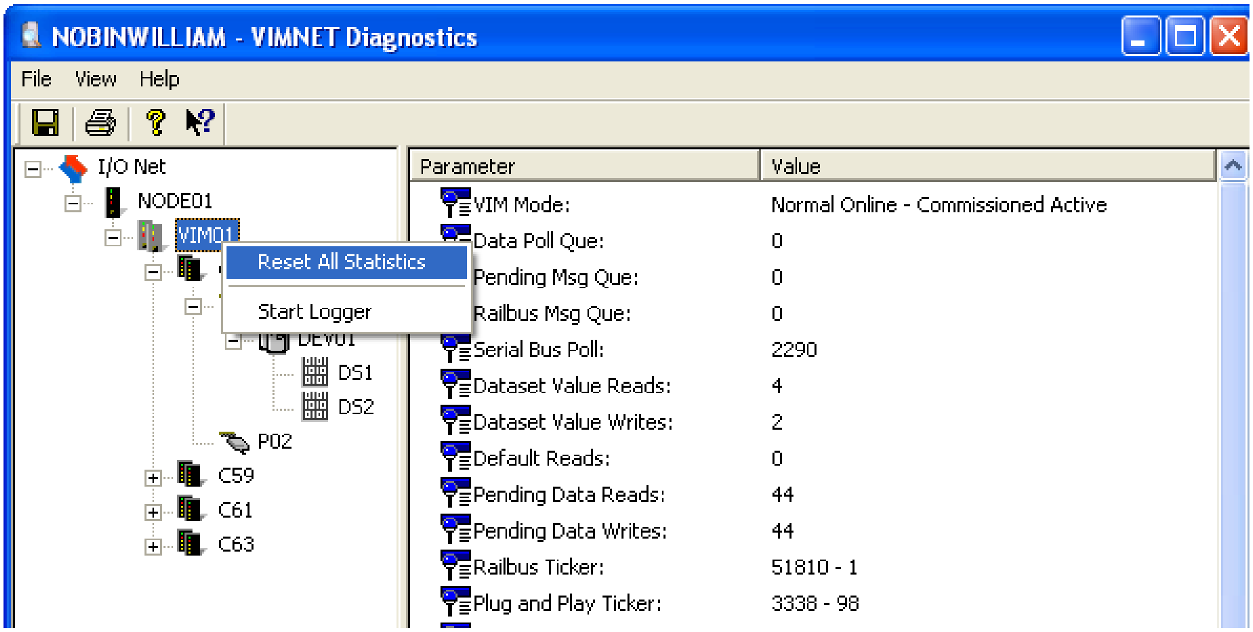
Table 2: VIMNet Diagnostics

This information is also sent to DeltaV if a VIMNet Diagnostics data is configured. Please refer to Section 5 for dataset configuration. The data transmitted to DeltaV dataset is as follows:

Register	Diagnostics Value
R1	VIM Mode O – Normal Online 1 – FailSafe Mode
R2	Number of Network devices
R3	Data Poll Queue
R4	Pending Message Queue
R5	Railbus Message Queue
R6	Counter - Serial Bus Poll
R7	Counter – Dataset Value Reads
R8	Counter – Dataset Value Writes
R9	Counter – Default Reads
R10	Counter – Pending Data Reads
R11	Counter – Pending Data Writes
R12	Total number of Datasets
R13	Maximum DS Scan
R14	Minimum DS Scan
R15	Average DS Scan
R16	Maximum Messages
R17	Minimum Messages
R18	Average Messages
R19	Maximum Scan Time
R20	Minimum Scan Time
R21	Average Scan Time
R22	Ticker - Railbus message handler
R23	Ticker – Plug and Play message handler
R24	Ticker – Dataset handler
R25	Rate of DS changes detected
R26	Rate of DS Polls received
R27	Logger IP address
R28	VIM Application type
R29	Current redundancy State Bits 0-3 are the VIM State as follows: 0000 – Decommissioned 0001 – Commissioned 0010 – Flash Mode 0011 – Configuration Mode Bits 4-5 are the Redundancy state as follows: 00 – Simplex 01 – Redundant Active 02 – Redundant Backup Note that DeltaV always reads the Active VIM. Consequently this value should always be 0x11.
R30	VIM Revision number

Table 3: VIMNet Diagnostics Dataset

You can right click on the VIM to get a context menu. From this menu, you can clear all statistics by selecting Reset All Statistics.

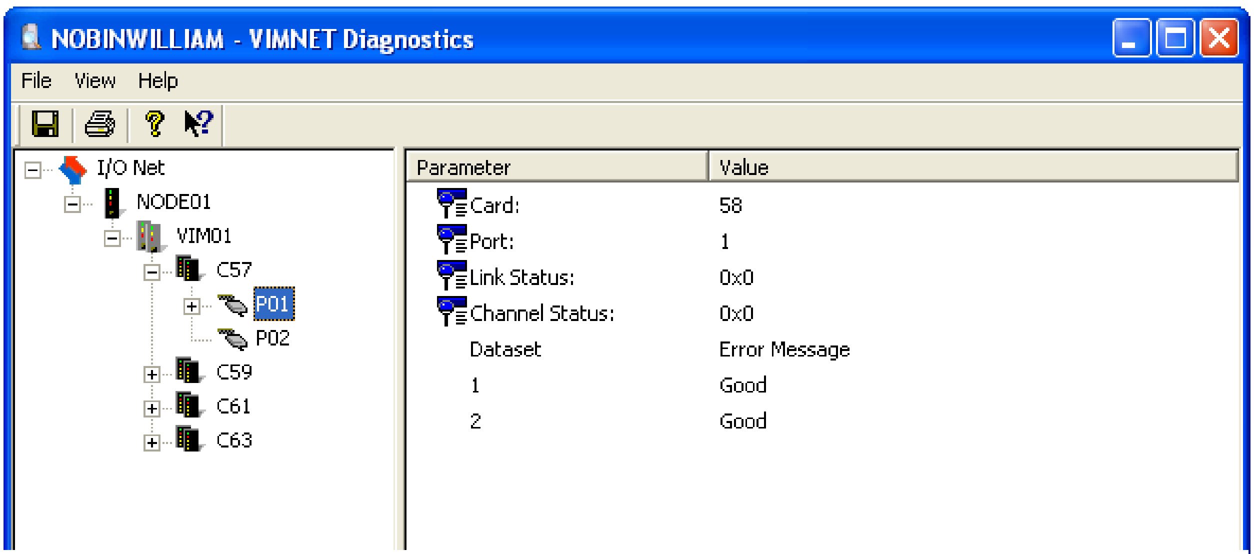


4.2 Port Level Diagnostics

Port level diagnostics show the port status, as well as status of datasets. Dataset status is shown as a character string corresponding to any error which might exist. This same error string is also displayed in DeltaV Diagnostics.

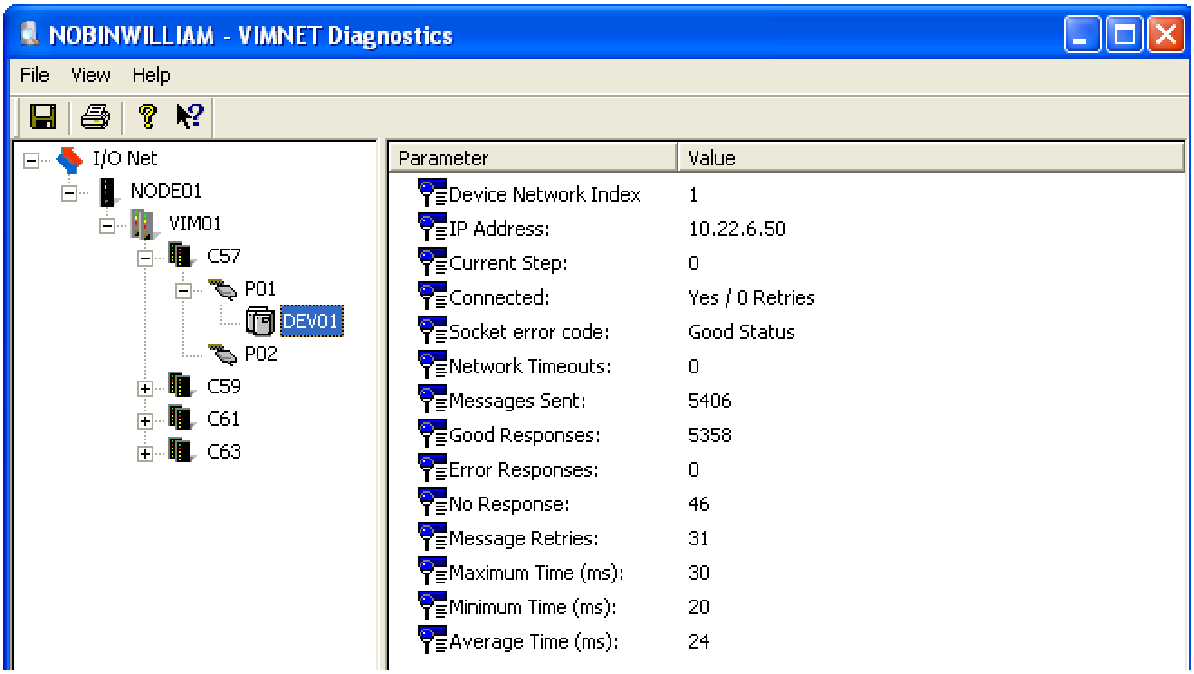
Note that if the serial card is redundant, only the odd numbered card is shown in the left pane. In the right pane, the actual active card is shown depending on Active VIM.

The Link Status and Channel Status are shown as hexadecimal error codes. The error message column contains any error that might exist. If no error exists, then the status shown is Good.

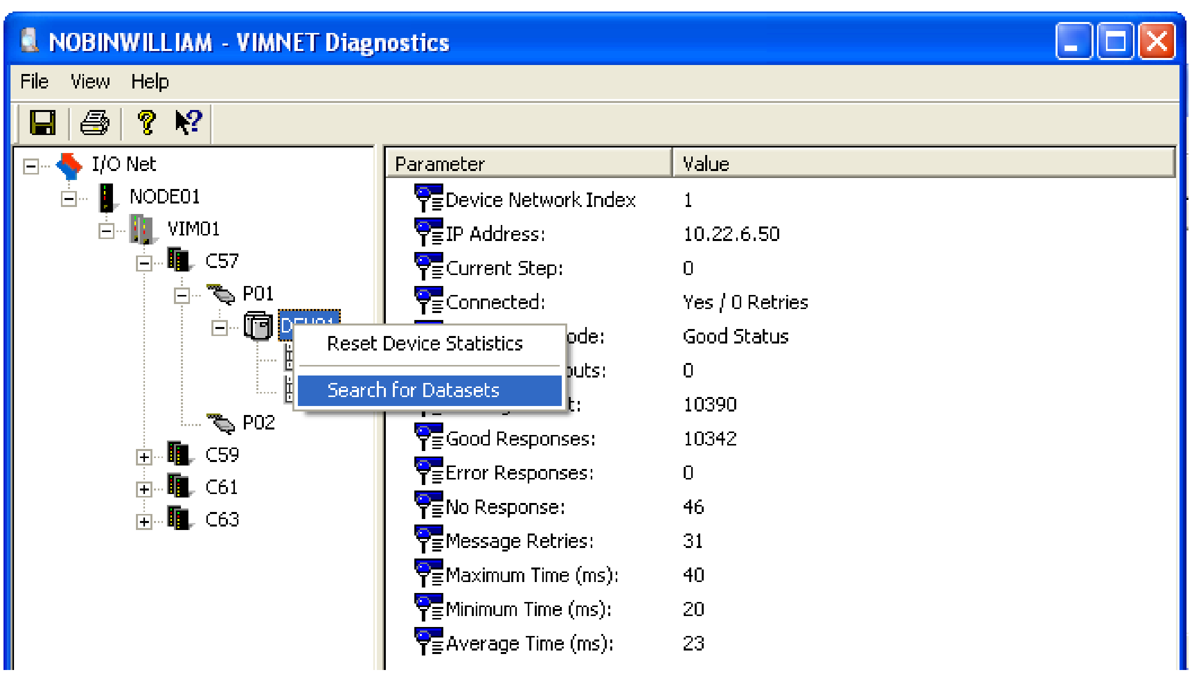


4.3 Device Level Diagnostics

Device level diagnostics show the statistics for selected device as follows:

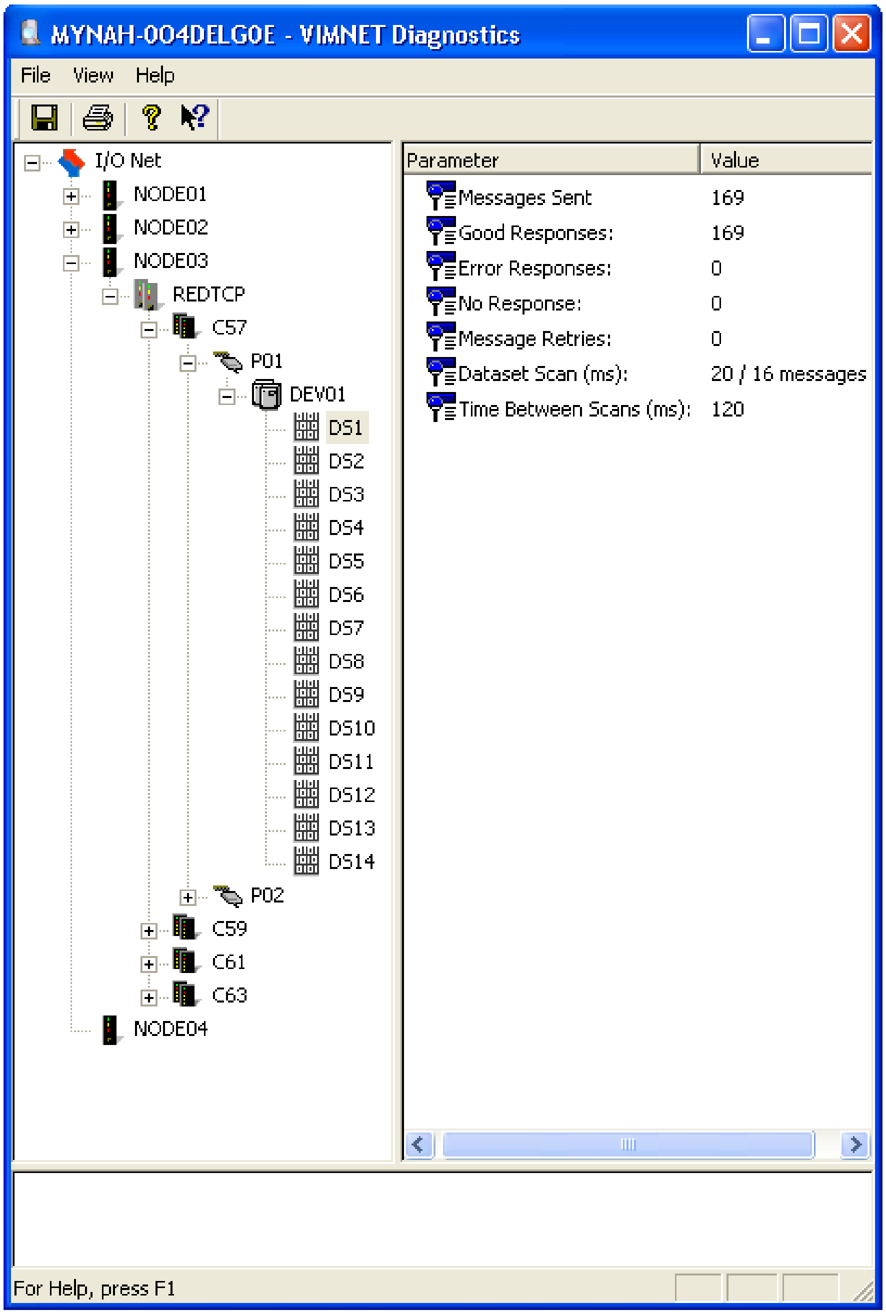


You can right click on the device to get a context menu as follows. This menu allows you to reset the statistics and also to search for configured datasets in this device.



4.4 Dataset Level Diagnostics

By clicking on the individual dataset under Device diagnostics, you will get the dataset specific diagnostic information as follows:

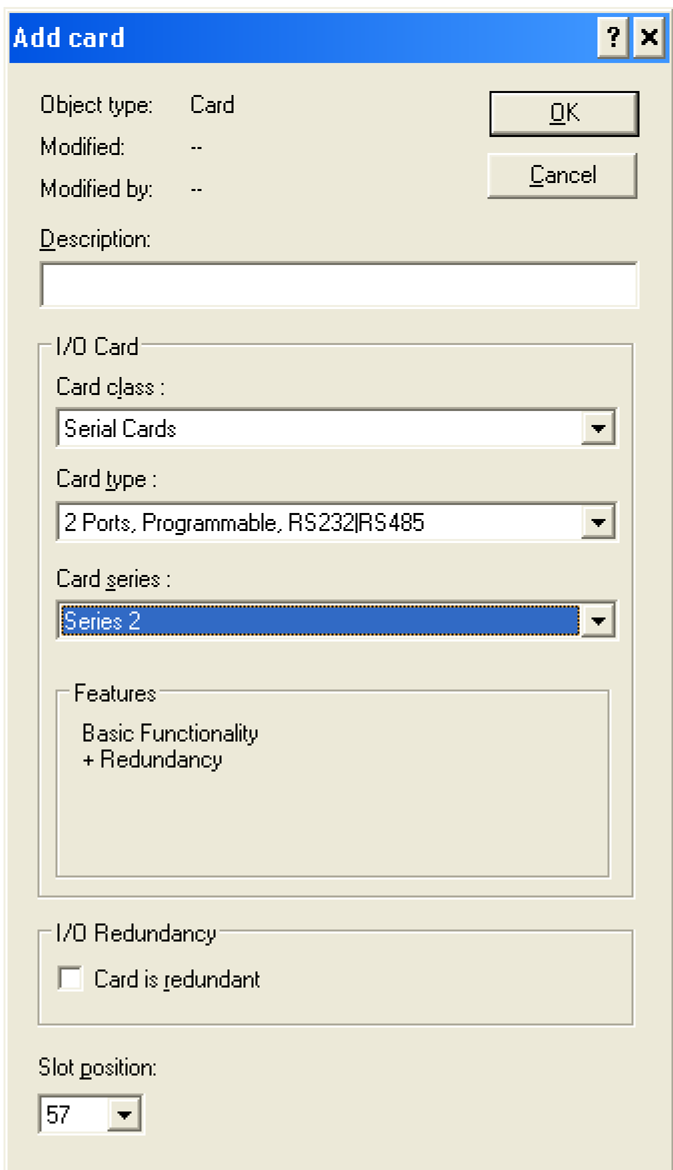


Items of specific interest at this level are the Dataset Scan and the Time Between Scans. These pieces of information tell us what the scan time for this dataset is, and how much time elapses between two consecutive scans.

5.0 Configuring DeltaV

For each VIM module used, four Programmable Serial Cards must be configured in the DeltaV Explorer. A maximum of 2 VIM modules can be used with each DeltaV controller. The simplex serial cards required must be configured in slots 57-60, or 61-64. Redundant serial cards must be configured in pairs in slots 57/58, 59/60, 61/62, and 63/64. To add these cards, follow the steps below. Note that cards can also be added via the DeltaV Explorer, using the Auto-sense I/O cards menu option. All four cards must be configured, even if you are not using all of them. In addition, disable all unused serial card ports.

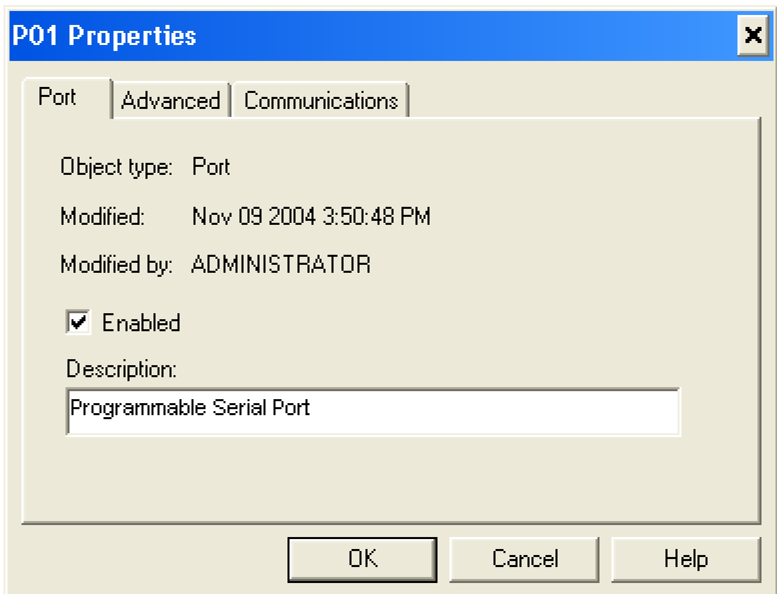
In DeltaV, configure the serial card. This will create a Programmable Serial Card and define 2 ports under it, P01 and P02. Select the Card is redundant Checkbox if you are creating a redundant serial card.



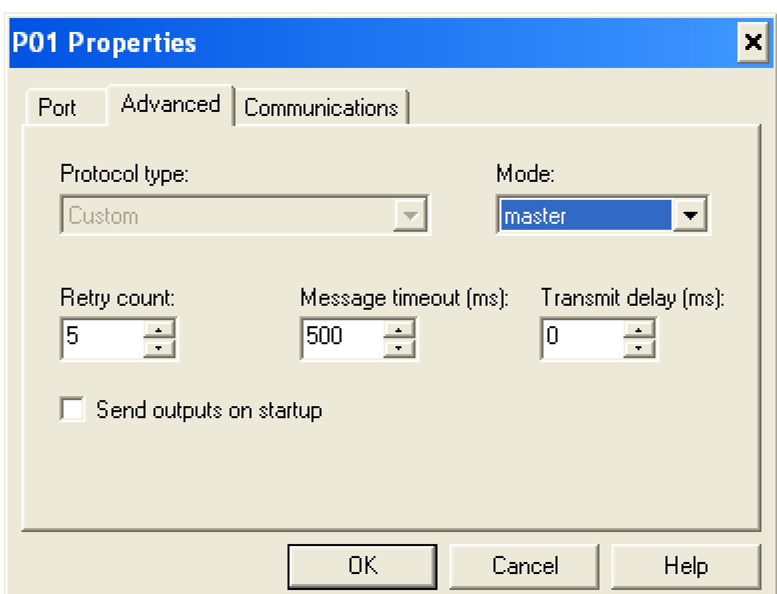
1. Right mouse click on Port 1. The following dialog will appear.

Note

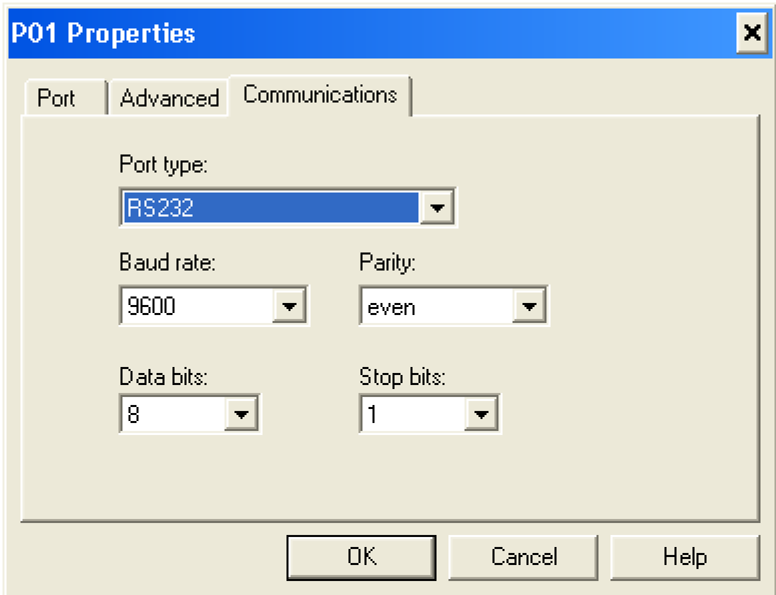
Make sure that you Enable the Port by clicking on the Enabled box. Unused ports should be left disabled.



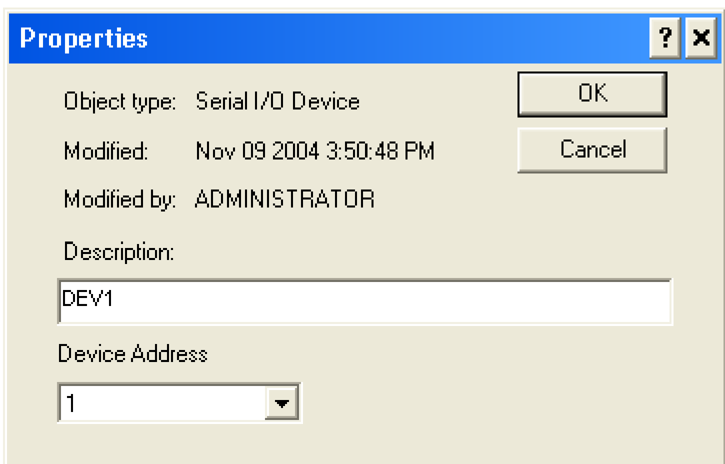
Next, select the Advanced tab. The following dialog will appear. In this dialog, select Master. Also select the message time parameters. All PLC devices configured under a given port will use the same time parameters.



Next, click the Communications tab. The following dialog will appear. These parameters are not used. Simply select the defaults and click OK.



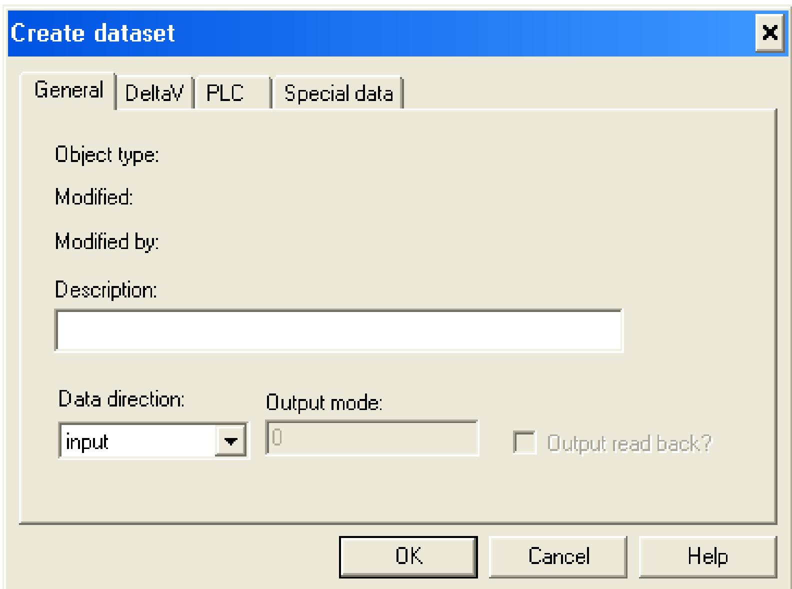
2. Configure a Serial Device under the Port by doing a Right Mouse click and selecting New Serial Device. The following dialog will appear:



Specify the device address and description. Then click OK. This will add the serial device. The Device Address corresponds to the PLC device address.

5.1 Configure Datasets

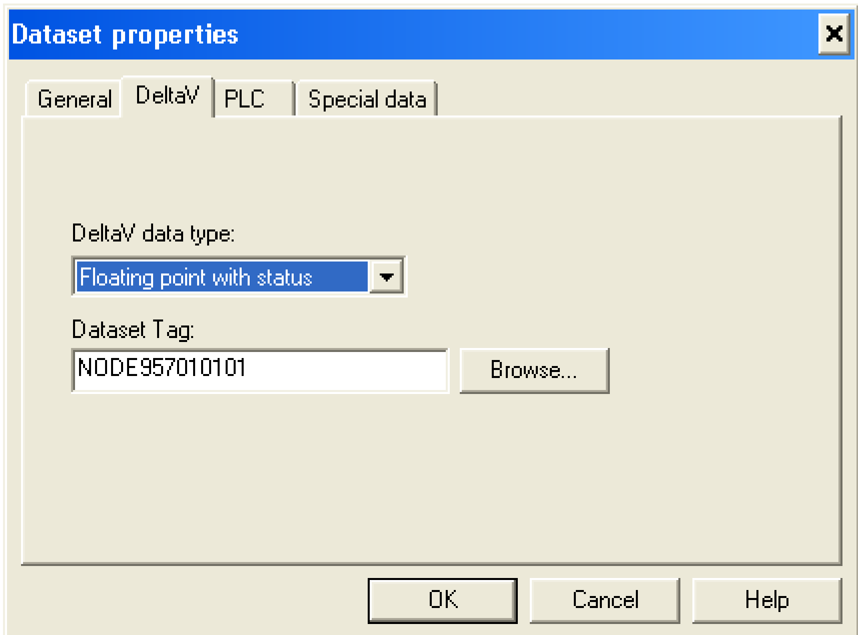
Next, configure datasets in the Serial Device. Each Serial Device can have 16 datasets under it. Or you can have 16 devices with 1 dataset each. A dataset can be input or output. To add a new dataset, right mouse click on the Serial Device and select New Dataset. The following dialog will appear.



Configure the data direction to be input or output. In the above example, we are configuring an input dataset.

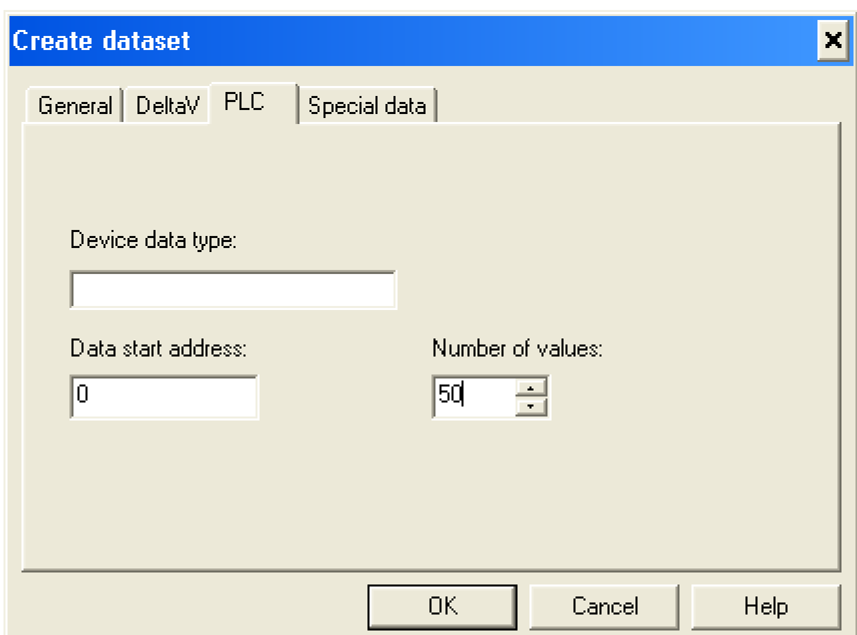
If the dataset direction is Output, you must select the Output mode. Output mode of 0 indicates Block outputs, i.e., the entire dataset is written out to the PLC if any dataset register changes. An Output mode of 1 indicates Single value output, i.e., only the value that has changed will be written out. Output datasets can also be read back from the PLC by selection the Output readback checkbox. Note that for Output datasets with readback, pending output changes always have precedence.

Next, click the DeltaV tab. The following dialog will appear:



In this dialog, configure the data type needed for DeltaV. You can see the available types by clicking on the drop down list. In the above example, we are configuring the input data type to be floating point. Please see Section 4.1 for additional details for this parameter.

Next click the PLC tab. The following dialog will appear.



In this dialog, we will map DeltaV data types to PLC (or external Device) data types. PLC data type values and corresponding PLC registers are:

Device Data Type	Device Register
0	COILS
1	INPUT STATUS
2	INPUT REGISTERS
3	HOLDING REGISTERS
6	VIM STATISTICS
All other values	Reserved

Table 4: PLC Data Type Values and Registers

The Start Address specifies where in the PLC to read the data. In this example, the starting address is 0. This can be any PLC specific address.

The following table describes examples of PLC registers and corresponding start addresses in the DS:

PLC address	Start Address	Description
1	0	First coil in PLC
501	500	PLC coil #501
400001	0	First holding register
400950	949	Only the offset for register is specified

Table 5: PLC Registers, Start Addresses, and Descriptions

In this example, the starting address is 0, the PLC data type is 3 and the maximum number of values is 50. This would result in the following DeltaV registers:

DeltaV Register	PLC or External Device Registers
R1	0 1
R2	2 3
R3	4 5
R50	99 100

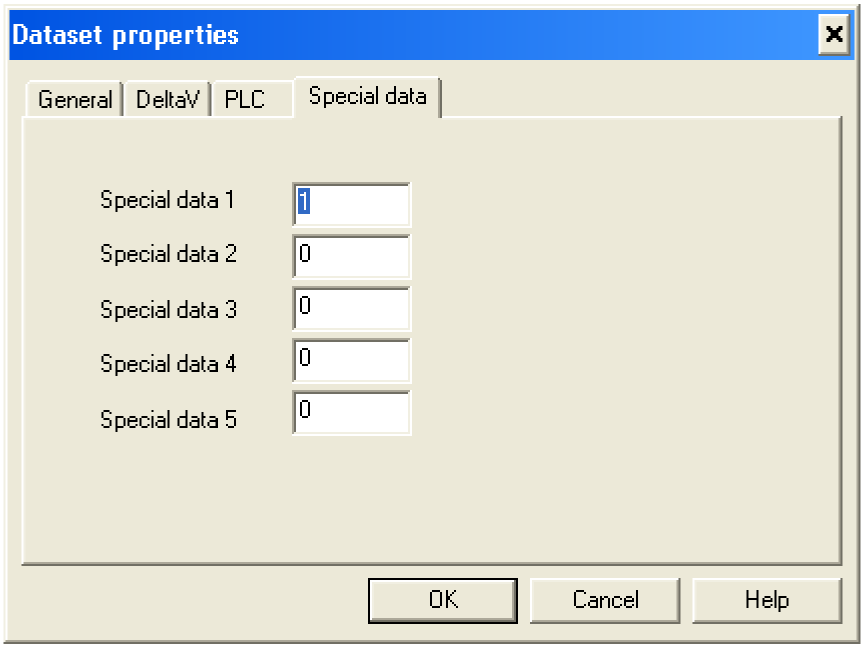
Table 6: DeltaV and PLC Registers

Each dataset has a maximum of 100 16-bit values of DeltaV data type (as configured in the previous dialog). Each DeltaV value is mapped to one or more PLC registers. If two registers are required, as is the case for floating point values, the registers must be consecutive. The following table describes the mapping:

DeltaV Data Type	**PLC Register Type**	**PLC Registers Required**	**Max Number of Values**
Boolean	Coils Input Status	1 – Bit	100
Discrete	Coils Input Status	1 – Bit	100
Signed 8 bit Integer	Coils Input Status	1 – Bit or Byte	100
Signed 16 bit Integer	Coils Input Status Input Registers Holding Registers	1 - 16 bit word	100
Signed 32 bit Integer	Holding Registers	2 - 16 bit words	50
Unsigned 8 bit Integer	Coils Input Status	1 - 16 bit word	100
Unsigned 16 bit Integer	Coils Input Status Input Registers Holding Registers	1 - 16 bit word	100
Unsigned 32 bit Integer	Holding Registers	2 - 16 bit word	50
Floating Point	Holding Registers	2 - 16 bit word	50
String	Holding Registers	1 byte	100

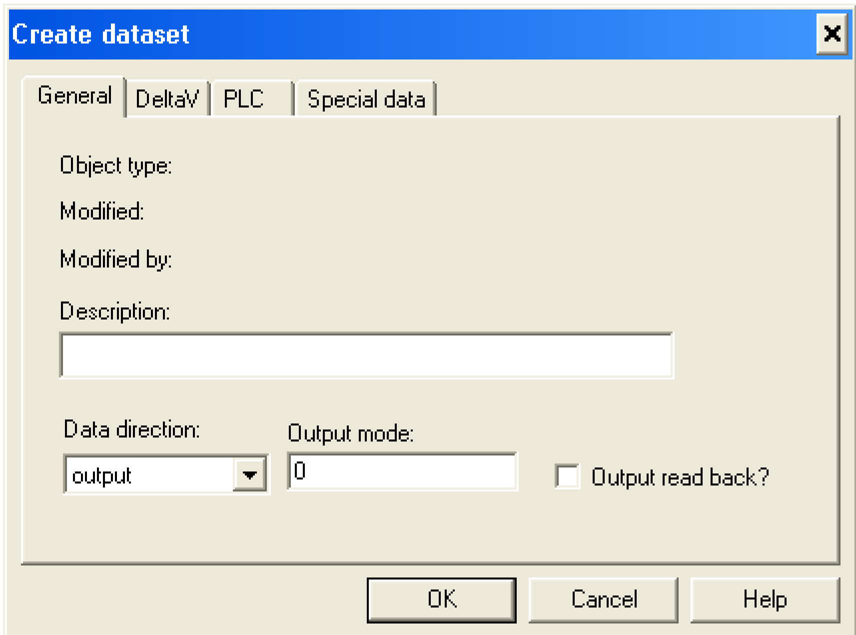
Table 7: Dataset Specification

3. Lastly, for each dataset click on the Special data tab. The following dialog will appear:

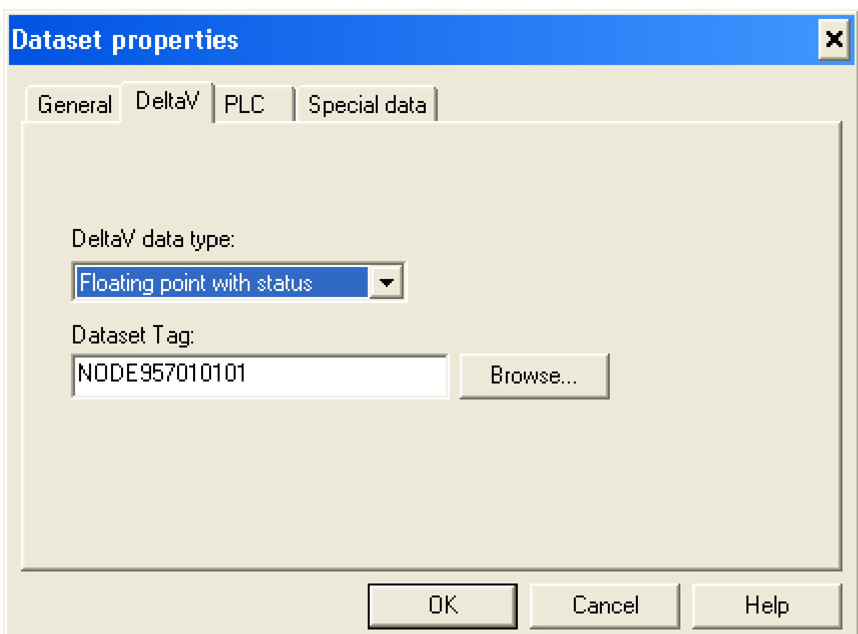


Special data 1 value is used when transferring data for Floating point, signed 32-bit integer and unsigned 32-bit integer registers. Special data 2 is used to indicate the number of registers used for Floating point and 32-bit integer values. Details of Special data usage are provided in the Customization section 5.4.

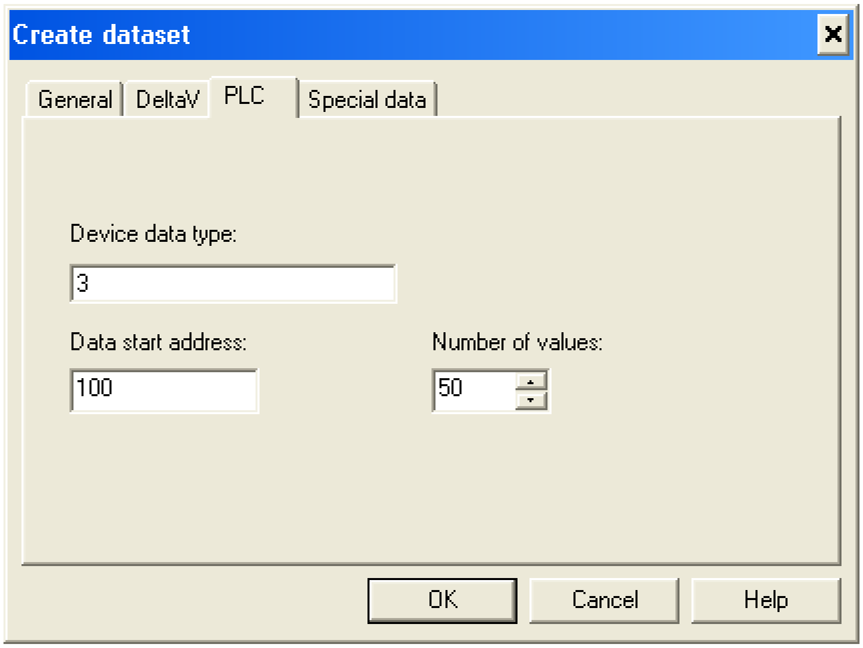
Next, configure an output dataset in the Serial Device. Repeat Step 4 above. The following dialog will appear.



Configure the data direction to be output. Next, click the DeltaV tab and select DeltaV data type as floating point.



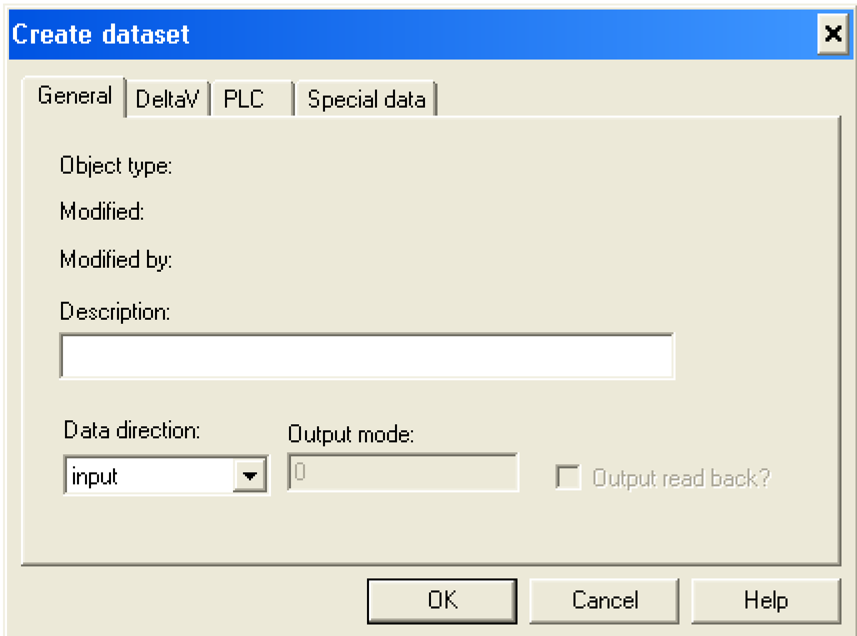
Next, click the PLC tab. The following dialog will appear.



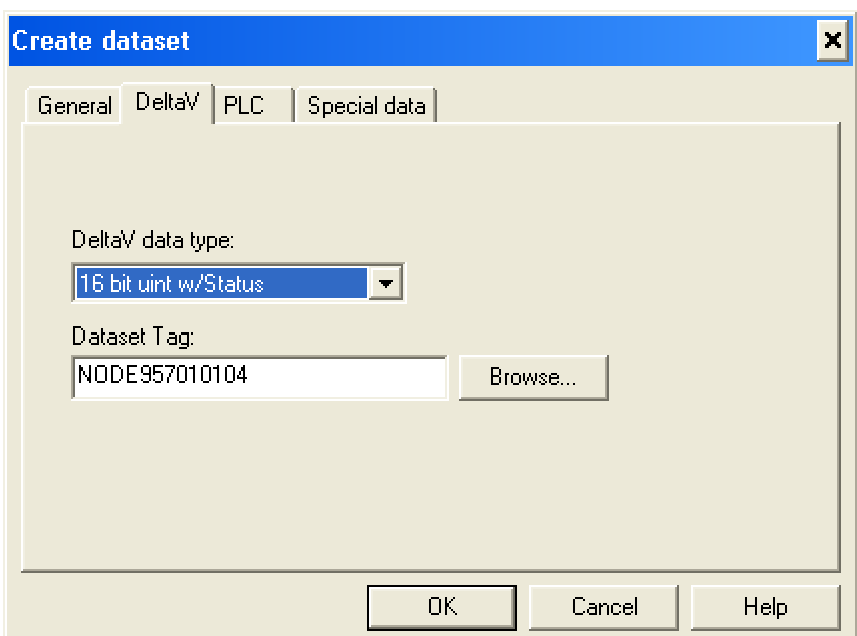
Note that the Data start address is now 100. This is because in our example, the first dataset has 50 floating-point values. Consequently, this dataset starts at R101.

5.2 Configuring a dataset of type 16-bit Unit with Status

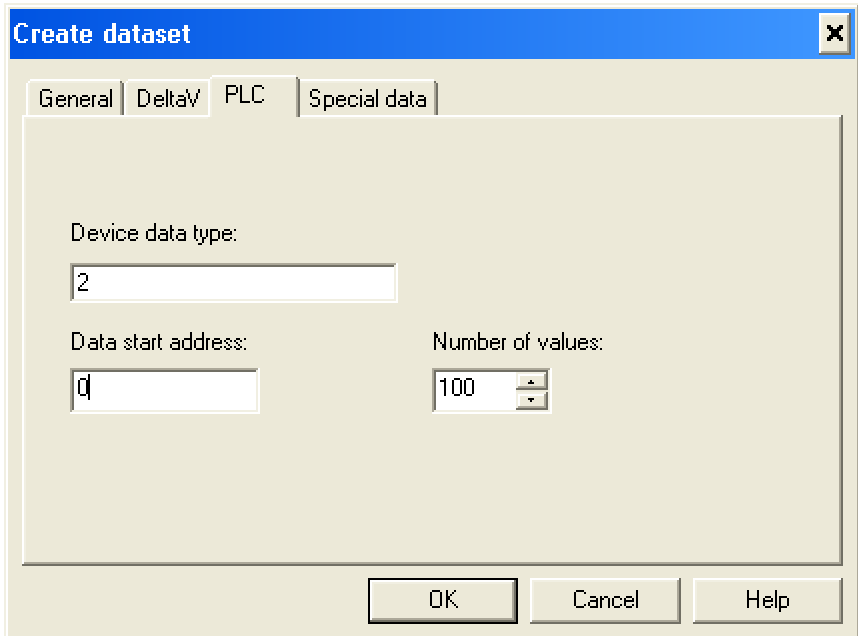
Follow steps one through four as above. You will be back to the dialog below:



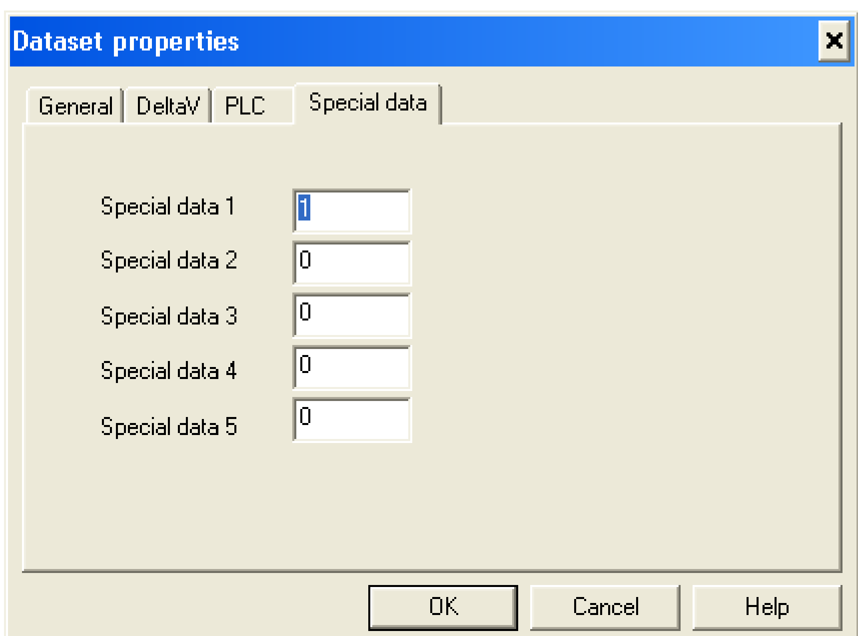
Configure the data direction to be input or output. Click on the DeltaV tab as show below. View the drop down menu and choose 16-bit Unit with status. Click OK.



Click the PLC tab and configure Device data type. To read Input Registers or Holding Registers, the Device data type would be 2 or 3, respectively. The maximum number of values will be 100.

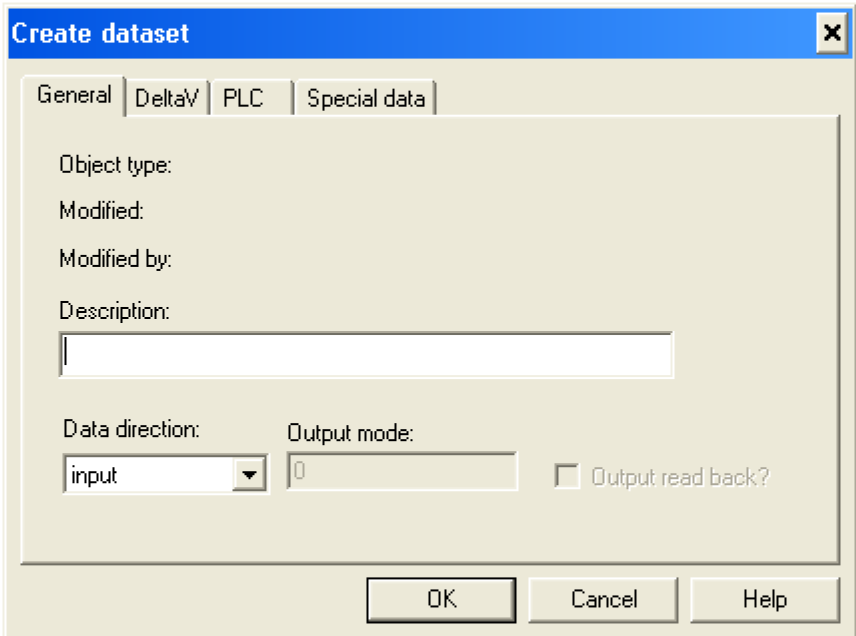


For each dataset, click on Special data and the following dialog will appear. If the byte order of 16-bit values read from the field must be swapped, then configure Special data 1 = 1. If byte swapping is not required, configure a 0 in the registers. Click OK to close the dialog.

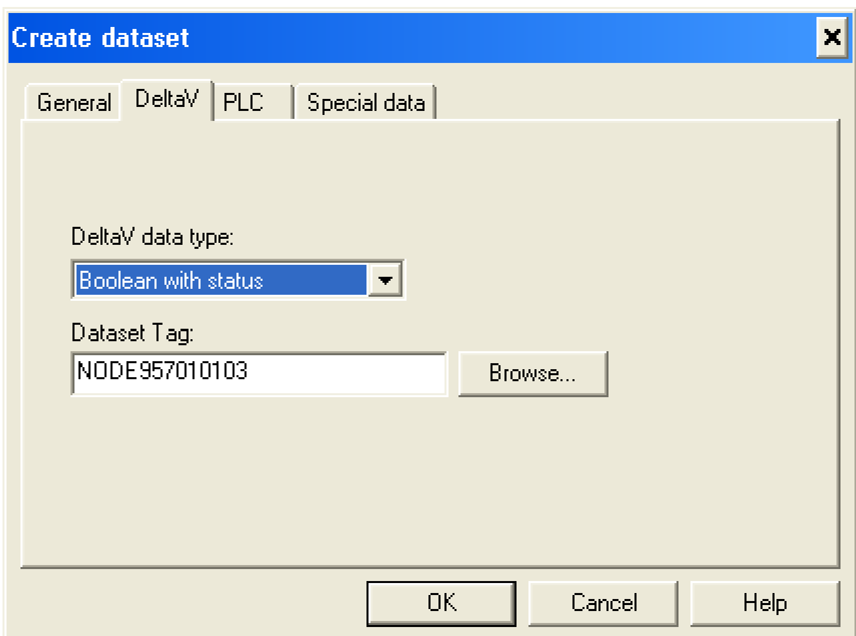


5.3 Configuring a dataset of type Boolean or Discrete with status

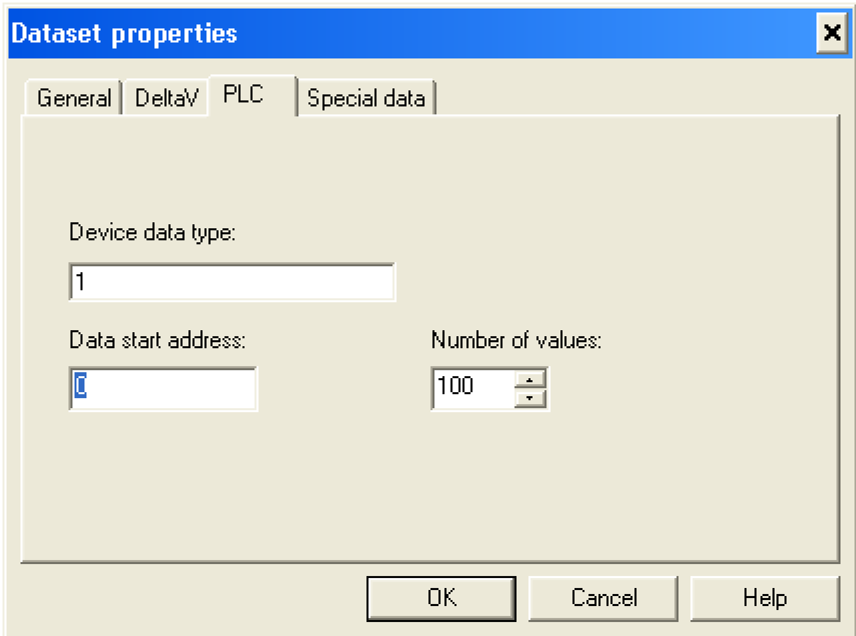
Follow steps one through four as above. You will be back to the dialog below:



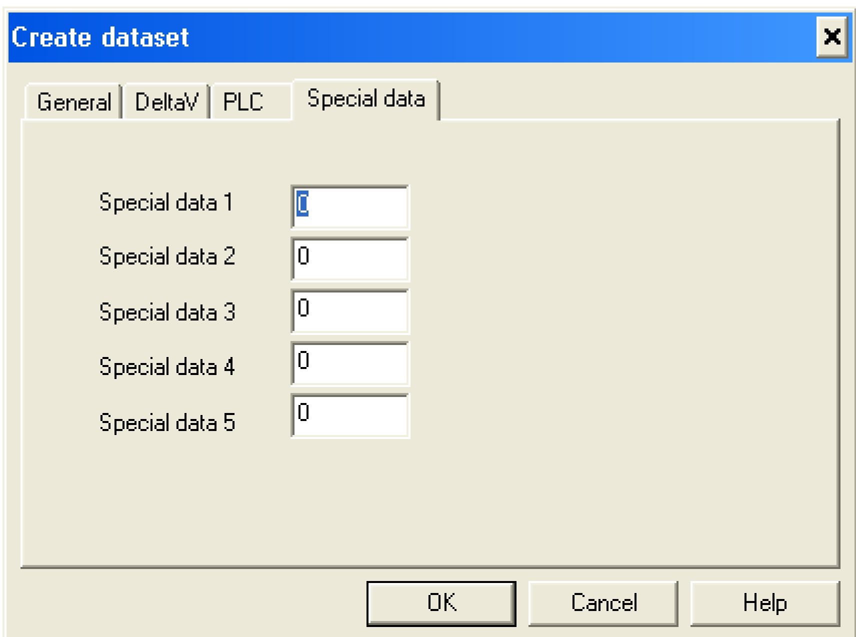
Configure the data direction to be input or output. Click on the DeltaV tab as show below. View the drop down menu and choose Boolean with status (or Discrete with status). Click OK.



Click the PLC tab and configure Device data type. To read Coils or Input Status Registers, the Device data type would be 0 or 1, respectively. The maximum number of values will be 100.

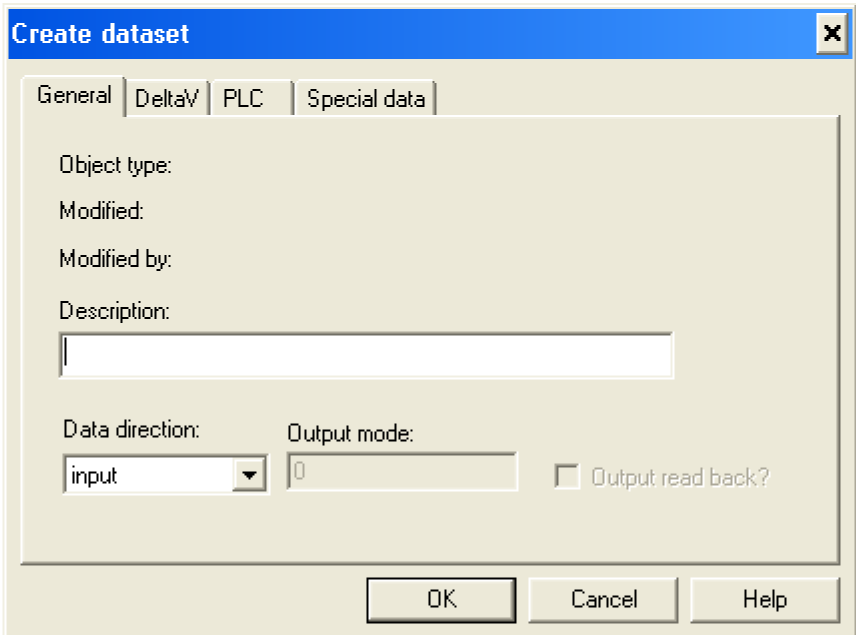


Special data values are not used for Boolean and Discrete datasets. Click OK to close the dialog.

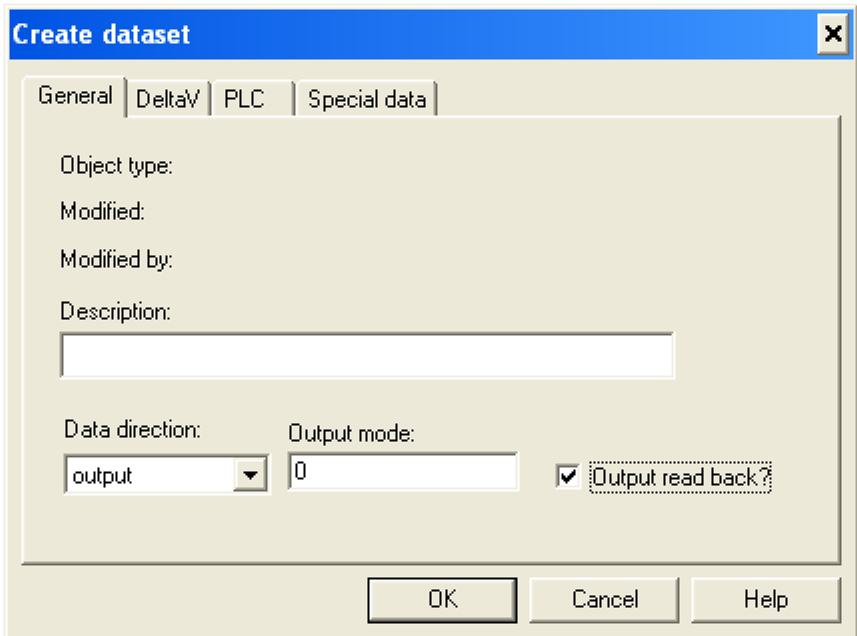


5.4 Configuring a dataset for VIM Diagnostics

Follow steps one through four as above. You will be back to the dialog below. Select the Data direction as Input. Please see Section 4, Table 3 for diagnostic data available in this dataset.

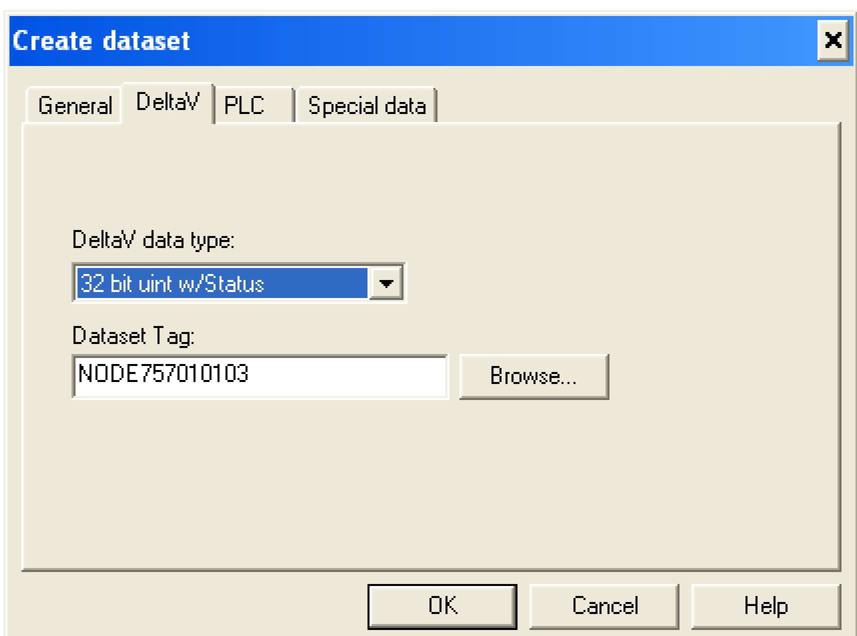


A special case of this dataset is to allow user control of redundancy switching at the DeltaV Control Module level. This capability is enabled only if the VIM is configured as redundant. In this case, this dataset should be configured with Data direction as Output, and the Output read back checkbox should be checked as follows:

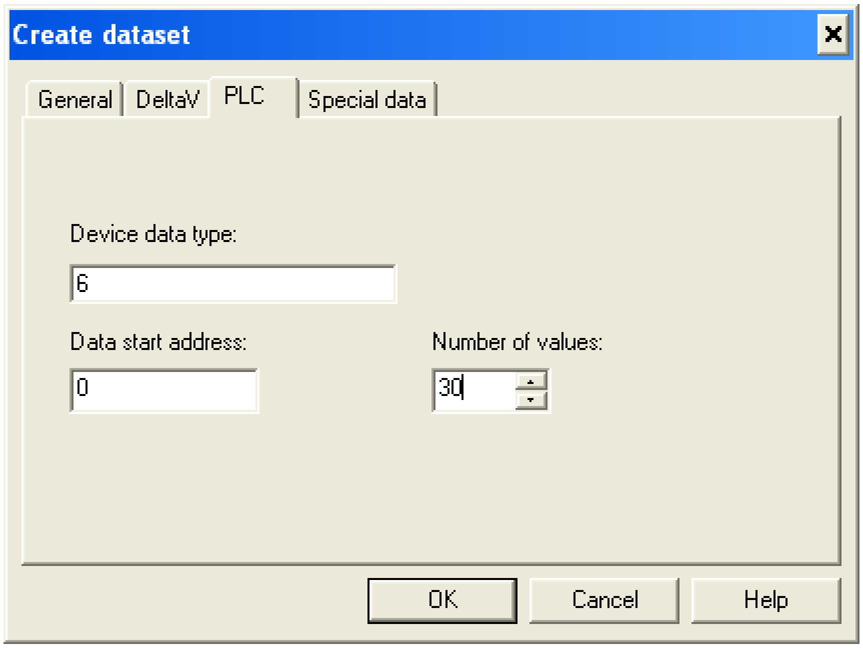


Configure the data direction to be input or output, as required for your application.

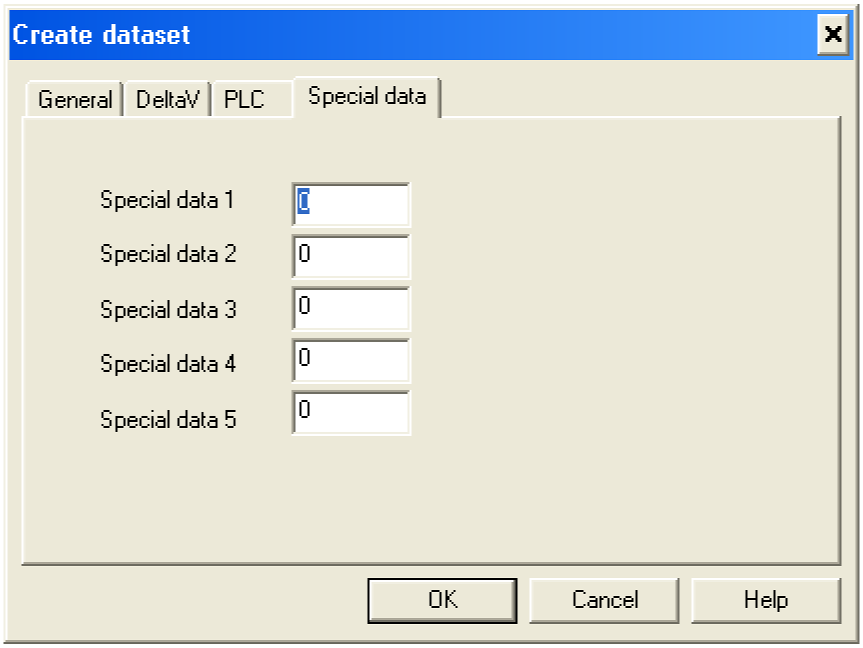
Next, click on the DeltaV tab as show below. View the drop down menu and choose 32 bit unit w/Status. Click OK.



Click the PLC tab and configure Device data type. For VIM Diagnostics, enter the Device data type as 6. The Data start address will always be 0, and the Number of values will be 30.



Special data values are not used for this dataset. Click OK to close the dialog.



5.5 Customization

The VIM firmware allows you to customize ModbusTCP communications and representation of data. Modification of data representation is typically only required when reading/writing Floating Point, or Signed/Unsigned 32-bit Integer registers. However, in some cases, you may need to use Special data 1 for 16-bit byte swapping as well.

To customize data representation in a data set, you can use the Special data 1 and 2 registers as flags. To customize communications for a specific data set, the Special data 3 and 4 registers are used. This is described below.

Assume a Floating-point number 123.45, and its representation in IEEE 754 format as follows:

Floating Point Number	**Representation as 2 16-bit words**	**Representation as 4 bytes**
123.45	58982, 17142	230, 102, 66, 246

Special Data	**Value and Description**
1	0 – Default is no customization – Floating Point and 32-bit data is represented as received. The transmitted byte order is 230, 102, 66, 246 1 – The transmitted byte order is 66, 246, 230, 102 2 – The transmitted byte order is 246, 66, 102, 230 3 – The transmitted byte order is 102, 230, 246, 66
2	0 – Default setting where 2 Modbus 16-bit registers are equivalent to 1 Floating Point or 32-bit value in DeltaV. There is a 1-2 correspondence between DeltaV value and read values. 1 – Data is not read as 2 MODBUS 16-bit registers but as individual Floating Point or 32-bit values. There is a 1-1 correspondence between DeltaV value and read value.
3	This is a delay parameter specified in 5ms units. When specified, it overrides the port Transmit delay for this dataset. The driver will delay for this amount of time before reading/writing the dataset.
4	This is a delay parameter specified in 5ms units. It applies only to datasets configured as Output, and Output mode 0 (block output). When specified, it provides a cyclic write of current dataset values from the VIM database to external devices. Note that normal functionality is for the VIM to write output data only when a change of state is detected.
5	Not Used

Table 8: 32-bit Data Byte Order

6.0 Redundant I/O Communications

Four types of communications connections are supported for redundant VIMs. These are described below. For each field device connected to the VIMs, the corresponding redundancy type must be configured in the VIMNet Explorer. This is described in Section 3.5, Step 5.

6.1 Simplex Field Device

In this case the field device is non-redundant. It only has a single network connection available and consequently a single IP address. Both the Active and Standby VIMs will communicate with the same IP address. Connect the field device to either one of the isolated switches. The Active VIM will perform actual data I/O communications. The standby VIM will send a periodic “ping” to the same IP address. The ping allows the standby VIM to ensure that the communication path is valid. If the standby VIM cannot verify network path validity, an error message will be generated back to DeltaV indicating Standby problems and switchover will not be available. If the standby path is valid, you can command VIM switchover using DeltaV Diagnostics. Note that the “ping” comprises the Modbus Diagnostics command. This command uses Function 8, Sub-Function Hi=0, Sub-Function Lo=0, and 2 bytes of data. The normal response to this command is to loop back the same data.

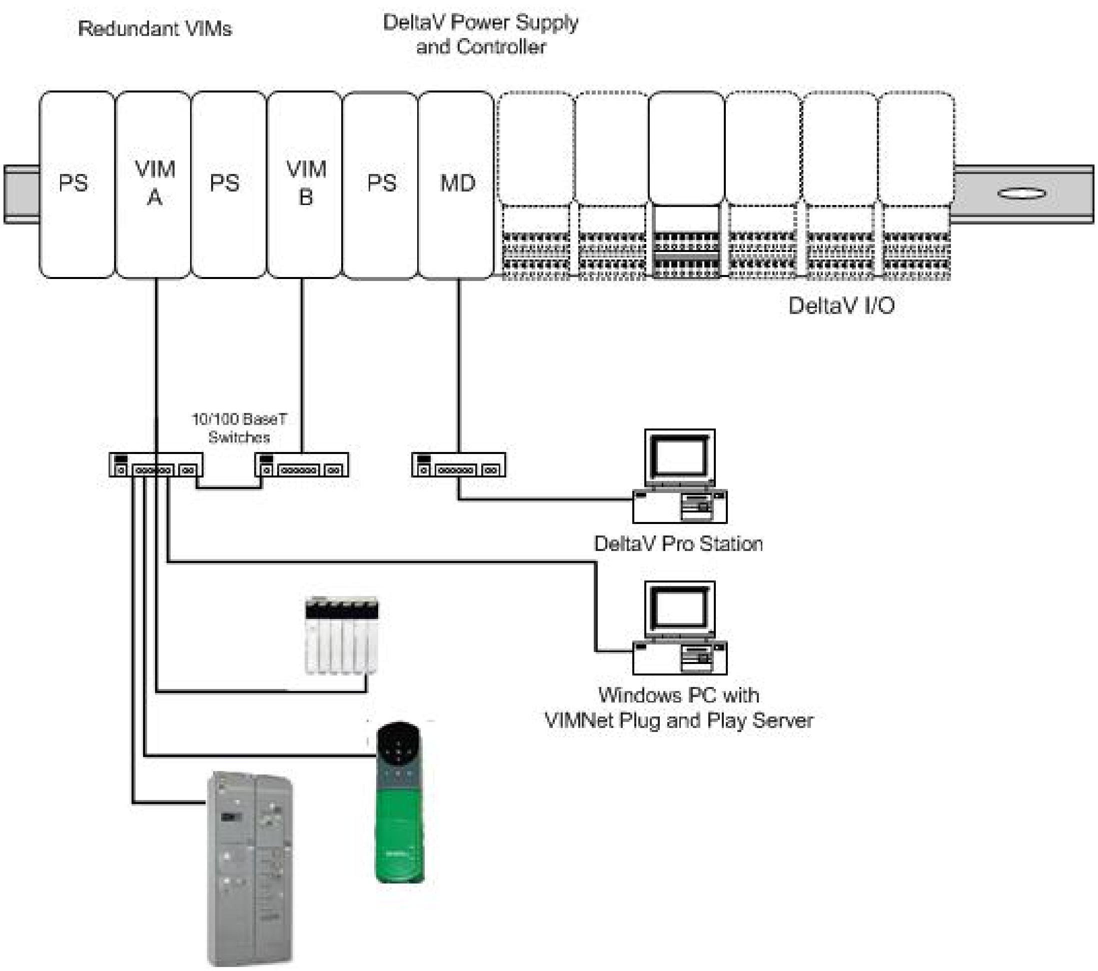


Figure 5: Redundant VIMs with Simplex Modbus Devices

6.2 Redundant Field Device with Single Chassis

In this case the field device has a single chassis but uses two network interface cards for communications with the VIMs. Connect each network card to a separate isolated switch corresponding to the VIM. The IP address of these network cards can be consecutive, e.g., 10.22.6.50 and 10.22.6.51. No IP switching is expected or performed.

VIM A will always communicate with the first IP address 10.22.6.50, and VIM B will always communicate with 10.22.6.51. If a VIM is in Standby, it will send a periodic “ping” to its assigned IP address. The ping allows the standby VIM to ensure that the communication path is valid. If the standby VIM cannot verify network path validity, an error message will be generated back to DeltaV. The error message will indicate the Standby problems, and switchover will not be available. If the standby path is valid, you can command VIM switchover using DeltaV Diagnostics. Note that the “ping” comprises the Modbus Diagnostics command. This command uses Function 8, Sub-Function Hi=0, Sub-Function Lo=0, and 2 bytes of data. The normal response to this command is to loop back the same data.

If the Active VIM loses communications and the Standby path is valid, the VIM will automatically request a switchover to its partner. The DeltaV controller verifies that the switchover is possible and then commands the currently Active VIM to go Standby, and at the same time commands the currently Standby VIM to go Active. Both VIMs maintain current DeltaV outputs. On switchover, the new Active VIM immediately starts scanning the field inputs to update its internal database. The following illustrates the network for this level of redundancy:

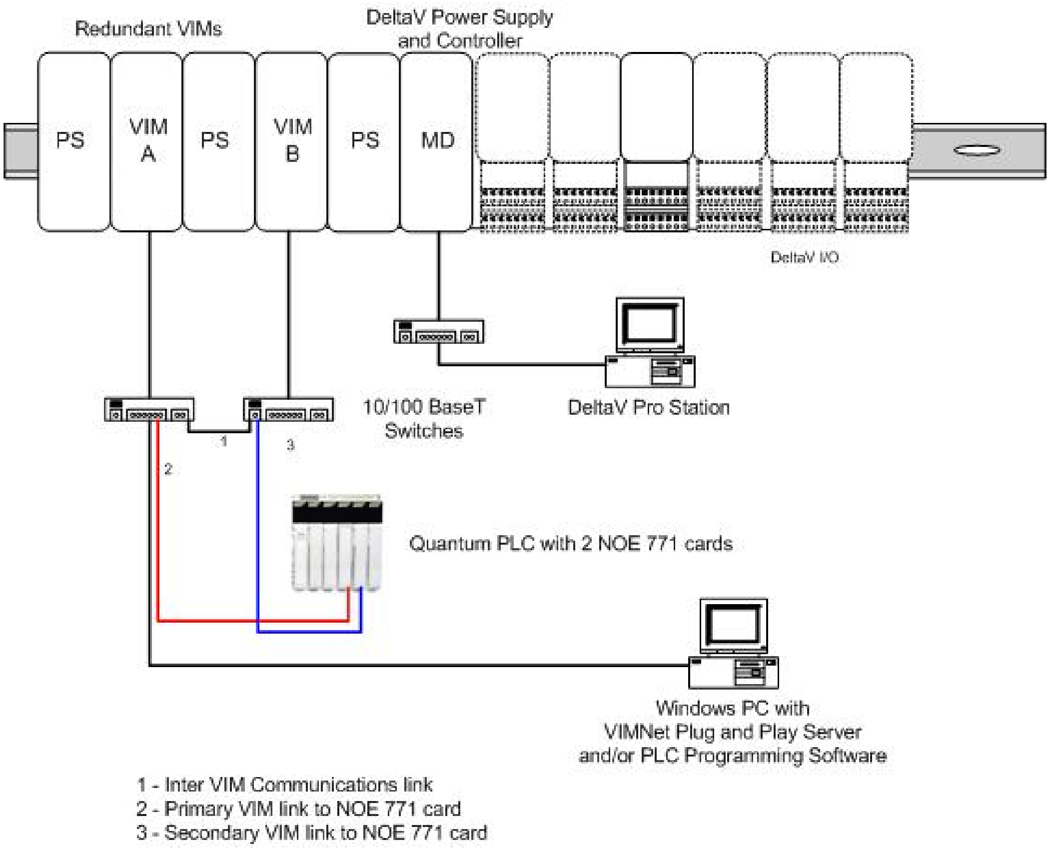


Figure 6: Redundant VIMs with Redundant PLC Network Connections

The operational states in this level of redundancy are as follows:

Scenario 1 Operating Conditions:

- VIM A is active
- VIM A is scanning 10.22.6.50
- VIM B is pinging 10.22.6.51

VIM A State	VIM B State	Redundancy State
Good	Good	VIM A stays active
Good	Bad	VIM A reports standby problems – switchover unavailable
Bad	Good	VIM A requests a switchover to VIM B

Table 9: Non-switching IP, VIM A Active

Scenario 2 Operating Conditions:

- VIM B is active
- VIM B is scanning 10.22.6.51
- VIM A is pinging 10.22.6.50

VIM A State	VIM B State	Redundancy State
Good	Good	VIM B stays active
Bad	Good	VIM B reports standby problems – switchover unavailable
Good	Bad	VIM B requests a switchover to VIM A

Table 10: Non-switching IP, VIM B Active

6.3 Redundant Field Device with Dual Chassis

In this case the field device has two chassis that behave as a redundant pair. There is one network card in each chassis for communications with the VIMs. Connect each network card to a separate isolated switch corresponding to the VIM. The IP address of these network cards must be consecutive, e.g., 10.22.6.50 and 10.22.6.51. It is expected that the IP address will switch between both chassis when a chassis switchover occurs. For example, if 10.22.6.50 is used, it will always be the ACTIVE IP address regardless of which chassis is active. This is also known as Hot Standby.

The Active VIM communicates with the IP address in the Active chassis. This is expected to be the first IP address. The Standby VIM sends a periodic “ping” to the IP address in the Standby chassis (second IP address). The ping allows the standby VIM to ensure that the communication path is valid. If the standby VIM cannot verify network path validity, an error message will be generated back to DeltaV indicating Standby problems and switchover will not be available. If the standby path is valid, you can command VIM switchover using DeltaV Diagnostics. Note that the “ping” comprises the Modbus Diagnostics loopback data command.

If the Active VIM loses communications and the Standby path is valid, the VIM will automatically request a switchover to its partner. The DeltaV controller verifies that the switchover is possible and then commands the currently Active VIM to go Standby, and at the same time commands the currently Standby VIM to go Active. Both VIMs maintain current DeltaV outputs. On switchover, the new Active VIM immediately starts scanning the field inputs to update its internal database. If the field device chassis switches from Active to Standby, it will assume the IP address of the previous Standby. Simultaneously, the current Active will assume the IP address of the previous Active. Note that this switchover will not result in VIM switchover. This is because the VIM is still communicating with the same Active IP. The VIM will switch only if it loses communications with the Active chassis and the standby path is valid.

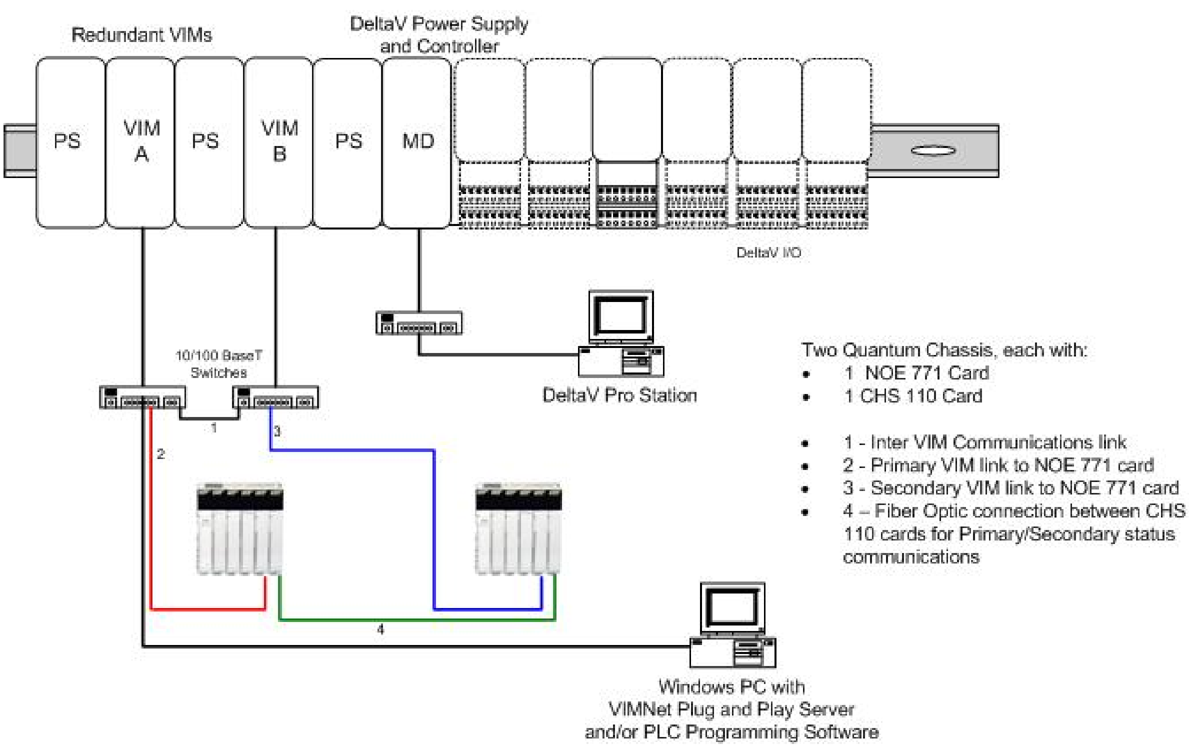


Figure 7: Redundant VIMs with PLC configured as Hot Backup

The operational states in this level of redundancy are as follows:

Scenario 1 Operating Conditions:

- VIM A is active
- VIM A is scanning 10.22.6.50
- VIM B is pinging 10.22.6.51

VIM A State	VIM B State	Redundancy State
Good	Good	VIM A stays active
Good	Bad	VIM A reports standby problems – switchover unavailable
Bad	Good	VIM A requests a switchover to VIM B

Table 11: Switching IP, VIM A Active

Scenario 2 Operating Conditions:

- VIM B is active
- VIM B is scanning 10.22.6.50
- VIM A is pinging 10.22.6.51

VIM A State	VIM B State	Redundancy State
Good	Good	VIM B stays active
Bad	Good	VIM B reports standby problems – switchover unavailable
Good	Bad	VIM B requests a switchover to VIM A

Table 12: Switching IP, VIM B Active

6.4 User Application Initiated Redundant Switchover

As described above, under normal operations, the Active VIM scans the primary IP address, while the Standby VIM "pings" the backup IP address. If the Active VIM loses communications and the Standby path is valid, the VIM will automatically request a switchover to its partner. This request for switchover is sent to the DeltaV controller. The DeltaV controller verifies that the switchover is possible. Then, it commands the currently Active VIM to go Standby and commands the currently Standby VIM to go Active. Consequently, the redundancy switchover is strictly driven by field communications.

In some cases, it is desirable to allow the user application to force a VIM switchover. This firmware supports this capability by using the VIM Diagnostics dataset. Please see Section 5.4 on how to configure the Diagnostics dataset, which can also be used for the switchover command.

If the VIM Diagnostics dataset is configured as Output with Output read back, all VIM Diagnostics will be sent up normally. In addition, a DeltaV Control Module can be configured to write a value (any value) to any register in this dataset. The VIM will interpret this write command as a user-triggered request for switchover and take the appropriate action. As in the normal operating case, the request will be sent to the DeltaV controller, and the controller will initiate the switch.

6.5 Hot Replacement of faulty Redundant VIM

During normal operation, a redundant VIM pair is in continuous communication with each other. This link is achieved over the network using an interconnecting cable between two switches as shown below in red.

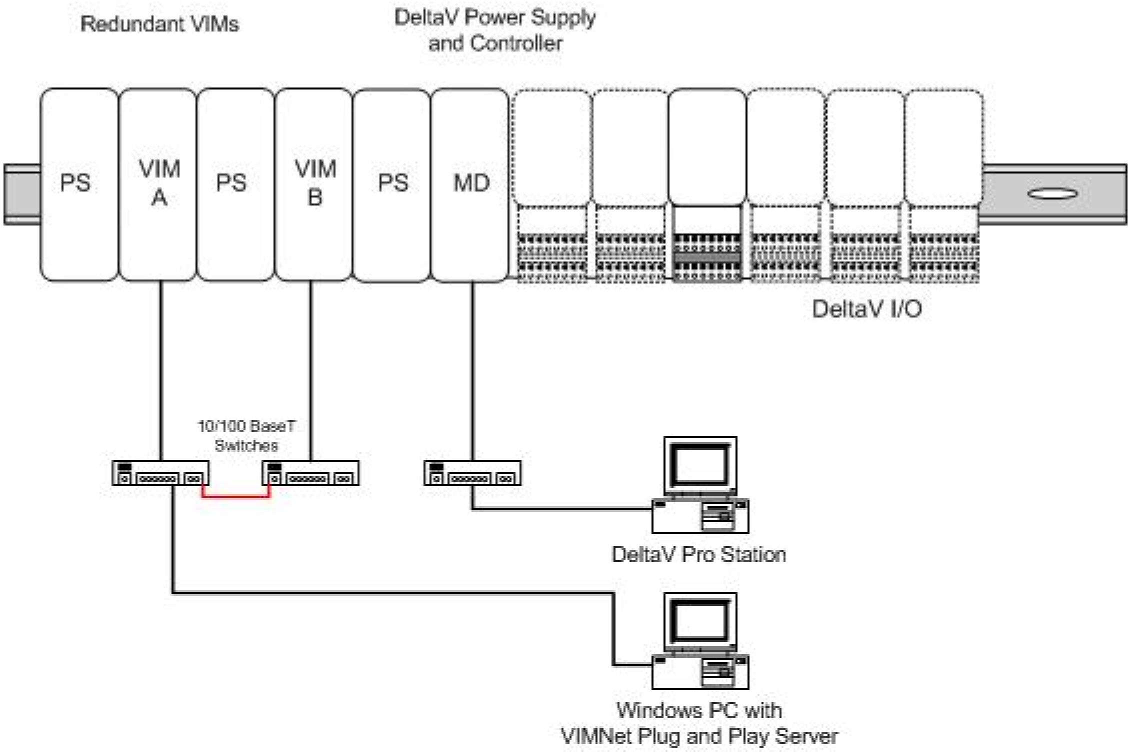


Figure 8: Redundant VIM Network

This link is crucial to the health of the redundant VIM network. Without it, each VIM is blind to its partner's state. If there is a VIM failure, the redundant partner will detect that there is a standby problem. Furthermore, this network link is used by the Active VIM to commission and configure a replacement VIM as described below.

If a VIM fails, the DeltaV Controller will immediately send a Go Active command to the partner VIM. The dead VIM will appear in one or more of the following ways:

1. The red Fault LED will be ON. The state of other LEDs is not significant if the fault LED is on.

2. The emulated serial cards will not be present in DeltaV Diagnostics. All Odd or Even number cards will appear as Configured but not present, depending on which VIM failed. By convention, VIM A handles all odd cards, and VIM B handles all even cards.

3. The active VIM will show that the standby is not communicating, and rebooting the dead VIM does not clear the problem.

4. One or more of the 4 tickers displayed in the VIMNet Diagnostics application are not counting, even after restarting the diagnostics application.

In these cases, the redundant VIM firmware allows for automated recovery from failed VIM hardware. The steps to recover are as follows:

1. Replace the dead VIM with a decommissioned VIM. The replacement VIM must already be flashed to the same firmware revision number and application type.

Caution

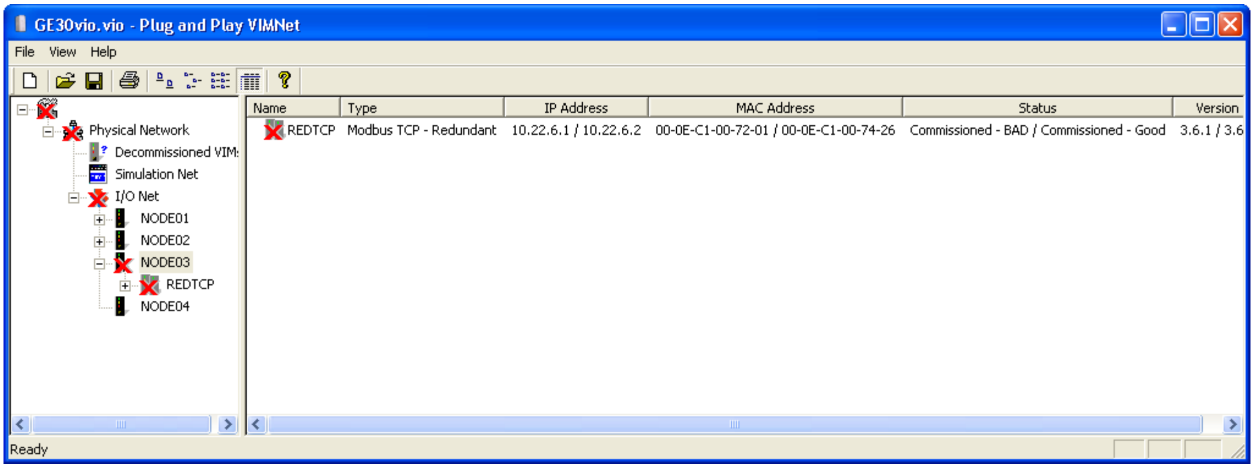
Never replace a dead VIM with an already commissioned VIM. The replacement must be decommissioned. Furthermore, the replacement VIM must be of the same firmware revision and application type.

Failure to follow these rules will cause disruption in field communications and on the Railbus.

For assistance or more information regarding these rules, please contact Mynah Technical Support.

2. Verify that the VIM network inter-communication link is in place.
3. Confirm that there is going to be only one decommissioned VIM on this sub-net. If the Active VIM detects more than one decommissioned VIM, the auto recovery will be aborted. The Active VIM will reset its list of detected VIMs and restart the detection process.
4. If during a 30-40 second period, one and only one decommissioned VIM is detected that matches the revision number and application type, the Active VIM will send a commissioning command to it. The IP address used will be the same as the removed dead VIM.
5. After the commissioning command is processed by the new VIM, it will reboot. After reboot, normal communications will begin between the redundant pair. The new VIM will report to the Active VIM that it does not have any configuration. The Active VIM will then upload the network devices configuration to the new VIM, after which the new VIM will reboot again. The new VIM will restart and now will be completely online, but in the Standby state. This entire process will complete in approximately one minute.

After the new VIM is online and operational, it will appear in the VIMNet Explorer as a mismatched device under the Decommissioned list. Furthermore, the placeholder will have an error asserted as shown below:



The following steps must be performed to reconcile the VIMNet configuration with the online network device states:

1. Right click on the VIM placeholder and decommission the VIM that was removed. This will clear the red X and clear the placeholder.
2. Right click on the VIM placeholder and select Reconcile VIM. Then select the VIM from the presented list. The selected VIM will be assigned to the empty placeholder.

7.0 Operational Check

7.1 Scope

The following sections provide some assistance to ensure the interface is working properly.

7.2 Verify Hardware and Software Version Number

You can verify that the Modbus TCP driver has been installed using the DeltaV Diagnostics tool. The Diagnostics tool will show the Hardware Revision No. (HwRev) and the Software Revision No. (SwRev).

To begin the DeltaV Diagnostic tool select Start-> DeltaV-> Operator-> Diagnostics. In the Diagnostics tool, expand the Controller, I/O, and then double click on the Programmable Serial Interface Card that has the driver installed.

The following information will be displayed:

HwRev	Hardware Revision	1.1 or later
SwRev	Software Revision	1.83 or later

Table 13: Verifying Hardware and Software Version Numbers

7.3 Verify Configuring

Verify the port configuration. The serial port must be enabled. You need to make sure that communication settings such as baud rate, parity, and number of data bits match the field device settings.

Verify dataset configuration: The datasets configured must be as shown above.

7.4 Verify I/O Communication with Control Studio

You can create I/O modules in the control studio to verify correct values are read from the VIM. For AI and DI data, the values should be changed in the field devices and verified that the new data are correctly reported in DeltaV. Similarly, verify that the AO and DO data is being written correctly from DeltaV to the field devices.

7.5 Using DeltaV Diagnostics

Verify VIM communication: Select the emulated PSIC in Diagnostics and press the right mouse button. Select Display Real -Time Statistics from the drop down menu. If the PSIC is functioning, then you will see the Valid Responses counter and the Async and/or Sync Transactions counters incrementing. There will not be any errors counting up.

Verify port statistics: Select the Port on the emulated PSIC and press the right mouse button. Then select Display Port Statistics form the drop down menu. Verify that the port communications statistics are being displayed properly and are counting as expected for the protocol’s functionality.

Verify dataset values: Select a dataset and press the right mouse button. Select View Dataset Registers from the Drop down window. Verify that the dataset values are displayed as expected.

7.6 LED Indication

The VIM has six LEDs in front. Top to bottom, these are as follows:

Green	Power
Red	Fault
Green	Active
Green	Standby
Amber	Network Communications
Amber	Railbus Communications

Table 14: LED Indication

The following table describes the VIM operational state as indicated by the LEDs:

LED	State	Description
Green Red Green Green Amber Amber	ON OFF OFF OFF Blink Off	The VIM is Decommissioned. It does not have an assigned IP address. Use the VIMNet Plug and Play Server to configure a VIM placeholder. The VIM placeholder contains the IP address for your network. Then commission the VIM as described in Section 3.
Green Red Green Green Amber Amber	ON OFF ON OFF Blink OFF	The VIM is Commissioned. It is not communicating on the Railbus. Verify that the DeltaV controller has been downloaded. Also, verify that the field device configuration has been uploaded into the VIM from the VIMNet Plug and Play Server.
Green Red Green Green Amber Amber	ON OFF ON OFF ON ON	The VIM is Commissioned, and communicating with the field devices, as well as with the DeltaV controller.
Green Red Green Green Amber Amber	ON OFF ON OFF Blink ON	The VIM is Commissioned, and communicating with the DeltaV controller. The field device communications have errors.
Green Red Green Green Amber Amber	ON OFF ON OFF Blink OFF	The VIM is Commissioned, but communications with the DeltaV controller are not active. The field device communications have errors.
Green Red Green Green Amber Amber	ON ON X X X X	The VIM is in fault. All other LED states (marked with X) are not significant.

Table 15: Simplex VIM LED State Specifications

LED	State	Description
Green Red Green Green Amber Amber	ON OFF OFF OFF Blink Off	The VIM is Decommissioned. It does not have an assigned IP address. Use the VIMNet Plug and Play Server to configure a VIM placeholder. The VIM placeholder contains the IP address for your network. Then commission the VIM as described in Section 3.
Green Red Green Green Amber Amber	ON OFF ON OFF Blink OFF	The VIM is Commissioned. The current redundancy role is ACTIVE. It is not communicating on the Railbus. Verify that the DeltaV controller has been downloaded. Also, verify that the field device configuration has been uploaded into the VIM from the VIMNet Plug and Play Server.
Green Red Green Green Amber Amber	ON OFF OFF ON Blink OFF	The VIM is Commissioned. The current redundancy role is STANDBY. It is not communicating on the Railbus. Verify that the DeltaV controller has been downloaded. Also, verify that the field device configuration has been uploaded into the VIM from the VIMNet Plug and Play Server.
Green Red Green Green Amber Amber	ON OFF ON OFF ON ON	The VIM is Commissioned, and communicating with the field devices, as well as with the DeltaV controller. The current redundancy role is ACTIVE. Note that if the Active LED is OFF and the Standby LED is ON, the current role is Standby.
Green Red Green Green Amber Amber	ON OFF ON OFF Blink ON	The VIM is Commissioned, and communicating with the DeltaV controller. The field device communications have errors. The current redundancy role is ACTIVE. Note that if the Active LED is OFF and the Standby LED is ON, the current role is Standby.
Green Red Green Green Amber Amber	ON OFF ON OFF Blink OFF	The VIM is Commissioned, but communications with the DeltaV controller are not active. The field device communications have errors. The current redundancy role is ACTIVE. Note that if the Active LED is OFF and the Standby LED is ON, the current role is Standby.
Green Red Green Green Amber Amber	ON ON X X X X	The VIM is in fault. All other LED states (marked with X) are not significant.

Table 16: Redundant VIM LED State Specifications

8.0 Technical Support

For technical support or to report a defect, please give MYNAH Technologies a call at (636) 681-1555. If a defect is discovered, please document it in as much detail as possible and then fax your report to us at (636) 681-1660.

You can also send us your questions via e-mail. Our address is:
support@mynah.com