Modbus® Router User Guide

COMTROL®
Network Enabling Devices

Trademark Notices

Comtrol, DeviceMaster, and PortVision are registered trademarks of Comtrol Corporation.

Modbus is a registered trademark of Schneider Electric.

PLC is a registered trademark of Allen-Bradley Company, Inc.

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective owners.

Third Edition, November 7, 2013

Comtrol Corporation makes no representations or warranties with regard to the contents of this document or to the suitability of the Comtrol product for any particular purpose. Specifications subject to change without notice. Some software or features may not be available at the time of publication. Contact your reseller for current product information.

Document Number: 2000547 Rev C

Table of Contents

Section 1. Overview

1.1. Introduction

The Modbus Router application provides enhanced connectivity from a wide variety of Modbus masters to a wide variety of Modbus slaves, advanced Master-to-Master connectivity, and connectivity from private Modbus serial networks to public Modbus networks.

Supported Modbus masters include:

- Modbus/TCP
- Modbus/RTU serial
- Modbus/ASCII serial
- Modbus/ RTU
- Modbus/ASCII over Ethernet TCP/IP

Supported Modbus slaves include:

- Modbus/TCP
- Public and private Modbus/RTU serial
- Public and private Modbus/ASCII serial

Connectivity can be easily achieved between any master(s) and any public slave(s) anywhere on an Ethernet network. Combined with a DeviceMaster UP running the Modbus/TCP application, both serial and Ethernet TCP/IP Raw/ ASCII devices can be accessed anywhere on a network from any Modbus master.

Modbus Router was designed to greatly enhance system maintenance capabilities. Included are comprehensive device and port specific diagnostic web pages that display status, message response timing, time-outs, other error counts, and overall message statistics. A serial log is also included to provide message level diagnosis for serial devices.

1.2. Terms and Definitions

This subsection defines terms and definitions used in this guide.

Term	Definition
Alias Device ID	The device ID that the original received ID is changed to when an Alias Device ID is configured.
Device ID	The Modbus device identification number. Device IDs include: 0 = Broadcast 1-247 = Standard device IDs 248-255 = Reserved device IDs (generally used for vendor specific gateway functions)
Device ID Offset	An offset applied at the slave serial port interface that changes the message device ID range to match the serial device(s) device ID range.

Term	Definition
Ethernet TCP/IP	A form of Ethernet connectivity that provides a level of guaranteed delivery and data verification. This is used for many upper layer protocols such Modbus/TCP and can be also used for transferring Modbus/RTU and Modbus/ASCII messages.
Local Slave Device	A local slave device is one that is connected directly to a serial port on the gateway.
Master (or Client) Mode	The method of operation when a PLC, a gateway, or an application is operating as a *Master* or the message originator.
Modbus/ASCII	An ASCII, or character based, form of Modbus. The base message is the same as Modbus/RTU, but the format is somewhat different.
Modbus/RTU	The standard Modbus messages, in hexadecimal format, that are typically transmitted over serial lines but can also be transmitted over other communication methods such as wireless or Ethernet TCP/IP socket connections. **Note:** *Modbus/RTU over Ethernet TCP/IP is not the same as Modbus/TCP.*
Modbus/TCP	An Ethernet based form of Modbus communication. The base message is the same as Modbus/RTU, but a special Modbus header is included for packet identification and routing purposes.
Polling	The process where a PLC or Application requests data on a continual basis. In this operation the Master sends the request messages while the Slave responds to the messages.
Public Slave Device	A public slave device is one that can communicate to all master(s) on the Modbus network. **Note:** *If a device is not specifically specified as a "public" or "private" slave device, then it can be assumed to be a public device.*
Private Slave Device	A private slave device is one that can only communicate to a master that is connected on the same serial bus, such as RS-485 or RS-422. Private slave device cannot communicate to other masters on the Modbus network.
Received Device ID	The original device ID received in the Modbus message from a Modbus master.
Remote Slave Device	A remote slave device is either a slave Modbus/TCP device or a serial slave device attached to another gateway and accessed as a remote Modbus/TCP device.
Slave (or Server) Mode	The method of operation when a PLC, a gateway, or an application is operating as a *Slave* or the message receiver.
Sockets	The method used to communicate between devices while communicating over Ethernet TCP/IP.

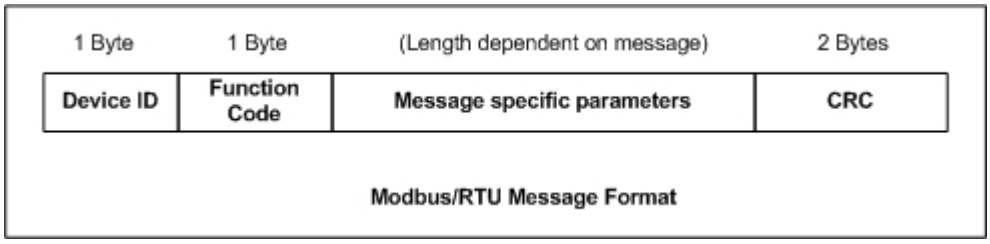
1.3. What is Modbus?

This subsection discusses:

- *Modbus/RTU*
- *Modbus/ASCII* on Page 8
- *Modbus/TCP* on Page 9

1.3.1. Modbus/RTU

Modbus/RTU is native Modbus in hexadecimal format. These are the base Modbus messages that contain simple read and write requests. The format is as follows:



Modbus/RTU Message Format

Where:

- The terms **Master** or **Client** are used to identify the sender of the message.
- The terms **Slave** or **Server** are used to identify the devices responding to the message.

Modbus/RTU is used primarily for:

- Serial port connectivity

  RS-485 is the most common serial mode, but RS-232 and RS-422 are also widely used. Commonly used by both Master and Slave devices.

- Ethernet TCP/IP socket connections

  This is not the same as *Modbus/TCP* on Page 9, but does provide a very simple method of interfacing to remote devices. It is used by many applications and some OPC servers.

  Note: *This communication method typically is not supported by PLCs.*

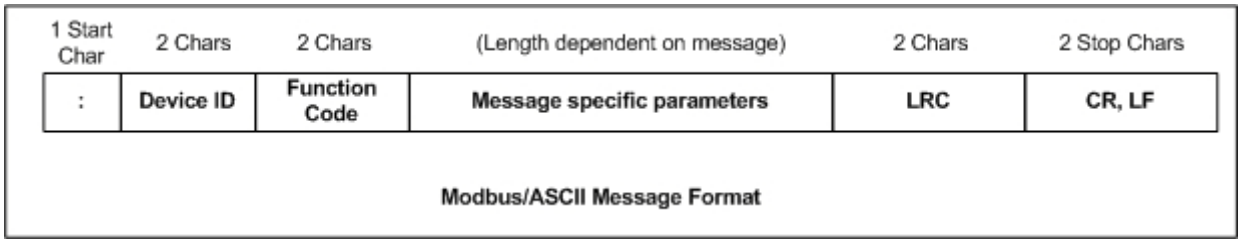
1.3.2. Modbus/ASCII

Modbus/ASCII is native Modbus in ASCII format. This protocol is used primarily by legacy devices and is no longer supported as widely as Modbus/RTU.

Like Modbus/RTU, Modbus/ASCII contains the base Modbus messages that contain simple read and write requests. The differences between Modbus/ASCII and Modbus/RTU are:

1. The message data is sent in ASCII format, so the message length is twice as long. It requires two ASCII characters for each byte of data.
2. An 8-bit LRC is attached to verify the message instead of a 16-bit CRC. The LRC is also transmitted in ASCII format.
3. There are defined starting and ending characters to determine a Modbus/ASCII messages.

The format is as follows:



Modbus/ASCII Message Format

Where:

- The terms **Master** or **Client** are used to identify the sender of the message.
- The terms **Slave** or **Server** are used to identify the devices responding to the message.

Modbus/ASCII is used primarily for:

- Serial port connectivity

  RS-485 is the most common serial mode, but RS-232 and RS-422 are also used. Used primarily by legacy Slave devices.

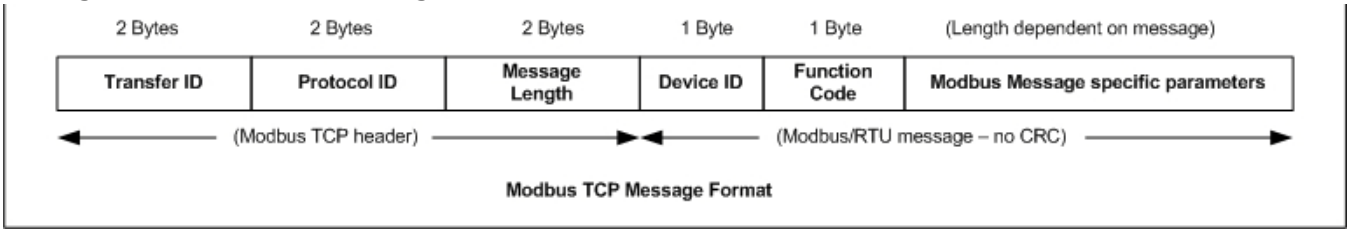
- Ethernet TCP/IP socket connections

  This is not the same as *Modbus/TCP*, but does provide a very simple method of interfacing to remote devices. It is used by some applications and some OPC servers.

  Note: *This communication method typically is not supported by PLCs.*

1.3.3. Modbus/TCP

Modbus/TCP is an Ethernet network based protocol that contains a Modbus/RTU message, with the exception of the 2-byte CRC. The Modbus/TCP message contains a header with information designed to provide message identification and routing information. The format is as follows:



Modbus TCP Message Format

Where:

- The terms **Master** or **Client** are used to identify the sender of the message.
- The terms **Slave** or **Server** are used to identify the devices responding to the message.
- Modbus/TCP messages are typically sent to and received on a defined Ethernet TCP/IP socket of 502.
- Modbus/TCP implementations provide more capability, but also require more processing than simpler Modbus/RTU implementations.

Modbus/TCP is used for connecting advanced Ethernet based devices, such as PLCs, HMIs, SCADA Systems, and most OPC Servers to:

- Other Ethernet devices supporting Modbus/TCP.
- Remote serial Modbus/RTU and/or Modbus/ASCII devices through gateways (such as the DeviceMaster UP running the Modbus Router or Modbus/TCP applications).
- Remote serial or Ethernet TCP/IP ASCII devices (barcode scanners, printers, RFID readers, visions systems, etc) through a gateway (such as the DeviceMaster UP running the Modbus/TCP application).

1.4. Modbus Router Functionality

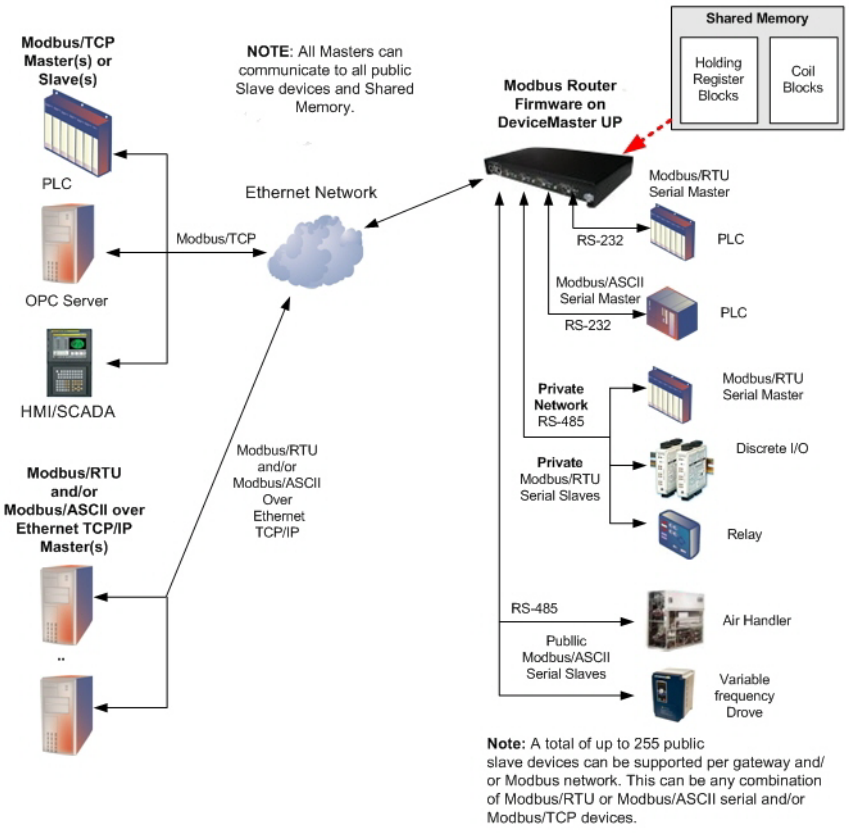
The Modbus Router application provides the following functionality:

- Provides robust connectivity from all supported master devices to all supported slave devices:
  - Supported Masters:
    - Modbus/TCP Master
    - Modbus/RTU Serial Master
    - Modbus/ASCII Serial Master
    - Modbus/RTU over Ethernet TCP/IP Master
    - Modbus/ASCII over Ethernet TCP/IP Master
  - Supported Slaves:
    - Modbus/RTU Serial Slaves
    - Modbus/RTU ASCII Slaves
    - Remote Modbus/TCP Slaves
    - Remote Modbus/RTU Serial Slaves (via Modbus/TCP using another gateway)
    - Remote Modbus/ASCII Serial Slaves (via Modbus/TCP using another gateway)
- Multiple DeviceMaster UP chassis can be used together to form a Modbus network.
- Supports up to 255 public Modbus devices per gateway and/or Modbus network. Both standard, (1-247), and reserved, (248-255), device IDs are supported.
- All Modbus devices not configured remotely are assumed to be locally connected to the gateway.
- The locations of all local Modbus devices are determined automatically.
- Modbus Device ID Aliasing. Modbus message device ID can be *aliased* when messages are received from a Modbus master.
- Device ID Offset. At the serial port interface, device IDs can have a positive or negative offset applied to change the device ID range.
- Supports up to 96 Modbus/TCP connections. This can include any combination of slave and master connections.
- Modbus/RTU and Modbus/ASCII over Ethernet TCP/IP Master specific:
  - Can support up to six TCP/IP connections per serial port configuration.
  - All messages received from any Ethernet TCP/IP Master connection enter the routing functionality and can be routed to any local or remote device.
  - Combined with a serial port redirector, such as the Comtrol Secure Port Redirector, can provide COM port functionality from a computer to the Modbus network.
- Supports connectivity to private Modbus serial busses, such as a serial master and slave(s) on a RS-485 loop:
  - The Modbus master is provided connectivity to the public Modbus network.
  - Only the master has direct access to the serial devices on the private serial network.
  - Private slaves are protected from intrusion by other master(s).
  - Both slave device specific and port level diagnostics are provided for private network communication.

- Advanced Master-to-Master connectivity is provided via the Shared Memory sub-system. Features are:
  - Eight Holding Register blocks containing 200 registers each.
  - Eight Coil blocks containing 160 coils each.
  - Read access to all master(s) on the Modbus network.
  - Block-specific configurable write-access control. For each block, writes can be enabled for all master(s) or restricted to a specific serial, Modbus/TCP, or Ethernet TCP/IP master.
  - Web pages provide configuration, diagnostics, display of block contents and shared memory clearing capabilities.
- Modbus specific message handling:
  - CRC verification of all messages received on the TCP/IP and serial Modbus/RTU interfaces.
  - LRC verification of all messages received on the TCP/IP and serial Modbus/ASCII messages.
  - Timing out of responses from slave Modbus devices.
  - Transfer ID verification of all remote Modbus/TCP messages.
  - Parameter checking of all slave responses.
  - Broadcast message handling.
- System monitoring to ensure gateway operation:
  - Gateway busy.
  - Application message time-outs
  - Message validity checking.
- Advanced diagnostics web pages:
  - Modbus device specific statistics, response timing, and status. Up to 255 Modbus devices, both attached and remote, can be monitored simultaneously.
  - Serial port specific statistics and status.
  - Serial port message logging.

Supported Connectivity includes:

Modbus Router Connectivity Grid		Masters				
		Modbus/ TCP Master	**Modbus/ RTU Serial Master**	**Modbus/ ASCII Serial Master**	**Modbus/RTU Master over Ethernet TCP/IP**	**Modbus/ ASCII Master over Ethernet TCP/IP**
Slave Devices	Modbus/TCP Slave (remote gateway or slave Modbus/TCP device)	X	X	X	X	X
	Local Public Modbus/RTU Serial Slave	X	X	X	X	X
	Local Public Modbus/ASCII Serial Slave	X	X	X	X	X
	Remote Modbus/ RTU Serial Slave (via remote gateway)	X	X	X	X	X
	Remote Modbus/ ASCII Serial Slave (via remote gateway)	X	X	X	X	X
	Local Private Modbus/RTU Serial Slave		X			
	Local Private Modbus/ASCII Serial Slave			X		


Modbus/TCP Master(s) or Slave(s)
PLC
OPC Server
HMI/SCADA
Modbus/RTU and/or Modbus/ASCII over Ethernet TCP/IP Master(s)
NOTE: All Masters can communicate to all public Slave devices and Shared Memory.
Ethernet Network
Modbus/TCP
Modbus/RTU and/or Modbus/ASCII Over Ethernet TCP/IP
Modbus Router Firmware on DeviceMaster UP
Shared Memory
Holding Register Blocks
Coil Blocks
Modbus/RTU Serial Master
RS-232
PLC
Modbus/ASCII Serial Master
RS-232
PLC
Private Network RS-485
Modbus/RTU Serial Master
Private Modbus/RTU Serial Slaves
Discrete I/O
Relay
RS-485
Air Handler
Publlic Modbus/ASCII Serial Slaves
Variable frequency Drove
Note: A total of up to 255 public slave devices can be supported per gateway and/ or Modbus network. This can be any combination of Modbus/RTU or Modbus/ASCII serial and/or Modbus/TCP devices.

1.5. Multiple Gateway Modbus Networks

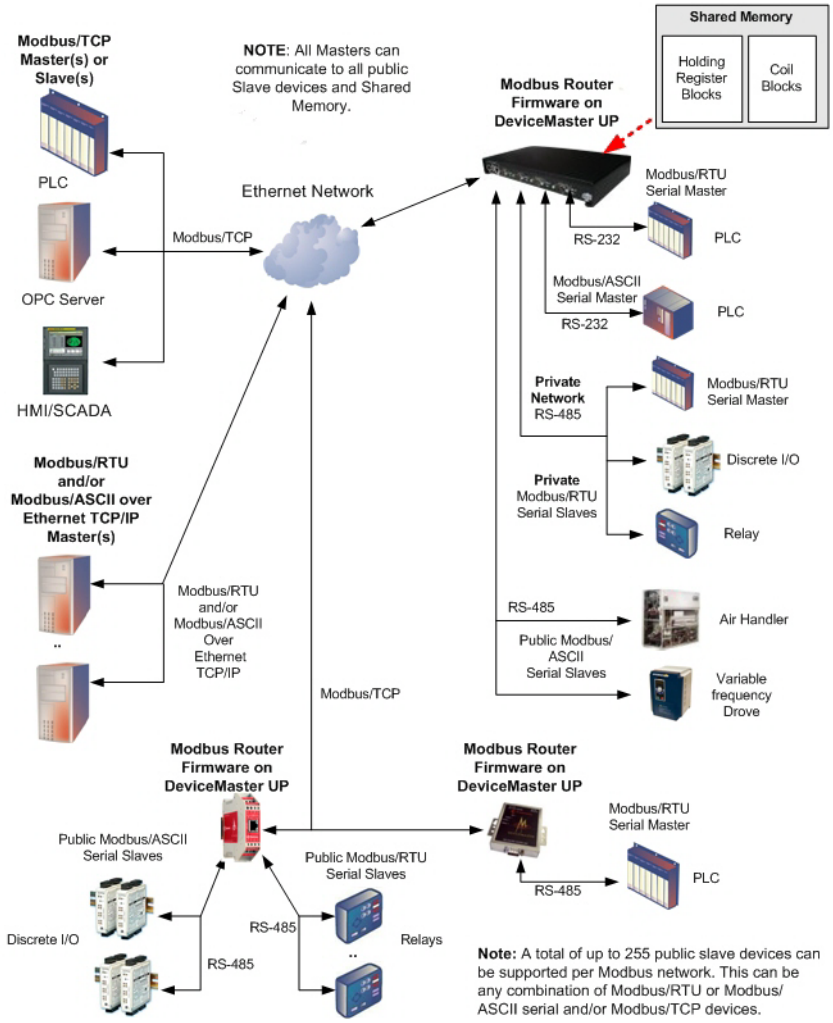
A multiple gateway Modbus Network is created by combining two or more Modbus slave(s) and master(s) with two or more DeviceMaster UPs running the Modbus Router and/or Modbus/TCP firmware applications.

- Typically, at least one chassis will be running the Modbus Router firmware application.
- The Modbus/TCP application can provide connectivity to local or remote serial and Ethernet TCP/IP Raw/ASCII devices such as barcode scanners, RFID readers, printers, vision systems, and weigh scales.
- All Modbus masters connected to a DeviceMaster UP running Modbus Router can communicate to all public slaves.

The following connectivity can be provided in a Modbus network when using multiple DeviceMaster UP chassis with Modbus Router and Modbus/TCP firmware applications:

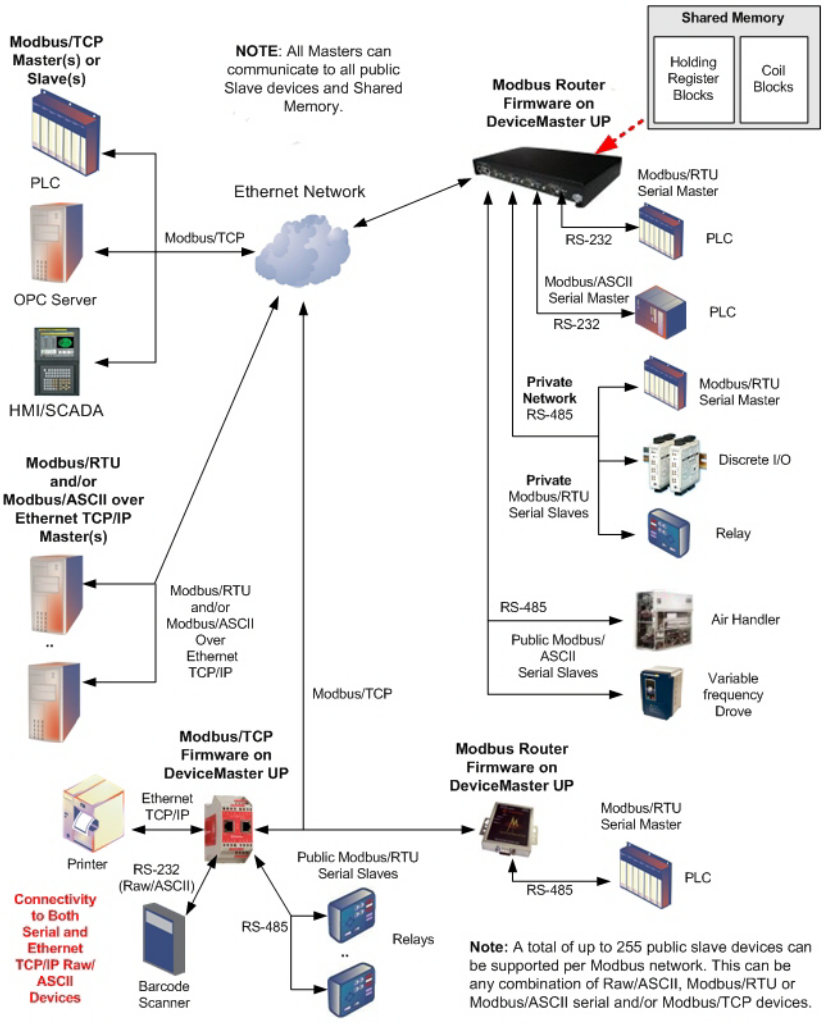
Modbus Network Connectivity Grid		**Masters**				
		Modbus/ TCP Master	**Modbus/ RTU Serial Master**	**Modbus/ ASCII Serial Master**	**Modbus/RTU Master over Ethernet TCP/IP**	**Modbus/ ASCII Master over Ethernet TCP/IP**
Slave Devices	Modbus/TCP Slave (remote gateway or slave Modbus/ TCP device)	X	X	X	X	X
	Local Public Modbus/RTU Serial Slave	X	X	X	X	X
	Local Public Modbus/ASCII Serial Slave	X	X	X	X	X
	Remote Modbus/RTU Serial Slave (via remote gateway)	X	X	X	X	X
	Remote Modbus/ASCII Serial Slave (via remote gateway)	X	X	X	X	X
	Serial Raw/ ASCII Devices (remote or local)	X	X	X	X	X
	Ethernet TCP/ IP Raw/ASCII Devices (remote or local)	X	X	X	X	X
	Local Private Modbus/RTU Serial Slave		X			
	Local Private Modbus/ASCII Serial Slave			X		

The following diagram demonstrates a multiple gateway Modbus network utilizing Modbus Router firmware.



Multiple DeviceMaster UPs Running Modbus Router firmware in a Modbus Network

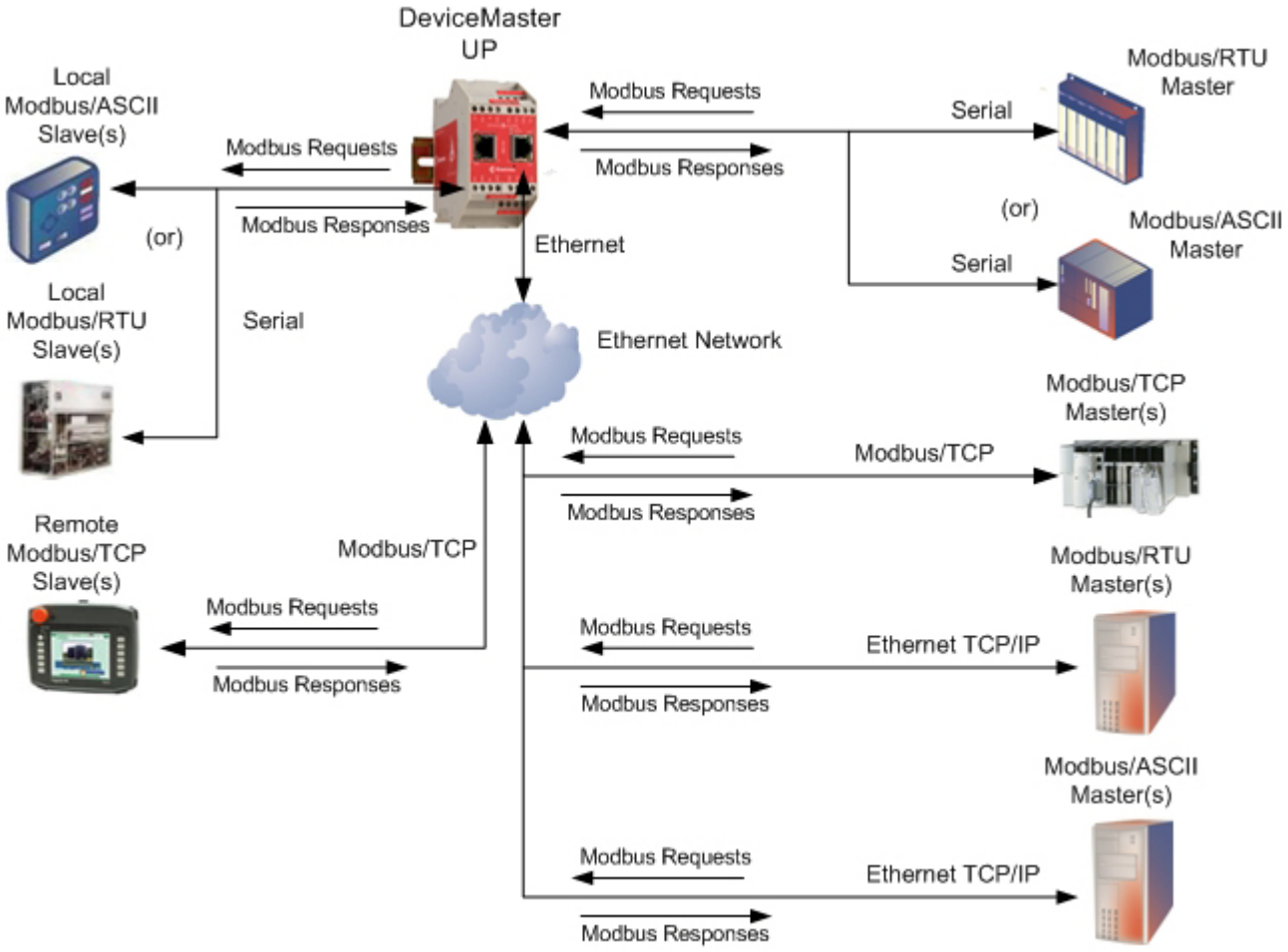
The following diagram demonstrates a multiple gateway Modbus network utilizing both Modbus Router and Modbus/TCP firmware.



DeviceMaster UPs Running Modbus Router and Modbus/TCP Firmware in a Modbus Network

1.6. Modbus/RTU and Modbus/ASCII To-Slaves Interface

The DeviceMaster UP provides access to serial Modbus/RTU and Modbus/ASCII slave devices. Modbus master messages are translated to appropriate Modbus slave messages, public slave devices are automatically located, and appropriate responses are returned to the Modbus master.



Modbus Master(s) to Public Modbus Slave(s)

1.6.1. Communication Methodology

The DeviceMaster UP translates various Modbus formats and forwards them to public slave devices attached to the Modbus To-Slaves configured serial ports. Each Modbus message is transmitted and a response is expected. The DeviceMaster UP times out the Modbus messages if there is no response returned within the configured timeout period.

The following diagram displays the Modbus message transfer. The following apply to Modbus To-slaves serial ports.

- All valid Modbus messages are translated to the appropriate format for serial port or Modbus/TCP transmission.
- Local public Modbus slave devices are automatically located on a DeviceMaster UP 2-port or 4- port.
- Local and remote public Modbus slave devices can be accessed from a Modbus master as if they were of that master's protocol type.
- Messages are timed out if no response is returned within the configured timeout period.
- Appropriate Modbus responses are returned to the Modbus master.
- Broadcast Modbus messages, those with a unit identifier of zero, are transmitted out all Modbus To-slaves serial ports on the DeviceMaster UP. Depending on the remote device configuration(s), remote slave devices may or may not receive broadcast messages.

From a message routing standpoint, all local and remote public Modbus slave devices attached to a DeviceMaster UP gateway (1, 2, or 4-port) must be addressed with unique Unit Identifiers. Valid Unit Identifiers are 1 to 255 and the Broadcast Identifier is zero. However, the Device ID Offset functionality can be configured to change the addressing of serial connected slave devices to allow multiple slave devices with the same unit ID to be connected to the same gateway, but addressed differently. For more information, see section on Alias and Device ID Offset functionality.

To communicate to local Modbus slave device(s) through a DeviceMaster UP, perform the following steps.

1. Using the *Serial Interface Configuration* page, select the appropriate Port.
2. Under *Serial Configuration*, configure the serial port parameters such as the Mode, Baud rate, Data Bits, and so forth.
3. Under *General Protocol Setting*s, set the *Select Serial Port Protocol* to **Modbus/RTU-To-Slaves** or **Modbus/ASCII-To-Slaves**.
4. Under *Modbus Slaves Protocol Settings*, set the **Device Response Timeout** to the desired value.

   Note: *2- and 4-Port only: Set the **Lost Device Search Enable** setting. See Local Public Modbus Slave Device Search Methodology.(below) for more information.*
5. If desired, configure the Device ID Offset Mode and Device ID Offset.

To communicate to remote Modbus slave device(s) through a DeviceMaster UP, perform the following steps.

1. Access the *Remote Modbus/TCP Device Configuration* page.
2. Click **Add/Modify Remote Modbus/TCP List**.
3. Configure each remote device as needed and click **Submit**.

To enable the Alias Device ID capability for one or more device Ids, perform the following steps.

1. Access the *Alias Device ID Configuration/Status* page.
2. Click **Add/Modify Alias Modbus Device ID List**.
3. Configure each alias device ID as needed and click **Submit**.

1.6.2. Local Public Modbus Slave Device Search Methodology

Locating a local Modbus slave device on a DeviceMaster UP 1-port is relatively simple. Either the Modbus slave device is connected to the port or it is not. However, if more than one port is configured for Modbus To-Slaves on a DeviceMaster UP 2- or 4-port, the device must be found. The following is an explanation of how the search algorithm works on a DeviceMaster UP 2- or 4-port.

Locating a Local Modbus slave device after a reboot or port reset: When the DeviceMaster UP receives a message for a public Modbus slave device for the first time since reboot or port initialization, it will transmit the Modbus message out all Modbus To-Slaves serial ports and wait for a response to be returned. Once the response is returned, the device port is known and all messages sent to the device will be routed through the serial port.

Lost Devices: Lost devices, or devices that time out, are a special case. The DeviceMaster UP provides two methods for handling lost devices via the **Lost Device Search Enable** option on the web page.

- Disabling this option on a Modbus To-Slaves port:
  - Prevents the DeviceMaster UP from searching for a lost device on other Modbus To-Slaves ports.
  - Prevents lost devices known to have been on other ports from being searched for on this port.

  Note: *This is the recommended setting whenever it is desired to prevent timeout delays on other Modbus To-Slaves ports in the event that a device times out.*
- Enabling this option on a Modbus To-Slaves port:
  - Allows the DeviceMaster UP to search for lost devices on all Modbus To-Slaves ports with the **Lost Device Search Enable** option turned on.
  - This will cause timeout delays on all Modbus To-Slaves ports with the **Device Search Enable** option turned on until the device is found.

  Note: *This can be useful for locating devices if a device has been moved onto another port by moving the serial cable or, perhaps, by moving the device onto a different Modbus To-Slaves serial bus.*

1.7. Alias Device ID and Device ID Offset Functionality

One of the most common challenges people face when setting up Modbus systems are the problems caused by the limited device ID range. The Alias Device ID and Device ID Offset functionality have been developed to help solve those problems.

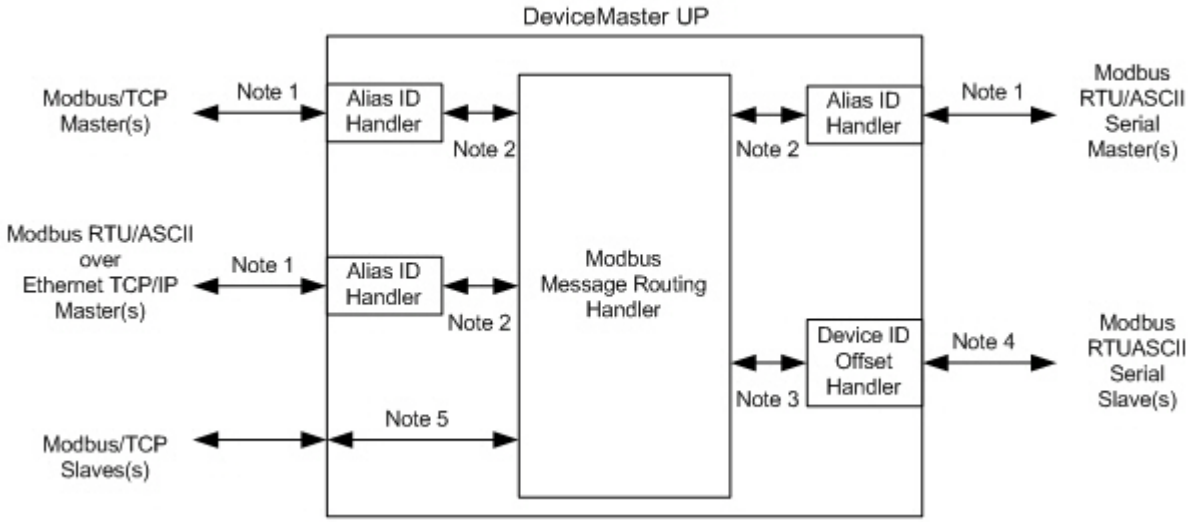
The Modbus specification has the following limitations:

- Requires all public devices attached to gateway to be addressed by a device ID.
- Allows only 256 device IDs with a range of 0 to 255.
- Not all device IDs can be used for addressing devices.
  - Device ID 0 is reserved for broadcast messages
  - 1-247 are for device addressing
  - 248 to 255 are reserved for such things as gateway functions. Depending on your environment, these device IDs may or may not be available for assignment to devices.

The following are common problems that can occur as a result of the device ID limitations:

- A gateway must route Modbus messages based on the device ID. Therefore, it cannot route to multiple Modbus devices with the same device ID.
- It is not always possible or practical to change the device ID of serial Modbus slave devices.
- Serial and Ethernet TCP/IP Modbus RTU/ASCII masters with one connection may need to access multiple devices with the same device ID. Furthermore, these devices may be located locally or remotely.
- It is not always possible or practical to modify the device IDs on existing Modbus master programs. This is often true when adding a SCADA system to an existing PLC controlled system.

The Alias Modbus Device ID and Device ID Offset functionality has been developed to solve these problems. This functionality is described in the following diagram:



Alias Device ID and Device ID Functionality

Note 1:Originally received Modbus messages. All responses will be returned with the original device ID.

Note 2:Modbus messages sent to and responses received from Modbus Message Routing Handler. Depending on the Alias ID configuration, these messages may contain the originally received device ID or the alias device ID.

Note 3:Modbus messages received from the Modbus Message Routing Handler. Depending on the Alias ID configuration, these messages may contain the originally received device ID from the Modbus master or the alias device ID. All responses contain the device ID as received from the Modbus Message Routing Handler.

Note 4:Modbus messages sent to Modbus serial slaves. Depending on the Device ID Offset configuration for the serial port, these messages may be the same as those received from the Modbus Message Routing Handler or have a device ID that has been either incremented or decremented to match the serial device ID range.

Note 5:Modbus messages received from the Modbus Message Routing Handler. Depending on the Alias ID configuration, these messages may contain the originally received device ID from the Modbus master or the alias device ID. Device ID Offset functionality does not apply to Modbus/TCP slaves.

1.7.1. Alias Modbus Device ID Functionality

The Alias Modbus Device ID functionality allows modification of device IDs only when messages are received from Modbus masters. When configured, a Modbus message from a master with the specified device ID is converted to the alias device ID, the message is then routed internally using the alias device ID. All responses are returned to the master with the original received message device ID.

The following table demonstrates several device ID aliasing examples:

Received Device ID	Alias Device ID	Routed Message Device ID	Description
1	10	10	Convert messages with received device ID 1 to 10. Route message with device ID 10.
50	5	5	Convert messages with received device ID 50 to 5. Route message with device ID 5.
100	254	254	Convert messages with received device ID 100 to 254. Route message with device ID 254.
10	10	10	Invalid configuration attempt. No change to device ID is performed.

1.7.2. Device ID Offset Functionality

The Device ID Offset functionality allows modification of device IDs when messages are transmitted to serial Modbus slave devices. When configured, the Device ID Offset functionality will modify the device ID received in the message to match the actual device ID range of the serial device(s). The device ID range is effectively either increased or decreased depending on the serial port Device ID Offset configuration.

The following table demonstrates several Device ID Offset examples:

Device ID Offset Mode	Device ID Offset	Valid Message Device ID Range	Valid Serial Device ID Range	Description
Off	0	1-255	1-255	Default mode. Device IDs are unchanged.
Add-to-Msg-ID	50	1-205	51-255	Increase device ID range by 50. *Examples*: Device ID 1 is converted to 51 Device ID 10 is converted to 60 Device ID 120 is converted to 170
Subtract-from-Msg-ID	100	101-255	1-155	Decrease device ID range by 100. *Examples*: Device ID 101 is converted to 1 Device ID 150 is converted to 50 Device ID 225 is converted to 125

It is highly recommended to take great care when configuring the Device ID Offset functionality. Verify the following when configuring the Device ID Offset:

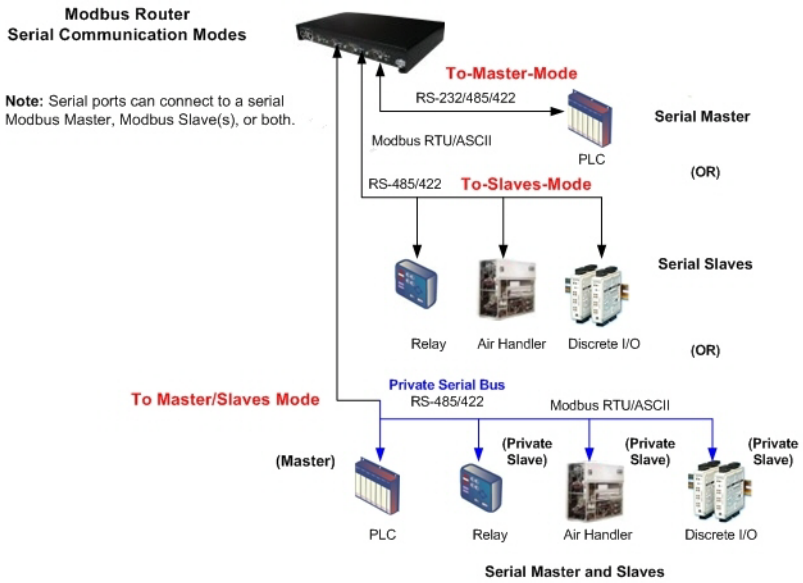
- Check for Device ID overlaps. Be certain that no two devices can have same device ID as recognized by the internal Modbus Message Routing Handler.
- Check for conflicts with the Alias device ID configuration. The Device ID Offset configuration must coincide with any Alias device ID configurations.
- Verify the valid device ID ranges are sufficient to address all serial devices.

1.8. Modus Master/Slaves Serial Port Mode (Private Serial Bus)

As of Modbus Router v5.10, Modbus masters and slave(s) can be connected together on the same serial port. This provides the following benefits:

1. A serial Modbus master can communicate to slaves on its' own private serial bus as well as public slaves on a Modbus network. In this configuration, a serial master can communicate to:
   a. Modbus RTU/ASCII slave(s) on its own serial bus.
   b. Public Modbus RTU/ASCII serial slave(s) connected to the same DeviceMaster UP.
   c. Modbus/TCP slaves.
   d. Remote public Modbus RTU/ASCII serial slave(s) via an Ethernet attached Modbus gateway.
   e. All other Modbus master(s) on the Modbus network via the Shared Memory functionality.
2. The Modbus slaves on the serial bus are *private* to the master on that serial bus.
   a. The slave device(s) are affectively protected from all other Modbus masters on the Modbus network.
   b. The master has total control of communication to the slaves on its own serial bus.
   c. The master can provide data to/from the slave(s) to the Modbus network, and other Modbus masters, via the Shared Memory functionality.
3. Deployment can be greatly simplified.
   a. An existing serial bus can be left intact, thusly reducing the rewiring effort.
   b. The only wiring change is to attach the DeviceMaster UP to the bus anywhere there is access.
4. The system can be more fault-tolerant.
   a. In the event the DeviceMaster UP is powered off, the master and slaves on the serial bus can still communicate.
   b. By preventing other masters from communicating to the slave devices on the serial bus:
      - Other masters cannot cause disruptions in communications between the master and slaves on the serial bus by overloading the gateway.
      - Message latency time between the master and slaves is minimized.
5. Maintenance and downtime costs can be minimized with detailed diagnostics web pages provided by the DeviceMaster UP.

The following diagram demonstrates the To-Master/Slaves mode compared to the To-Master and To-Slaves modes.



1.8.1. Master/Slaves Message Routing

On a serial port configured to Master/Slaves, only the master on the private serial bus has access to the serial slaves on that serial bus. However, the master can also communicate to public devices and shared memory anywhere on the Modbus network.

The *Serial Interface Configuration* web page contains two configuration settings that determine how the serial bus routing is controlled.

1.8.2.1. Forward Broadcasts From Master Option

If this option is selected, all broadcast messages received on this port will be forwarded to the Modbus network. If this option is not selected, all broadcast messages received on this serial port will be dropped and only the private slave devices will receive them.

Note: *Make sure that broadcasts should be forwarded before selecting this option. If this option is selected, all public slave devices on the local Modbus network will receive the broadcast messages and that may cause unpredictable results.*

1.8.3.2. Private Device ID Range Setting

This range defines the expected private slave device ID range on the serial bus.

- Modbus request messages received on this port that are within this device ID range will not be forwarded to the Modbus network.
- All communication to device(s) in that range must occur between the Modbus master and slave(s) on that serial bus.
- The private device ID range must not include public device(s) addressed by the serial master. Loss of communication errors will result if this occurs. It is recommended to use either the Alias and/or the Device ID Offset functionality to address those cases where a public device ID falls into the private device ID range.
- All received Modbus request messages that are not within the private device ID range or have been previously detected on the serial bus, will be forwarded to the Modbus network via the DeviceMaster UP.

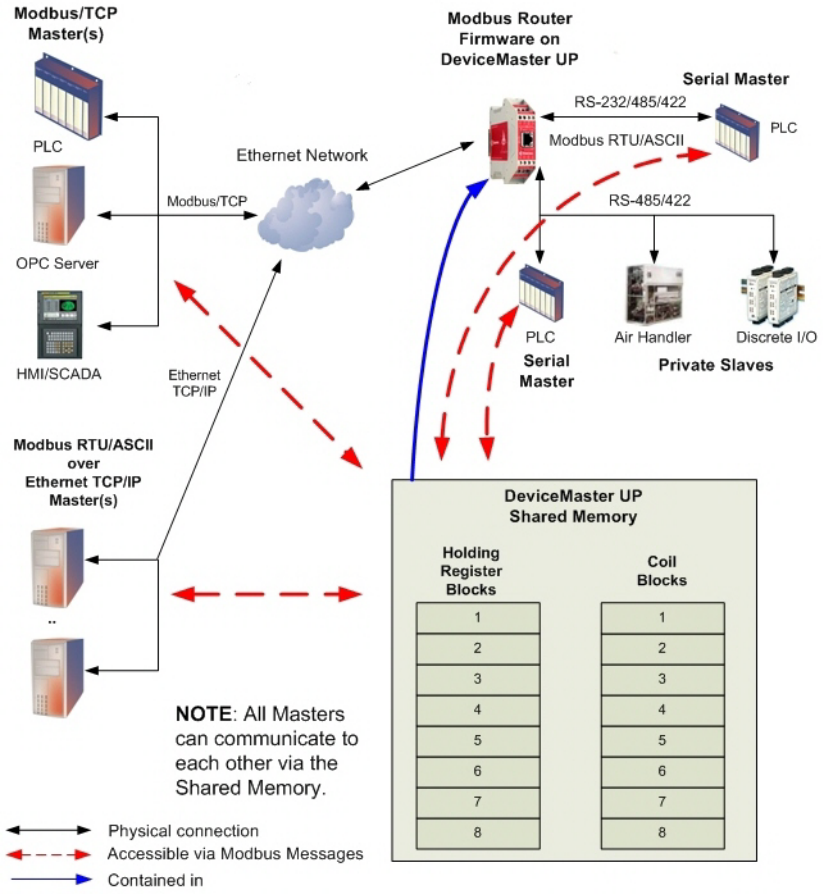
- Responses, including error messages such as timeouts, received from the Modbus network will be forwarded to the serial master.
- The DeviceMaster UP has a built-in auto-detect algorithm for detecting private slave device(s) with ID(s) not defined within the private device ID range. If a response from such a device is received on the serial bus, the DeviceMaster UP will add that device ID to the private device list and will no longer route those messages to the Modbus network.

Note: *There is a potential race condition if a slave device ID is not within the private device ID range and exists both as a public device on the Modbus network and as a private device on the serial bus. If this situation occurs and the public device responds before the private serial device, the master would receive two responses for that request, with the first response coming from the public device. If the public and private device responded at the same time, the Modbus master could detect a corrupted response. However, once the private slave device responds with a valid response, the auto-detect algorithm will prevent forwarding of additional requests to the Modbus network.*

1.9. Shared Memory Functionality

The Shared Memory functionality has been added to provide a simple and robust method for master-to-master communication.

- The Shared Memory interface contains eight 200 Holding Register blocks and eight 160 Coil blocks.
- All Modbus masters, (Modbus/TCP, serial Modbus RTU/ASCII, and Modbus RTU/ASCII over Ethernet TCP/IP), can read the contents of the Shared Memory blocks.
- Write access can be controlled to each Holding Register and Coil block. Each block can be configured to provide all masters write access or be restricted to a port-specific serial master, a Modbus/TCP master or an Ethernet TCP/IP master.
- The Shared Memory contents can be displayed and cleared via the embedded web pages.
- Diagnostics for each block include read, write and blocked write message counts.
- Blocked write messages are recorded in the *Write Violation Log*.



Modbus Router Shared Memory Functionality

This table displays the supported Holding Register Block Function Codes.

Function Code	Description
3	Read Holding Registers
6	Write Single Register
16	Write Multiple Registers
22	Write Mask Register
23	Read Write Registers

This table shows the supported Coil Block Function Codes.

Function Code	Description
1	Read Coils
5	Write Single Coil
15	Write Multiple Coils

Section 2. Embedded Configuration and Diagnostic Pages

This section discusses the embedded Modbus Router pages, which are used to configure the DeviceMaster UP and provide status information.

2.1. Prerequisites

Before you can configure Modbus Router on the DeviceMaster UP, you must have previously performed the following steps:

- Install the hardware
- Install *PortVision DX*
- If necessary, upload the *Modbus Router firmware* using PortVision DX

  Note: *Models that have Modbus Router loaded on the DeviceMaster UP are identified in PortVision DX and the DeviceMaster UP is labeled accordingly.*
- Configure the DeviceMaster UP IP address using PortVision DX

Note: *If necessary, refer to the DeviceMaster UP Hardware Installation and Configuration Guide for the above procedures.*

2.2. Configuration Overview

The following overview shows how to access the DeviceMaster UP Router Configuration embedded pages.

If you have not configured the network information into the DeviceMaster UP during initial setup, you must configure the network information before configuring serial/socket port characteristics. See the *DeviceMaster UP Hardware Installation and Configuration Guide* or the PortVision DX help system for help configuring the network settings.

1. From PortVision DX, highlight the DeviceMaster UP that you want to configure and select **Webpage**.

   Note: *Optionally, enter the IP address of the device in the Address box of your web browser.*
2. Select the appropriate procedure for your environment.

   Serial Modbus Master, Slave, or Master/Slaves

   a. Click **Serial Interface Configuration**.

   b. Click the port that you want to configure.

   c. Change the serial port configuration properties as required for your installation.

   Ethernet TCP/IP Modbus Master

   a. Click **Ethernet TCP/IP Interface Configuration**.

   b. Click the interface that you want to configure.

   c. Change the Ethernet TCP/IP configuration properties as required for the interface.

   Remote Modbus/TCP Device

   a. Click **Remote Modbus/TCP Device Configuration**.

   b. Update the remote Modbus/TCP configuration properties as required for the device.

Alias Modbus Device ID

a. Click **Alias Modbus Device ID Configuration/Status**.
b. Click **Add/Modify Alias Modbus Device ID List**.
c. Configure Alias Modbus Device IDs as required.

Shared Memory

a. Click **Shared Memory Config/Status**.
b. Click **Edit Shared Memory Configuration**.
c. Configure Shared Memory blocks as required.

3. Click **Submit** to commit the changes and repeat for each interface that requires configuration.

2.3. Modbus Router Home Page

Access the main DeviceMaster UP Modbus Router page (*Router Configuration*) from PortVision DX or enter the IP address of the DeviceMaster UP in the **Address** box of your web browser.

The *Router Configuration* page displays the software version and current network configuration for the DeviceMaster UP. In addition, the *Router Configuration* page links to the configuration, statistics, and diagnostics pages. These pages are available from each page for your convenience.

Router Configuration Page	
Software	Modbus Router firmware version currently running on the DeviceMaster UP.
Serial Number	DeviceMaster UP serial number.
IP Config	Type of IP configuration currently in use (static or DHCP).
IP Address, IP Netmask, IP Gateway	IP address, netmask, and gateway configured in the DeviceMaster UP.

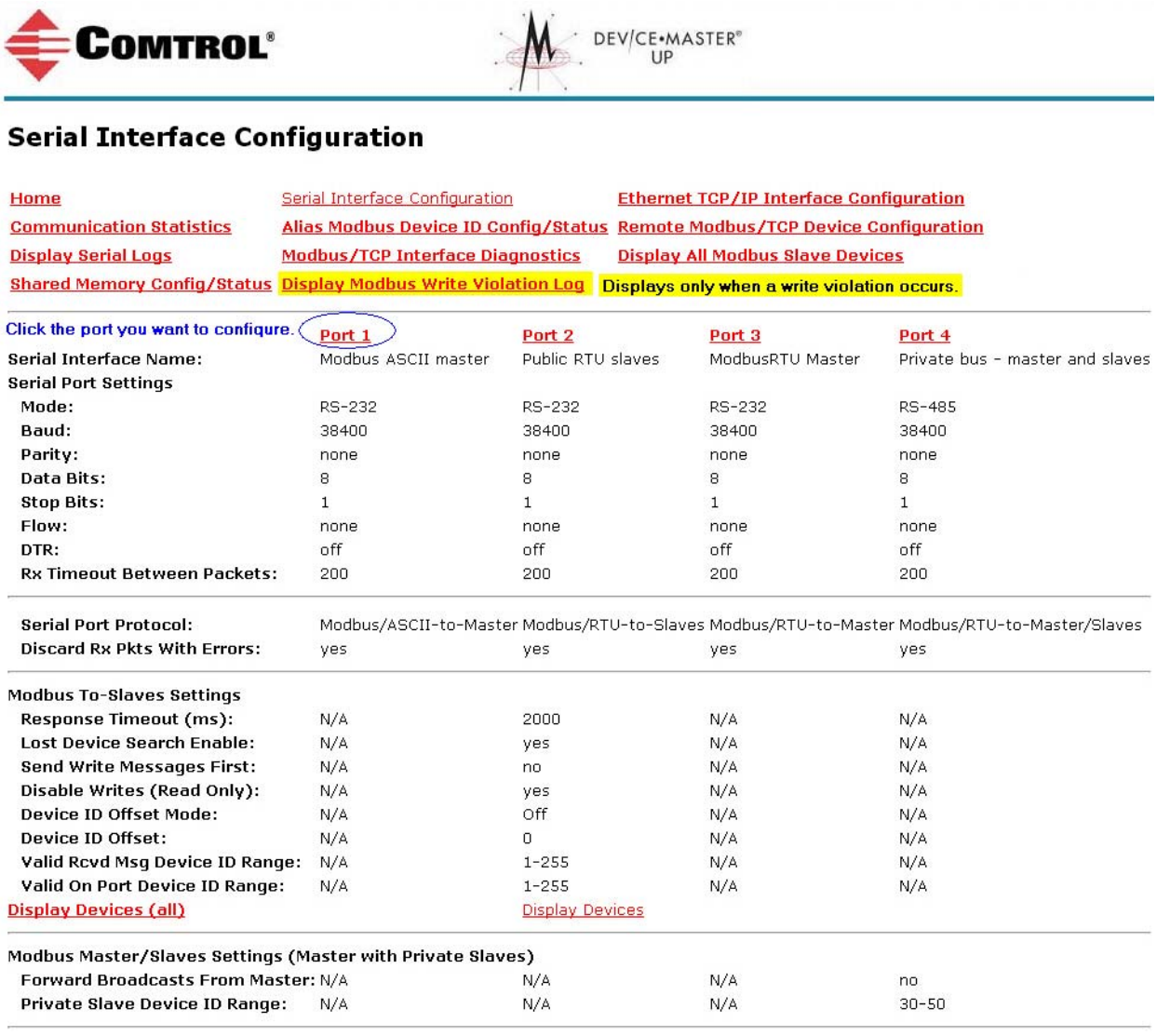
Router Configuration Page (Continued)	
Serial Interface Configuration	Opens the *Serial Interface Configuration* page (Page 30), which provides an overview of the serial interface settings and access to the *Edit Port Configuration* page (Page 31) for serial port configuration on the selected port.
Ethernet TCP/IP Interface Configuration	Opens the *Ethernet TCP/IP Interface Configuration* page (Page 35), which provides an overview of the Ethernet TCP/IP interface settings and access to the *Edit Socket Configuration* page (Page 36) for socket port configuration.
Remote Modbus/TCP Device Configuration	Opens the *Remote Modbus/TCP Device Configuration* page (Page 38), which provides an overview of the Remote Modbus/TCP configuration and access to the Add/Edit/Modify/Delete Remote Modbus/TCP configuration pages.
Alias Modbus Device ID Configuration/Status	Opens the *Alias Modbus Device ID Configuration/Status* page (Page 43), which provides a method to create alias Modbus Device IDs and access to the *Add/Edit/Modify/Delete Alias Modbus Device ID List* and statistics.
Shared Memory Config/ Status	Opens the *Shared Memory Configuration/Status* page (Page 47), which provides shared memory configuration and statistics.
Communication Statistics	Opens the *Serial Port and Ethernet TCP/IP Communication Statistics* page (Page 54), which contains the serial and Ethernet interface statistics.
Modbus/TCP Interface Diagnostics	Opens the *Modbus/TCP Interface Diagnostics* page (Page 57), which contains Modbus/TCP interface statistics.
Display All Modbus Slave Devices	Opens the *Known Modbus Slave Device List* page (Page 62), which contains statistics for the automatically located serial Modbus devices and configured remote Modbus devices.
Display Modbus Write Violation Log	Opens the *Modbus Write Violoation Log* page (Page 60), which contains information for any write attempts to either read-only serial ports or shared memory blocks with writes restricted to other master(s).
Display Serial Logs	Opens the *Serial Interface Logs* page (Page 65), which provides access to the receive and transmit serial logs.
Configure Network	Opens the *Configure Network* page (Page 66), which can be used to modify DeviceMaster UP network configuration after initial configuration using PortVision DX.
Configure Security	Opens the E*dit Security Configuration* and *Key and Certificate Management* page (Page 67).
Reboot	Reboots the DeviceMaster UP.

2.4. Serial Interface Configuration

This subsection discusses the following serial interface configuration pages:

2.4.1. Serial Interface Configuration Page

The *Serial Interface Configuration* page provides access to the *Edit Serial Port Configuration* page when you click the port number. This page also displays the configured serial settings for each port.



Serial Interface Configuration

Home | Serial Interface Configuration | Ethernet TCP/IP Interface Configuration
Communication Statistics | Alias Modbus Device ID Config/Status | Remote Modbus/TCP Device Configuration
Display Serial Logs | Modbus/TCP Interface Diagnostics | Display All Modbus Slave Devices
Shared Memory Config/Status | Display Modbus Write Violation Log | Displays only when a write violation occurs.

Click the port you want to configure.

	Port 1	Port 2	Port 3	Port 4
Serial Interface Name:	Modbus ASCII master	Public RTU slaves	ModbusRTU Master	Private bus - master and slaves
Serial Port Settings				
Mode:	RS-232	RS-232	RS-232	RS-485
Baud:	38400	38400	38400	38400
Parity:	none	none	none	none
Data Bits:	8	8	8	8
Stop Bits:	1	1	1	1
Flow:	none	none	none	none
DTR:	off	off	off	off
Rx Timeout Between Packets:	200	200	200	200
Serial Port Protocol:	Modbus/ASCII-to-Master	Modbus/RTU-to-Slaves	Modbus/RTU-to-Master	Modbus/RTU-to-Master/Slaves
Discard Rx Pkts With Errors:	yes	yes	yes	yes
Modbus To-Slaves Settings				
Response Timeout (ms):	N/A	2000	N/A	N/A
Lost Device Search Enable:	N/A	yes	N/A	N/A
Send Write Messages First:	N/A	no	N/A	N/A
Disable Writes (Read Only):	N/A	yes	N/A	N/A
Device ID Offset Mode:	N/A	Off	N/A	N/A
Device ID Offset:	N/A	0	N/A	N/A
Valid Rcvd Msg Device ID Range:	N/A	1-255	N/A	N/A
Valid On Port Device ID Range:	N/A	1-255	N/A	N/A
Display Devices (all)		Display Devices		
Modbus Master/Slaves Settings (Master with Private Slaves)				
Forward Broadcasts From Master:	N/A	N/A	N/A	no
Private Slave Device ID Range:	N/A	N/A	N/A	30-50

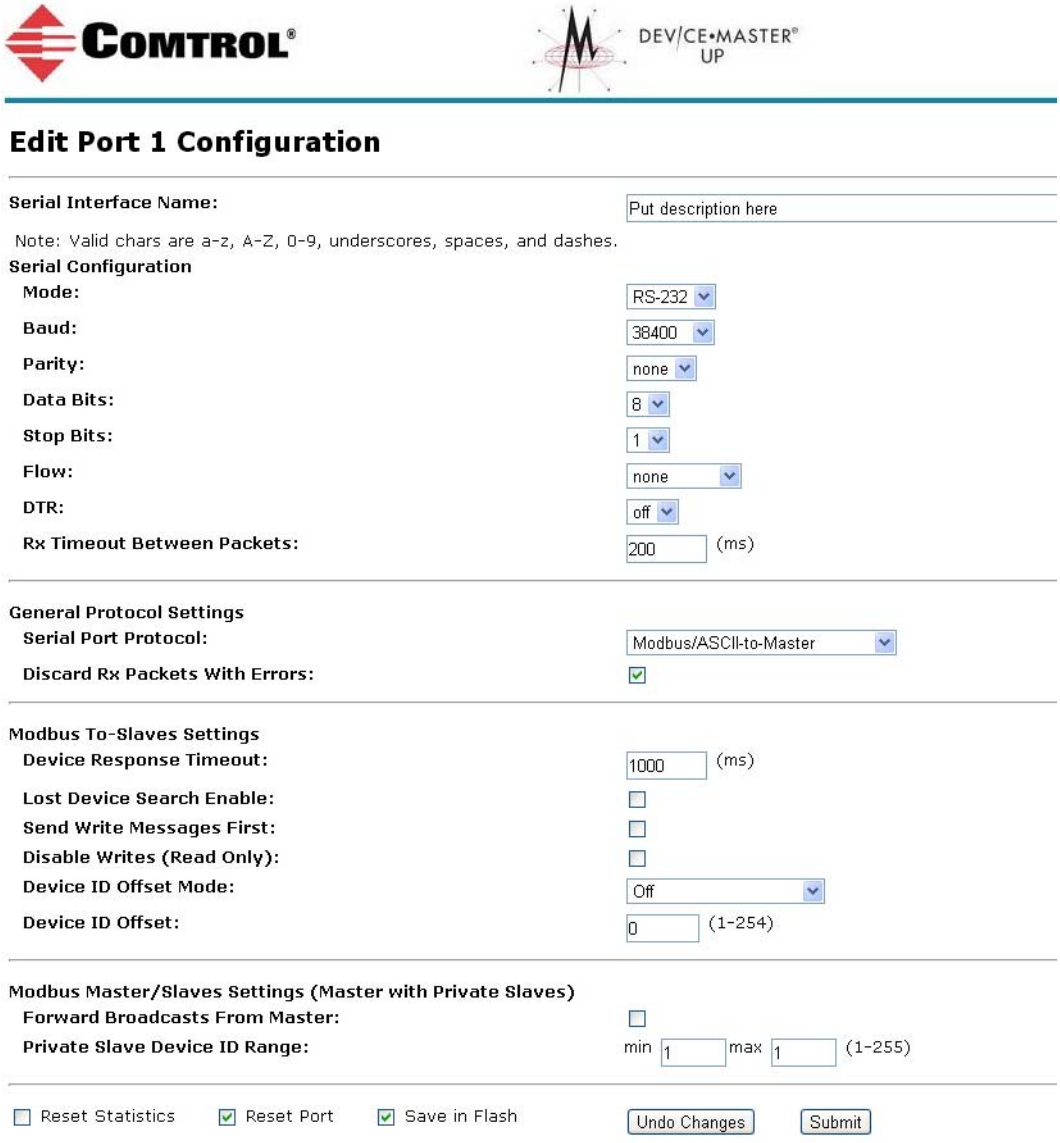
Valid Rcvd Msg Device ID Range (1-255) – (1-255): Displays the valid received message device ID range. Messages with other valid device IDs will not be transmitted out this port. (Informational only.)

Valid On Port Device ID Range (1-255) – (1-255): Displays the valid device ID range for slave devices connected to the serial port. Slave devices with other device IDs will not be accessible on this port. (Informational only.)

Click **Display All Modbus Slave Devices** for statistics for all Modbus slave devices or click **Display Devices** to review statistics for the selected port.

2.4.2. Edit Serial Port Configuration Page

Access this page from the *Serial Interface Configuration* page for each port. See the following table for information about the settings on this page.



Name	Value or Values	Edit Port Configuration Descriptions
Serial Interface Name	Up to 80 character ASCII string (default = blank)	This is a user definable string used to describe the serial interface. Valid characters include a-z, A-Z, 0-9, underscores, spaces and dashes. All other characters are discarded.
Serial Configuration		
Mode	Serial communications	1 and 4-port models: • RS-232 (default) • RS-485 • RS-422 2-port models only: • RS-232 (default) • RS-422 • RS-485_2-wire • RS-485_4-wire_Master • RS-485_4-wire_Slave
Serial Configuration (continued)		
Baud Rate	300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (default), 57600, 115200, and 230400	Selectable serial port baud rates.
Parity	None (default) Even Odd	Selectable parity.
Data Bits	5, 6, 7, 8 (default = 8)	Selectable data bits.
Stop Bits	1 or 2 (default = 1)	Selectable stop bits.
Flow	None (default) RTS/CTS XON/XOFF Half Duplex	Selectable flow control.
DTR	0=Off (default) 1=On	Selectable DTR Control.

Name	Value or Values	Edit Port Configuration Descriptions
Rx Timeout Between Packets	0-65535 (default = 200)	Receive time-out between packets in msec. This is the maximum spacing between received bytes allowed before the received Modbus serial message is expected to be complete.
General Protocol Settings		
Serial Port Protocol	Modbus/RTU-to-Slaves Modbus/ASCII-to-Slaves Modbus/RTU-to-Master Modbus/ASCII-to-Master Modbus/RTU-to-Master/ Slaves Modbus/ASCII-to-Master/ Slaves (default = Modbus/RTU-to-Slaves)	The Modbus Protocol, setting for this serial port: • **Modbus/RTU-to-Slaves** – Configures the serial port to communicate to Modbus/RTU slaves. • **Modbus/ASCII-to-Slaves** – Configures the serial port to communicate to Modbus/ASCII slaves. • **Modbus/RTU-to-Master** – Configures the serial port to communicate to a Modbus/RTU master. • **Modbus/ASCII-to-Master** – Configures the serial port to communicate to a Modbus/ASCII master. • **Modbus/RTU-to-Master/Slaves** – Configures the serial port to communicate to a serial bus with a Modbus/RTU master and Modbus/RTU slave(s). • **Modbus/ASCII-to-Master/Slaves** – Configures the serial port to communicate to a serial bus with a Modbus/ASCII master and Modbus/ASCII slaves.
Discard Rx Packets With Errors	On/Off (default = On)	If selected, the DeviceMaster UP drops all packets received with parity, framing, or overrun errors. **Note:** *Modbus/RTU messages with invalid CRCs and Modbus/ASCII messages without correct start and end of transmission characters are always discarded independent of this setting.*

Name	Value or Values	Edit Port Configuration Descriptions
Modbus To-Slaves Settings		
Device Response Timeout	0 to 65535 msec. (default = 1000 msec)	The maximum allowable time for a slave device to respond to a message before the message is considered timed out.
Lost Device Search Enable Not supported: 1-port.	On/Off (default = Off)	If selected, lost devices that were on this port are searched for on other Modbus/RTU and Modbus/ASCII slave ports that also have this option set.
Send Writes First	On/Off (default = Off)	If selected, it transmits any write messages before transmitting any read messages that may have already been queued for transmission.
Disable Writes (Read Only)	On/Off (default = Off)	If selected, it disables transmission of all standard Modbus write messages.
Device ID Offset Mode	Off (default) Add-To-Msg ID Subtract-From-Msg ID	• **Off** disables Device ID Offset functionality. • **Add-to-Msg-ID** adds the Device Offset to the message device ID. • **Subtract-from-Msg-ID** subtracts the **Device ID Offset** from the message device ID.
Device ID Offset	0 to 254	• 0 = disables Device ID Offset functionality. • 1-254 = dependent on the **Device ID Offset Mode**, is added to or subtracted from the message device ID before the Modbus message is transmitted out the serial port.
Modbus Master/Slaves Settings (Master with Private Slaves)		
Forward Broadcasts from Master	On/Off (default = Off)	If selected, all broadcast messages from the serial master will be forwarded to the Modbus network through the DeviceMaster UP.
Private Slave Device ID Range	Min = 1-255 Max = 1-255 (default: Min = 1, Max =1) **Note:** *The minimum value must be less than or equal to the maximum value.*	This range defines the expected slave device ID range on the serial bus. Modbus request messages received on this port within this device ID range will not be forwarded to the Modbus network. **Note:** *The DeviceMaster UP has a built-in auto-detect algorithm for detecting private slave device(s) with ID(s) not defined within the private device ID range.* For a more complete discussion, refer to *1.8. Modus Master/Slaves Serial Port Mode* on Page 22.

2.5. Ethernet TCP/IP Interface Configuration

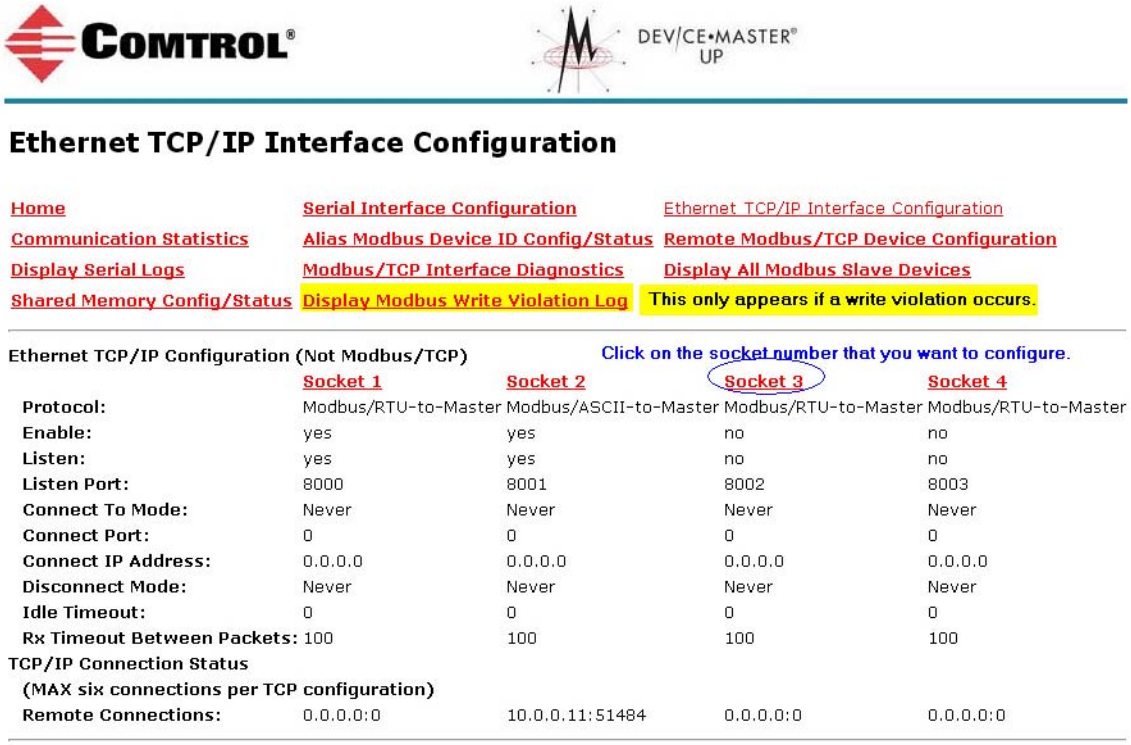
This subsection discusses the following page, which provide configuration and connection status information for the Ethernet TCP/IP interface:

Each Ethernet TCP/IP interface provides connectivity to either Modbus/RTU or Modbus/ASCII masters. All request messages received from the Ethernet TCP/IP interface are forwarded to the routing process which, in turn, forwards the messages to either local or remote Modbus slave devices.

Note: *The Ethernet TCP/IP interfaces are not directly tied to a serial port. All messages received over the Ethernet TCP/IP interface receive the same local and remote routing capabilities as messages received over the Modbus/TCP or serial Master interfaces.*

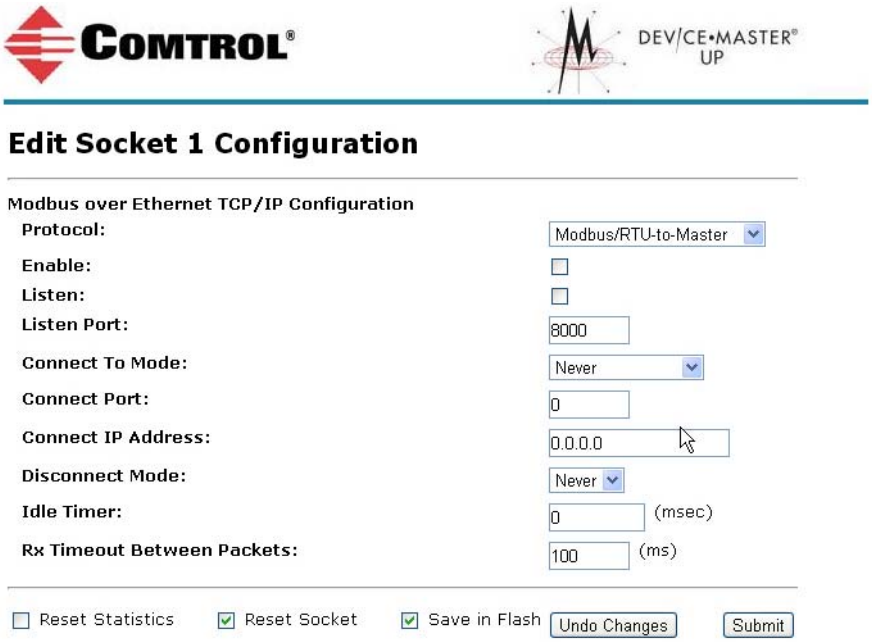
2.5.1. Ethernet TCP/IP Interface Configuration Page

This page provides access to the *Edit Socket Configuration* page when you click on the corresponding socket port. The number of Ethernet TCP/IP connections configuration is equal to the number of serial ports, but not directly tied to any particular serial port. Messages received on all Ethernet TCP/IP configurations are routed to all local and remote Modbus slave devices.



2.5.2. Edit Socket Port Configuration Page

This page is accessed from the *Ethernet TCP/IP Interface Configuration* page.The following table provides information about each configuration setting.



Name	Value or Values	Edit Socke Descriptions
Protocol	Modbus/RTU-to-Master Modbus/ASCII-to-Master	• **Modbus/RTU-to-Master** – Configures the TCP/IP connection(s) to communicate to Modbus/RTU master(s). • **Modbus/ASCII-to-Master** – Configures the TCP/IP connection(s) to communicate to Modbus/ASCII master(s).
Enable	On/Off (default = Off)	If selected, this TCP/IP socket interface will be enabled.
Listen	On/Off (default = Off)	If selected, the TCP/IP socket interface will listen for a connection at the specified **Listen Port.**
Listen Port	1-65535 defaults: Port 1=8000 Port 2=8001 Port 3=8002 Port 4=8003	If **Enable** and **Listen** are both selected, allows acceptance of: • Up to six connections from external applications if there is no active **Connect-to** connection. • Up to five connections if there is an active **Connect-to** connection. **Note:** *The defined Modbus/TCP socket port of 502 will not be accepted as valid configuration data.*

Name	Value or Values	Edit Socke Descriptions (Continued)
Connect to Mode	Never Connect-Always (default = Never)	• If **Enable** is selected, this setting determines how to connect to an application. • If **Never**: Do not attempt to make a connection. • If **Connect-Always**: Always attempt to maintain a connection to the application at **Connect IP Address** and **Connect Port**.
Connect Port	1 to 65535 (default = 0)	Socket port to connect to. Used in conjunction with **Connect to Mode** and **Connect IP Address.**
Connect IP Address	Standard IP address format: xxx.xxx.xxx.xxx	IP Address of application to create a connection. Used in conjunction with **Connect to Mode** and **Connect Port**. **Note:** *The IP Address of this DeviceMaster UP will not be accepted as valid configuration data.*
Disconnect Mode	Never Idle (default = Never)	Mode on which to disconnect from the application. • **Never** – Will not disconnect when connection(s) are idle. • **Idle** – Utilizes the Idle Timer to determine when to close the connection.
Idle Timer	1 to 65535 (default = 0)	If the **Disconnect Mode** is set to **Idle**, the idle or inactivity time when the connection(s) will be closed.
Rx Timeout Between Packets	0-65565 (default = 100)	Receive timeout between packets in msec. This is the maximum spacing between received bytes allowed before the received Modbus message is expected to be complete.

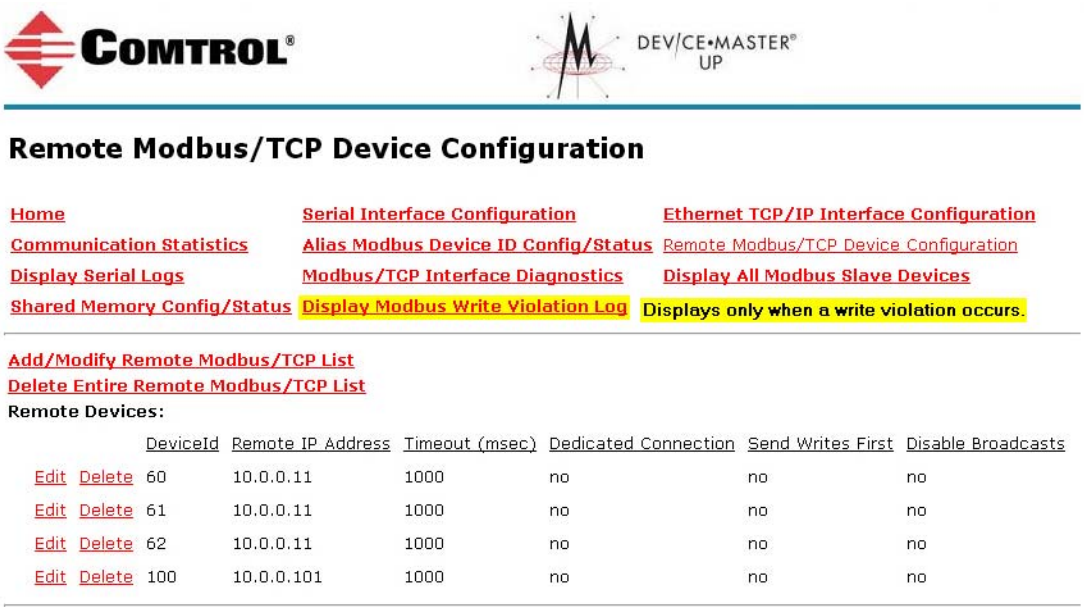
2.6. Remote Modbus/TCP Device Configuration

This subsection discusses the *Remote Device Configuration Page* and the supporting Modbus/TCP device pages:

2.6.1. Remote Device Configuration Page

This page provides access to the *Add/Modify Remote Modbus/TCP Device Configuration* and *Delete All Remote Modbus/TCP Device Configurations* pages. You can configure up to four remote devices at one time through the *Add/Modify Remote Modbus/TCP Device Configuration* page.

After configuring the remote devices, click **Edit** to make changes to a specific remote device. Click **Delete** to remove the corresponding remote device.

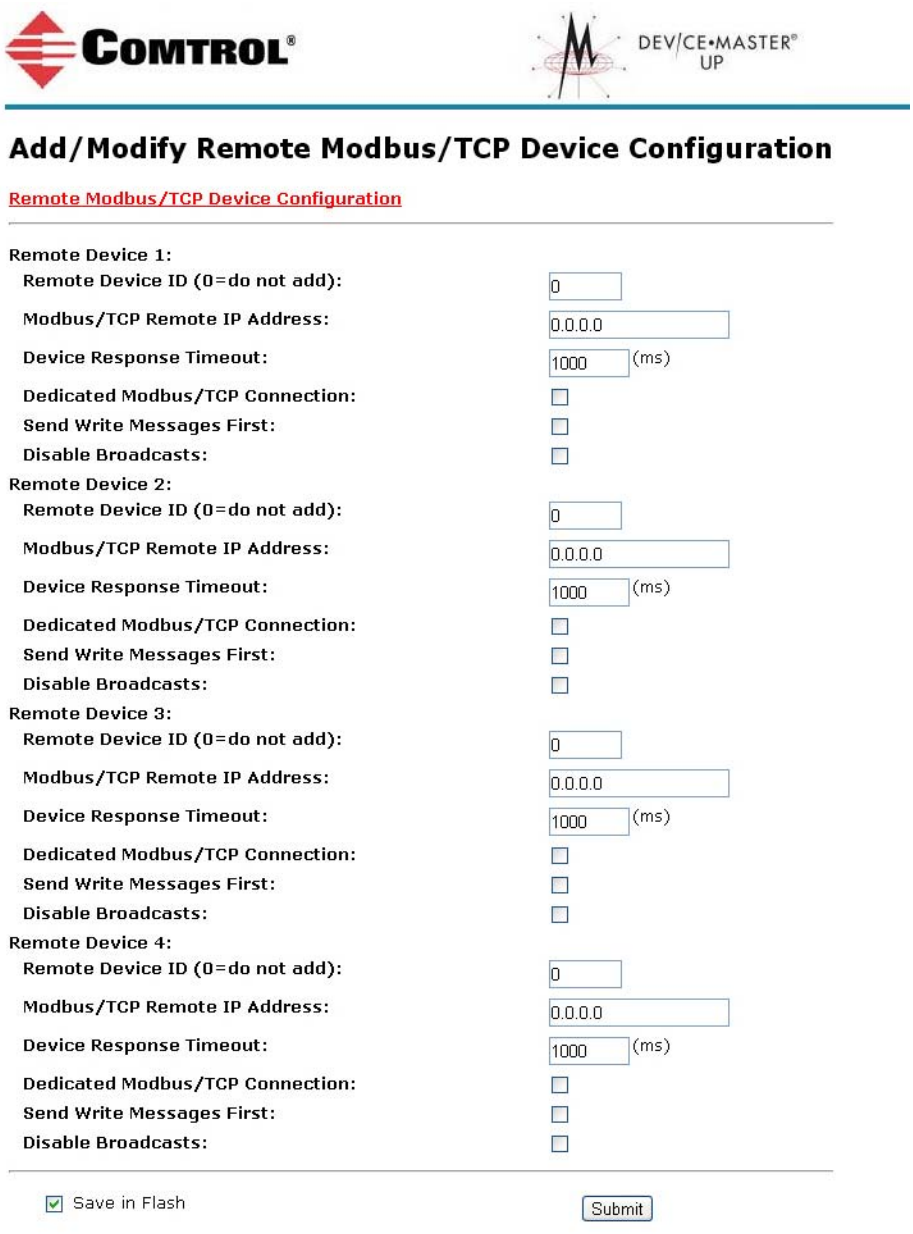


2.6.2. Add/Modify Remote Device Configuration Page

Access this page from the Remote Modbus/TCP Device Configuration page. You can configure up to four remote devices at one time. **Save in Flash** must be selected to make the configuration persistent.

All Modbus devices not configured remotely are assumed to be local, or connected to this gateway. The location of all local Modbus devices is determined automatically.

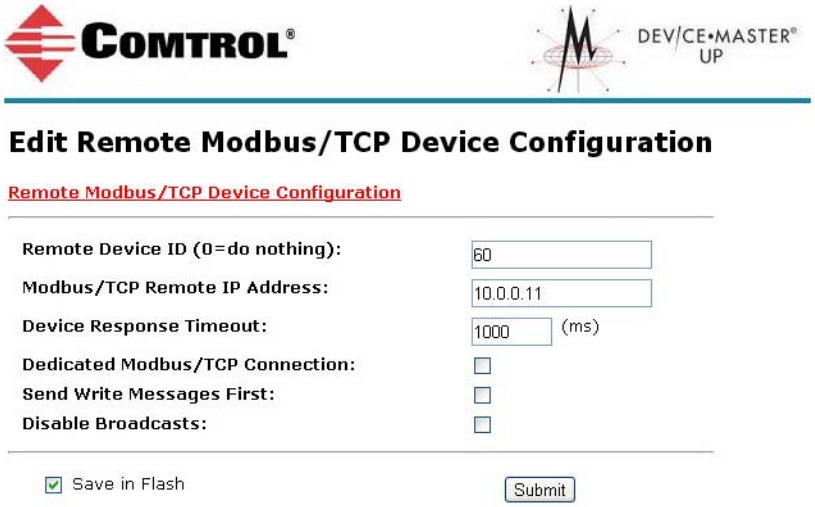
The following table provides details about configuration options.



Name	Value or Values	Add/Modify Remote Modbus/TCP Device Configuration Descriptions
Remote Device ID #	1-255 0=No configuration	The device ID (also often called the unit ID) of the remote device must be unique.
Modbus/TCP Remote IP Address	Standard IP address format: xxx.xxx.xxx.xxx 0=No configuration	IP address of the Modbus/TCP device. **Note:** *The IP address of the DeviceMaster UP will not be accepted as valid configuration data.*
Device Response Timeout	0 to 65535 msec. (default = 1000 msec)	The maximum allowable time for a slave device to respond to a message before the message is considered timed out.
Dedicated Modbus/ TCP Connection	On/Off (default = Off)	If selected, a dedicated Modbus/TCP connection will be used to connect to this remote device. **Note:** *This is most commonly used when connecting to another gateway, multiple devices are being accessed, and maximum bandwidth is desired.*
Send Write Messages First	On/Off (default = Off)	If selected, will forward write messages before forwarding any pending read messages. **Note:** *This is most commonly used when multiple messages may be outstanding for the remote device(s) and low latency for write messages is desired.*
Disable Broadcasts	On/Off (default = Off)	If selected, will disable broadcasts to this remote device. **Note:** *If multiple remote devices are accessed through another gateway, then this option must be selected for all remote devices configured to that gateway to prevent broadcast messages from being sent to those devices.*

2.6.3. Edit Remote Device Configuration Page

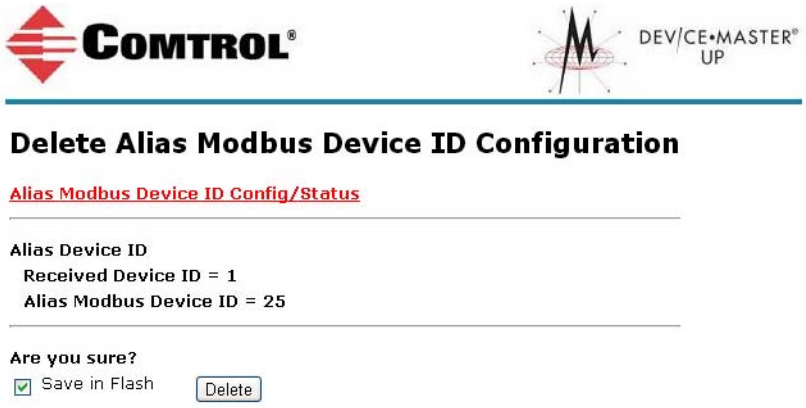
Use this page to edit an existing remote device configuration. Access this page from the *Remote Modbus/TCP Device Configuration* page and click **Edit** next to the remote device that you want to modify.



See *2.6.2. Add/Modify Remote Device Configuration Page* on Page 39 for detailed information about the entries on this page.

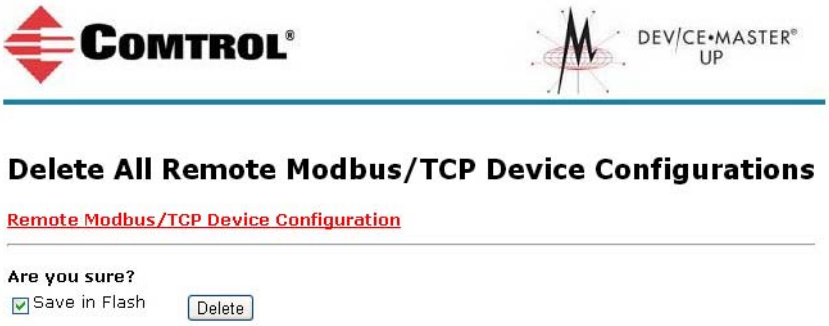
2.6.4. Delete Remote Device Configuration Page

This page allows deletion of the selected remote device configuration. Access this page by clicking the **Delete** next to the remote device that you wish to remove on the *Remote Modbus/TCP Device Configuration* page and then click the **Delete** button.



2.6.5. Delete All Remote Devices Configuration Page

This page allows deletion of all remote device configurations. Click **Delete Entire Remote/TCP List** on the *Remote Modbus/TCP Device Configuration* page and the click the **Delete** button.



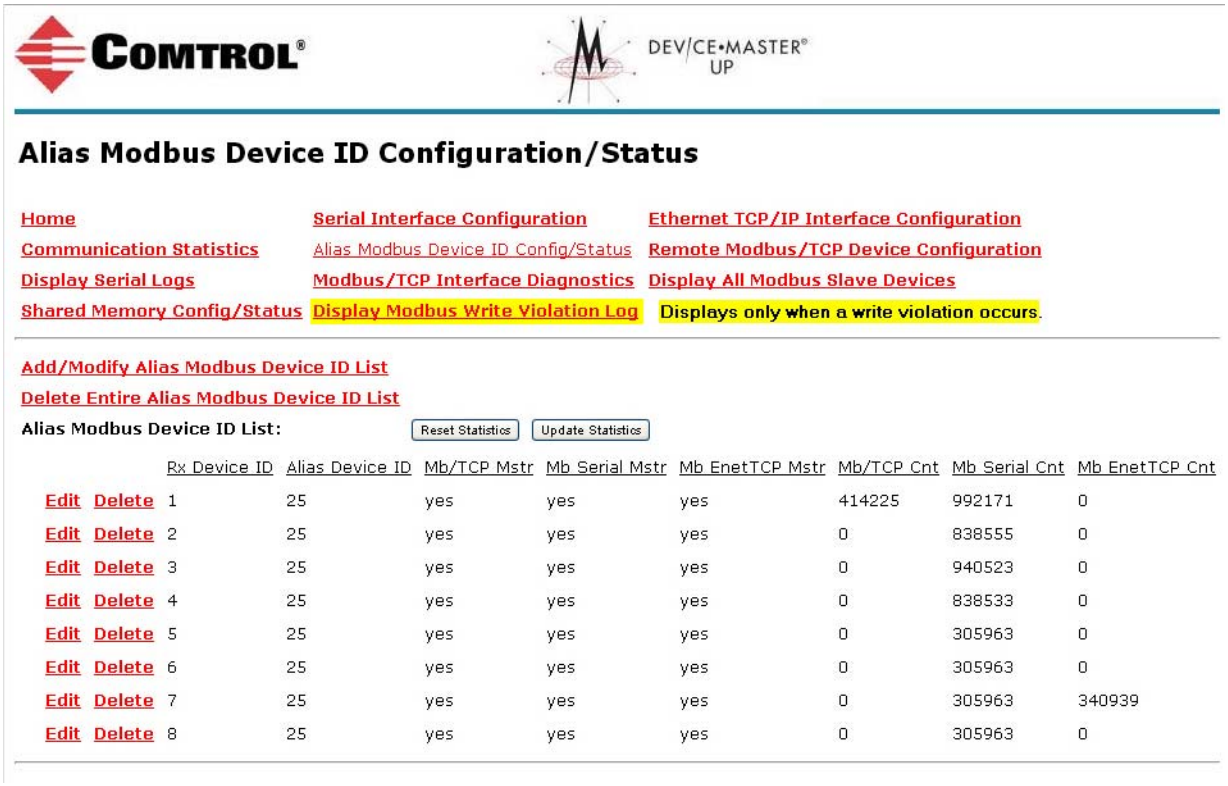
2.7. Alias Modbus Device ID Configuration/Status

This subsection discusses the *Alias Modbus Device ID Configuration / Status* page and supporting pages:

2.7.1. Alias Modbus Device ID Configuration/Status Page

This subsection discusses the *Alias Modbus Device ID Configuration / Status* page, which provides access to the following pages:

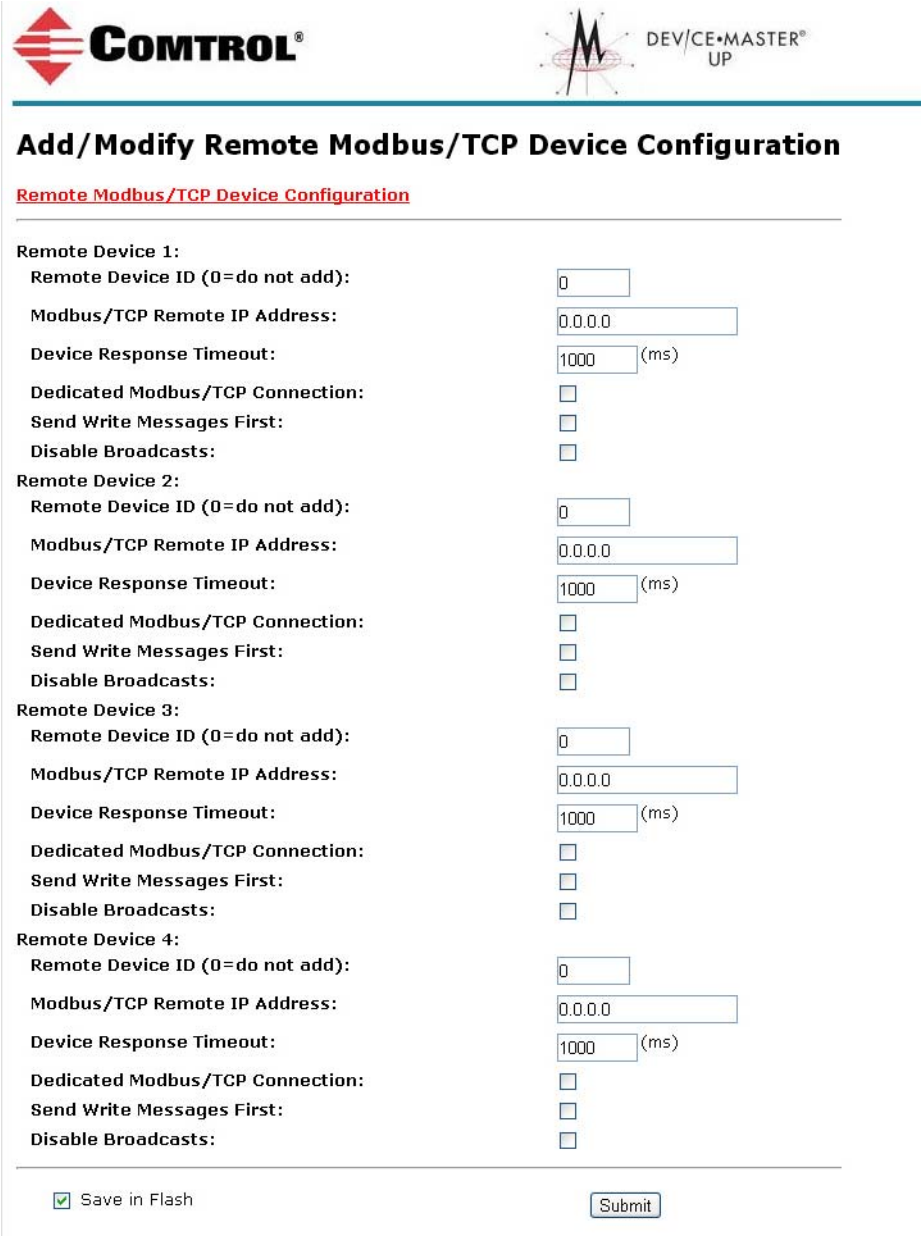
- Add/Modify Alias Modbus Device ID List
- Delete Entire Alias Modbus Device ID List



Name	Alias Modbus Device ID Configuration Descriptions
Mb/TCP Cnt	The number of Alias conversions performed for this configuration to messages received from Modbus/TCP masters.
Mb Serial Cnt	The number of Alias conversions performed for this configuration to messages received from serial Modbus masters.
Mb EnetTCP Cnt	The number of Alias conversions performed for this configuration to messages received from Modbus over Ethernet TCP/IP masters.

2.7.2. Add/Modify Alias Device ID Configuration Page

Use this web page to add or modify an Alias Device ID.



Where:

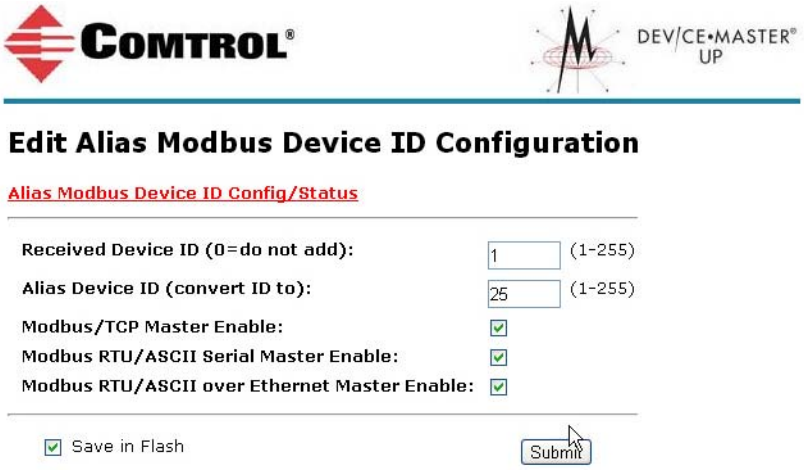
- Up to four alias device IDs may be configured at one time.
- A received or alias device ID of zero will indicate no configuration.
- **Save in Flash** must be selected to make the configuration persistent.

The following configuration options apply:

Name	Value(s)	Add/Modify Alias Device ID Configuration Descriptions
Received Device ID	1-255	The device ID (also often called the unit ID) of the received message from a master.
Alias Device ID	1-255	The alias device ID to convert the received device ID to.
Modbus/TCP Master Enable	On/Off (Default=Off)	If selected, this applies the alias device ID configuration to messages received from Modbus/TCP masters.
Modbus RTU/ASCII Serial Master Enable	On/Off (Default=Off)	If selected, this applies the alias device ID configuration to messages received from serial Modbus masters.
Modbus RTU/ASCII over Ethernet Master Enable	On/Off (Default=Off)	If selected, this applies the alias device ID configuration to messages received from Modbus RTU/ASCII over Ethernet TCP/IP masters.

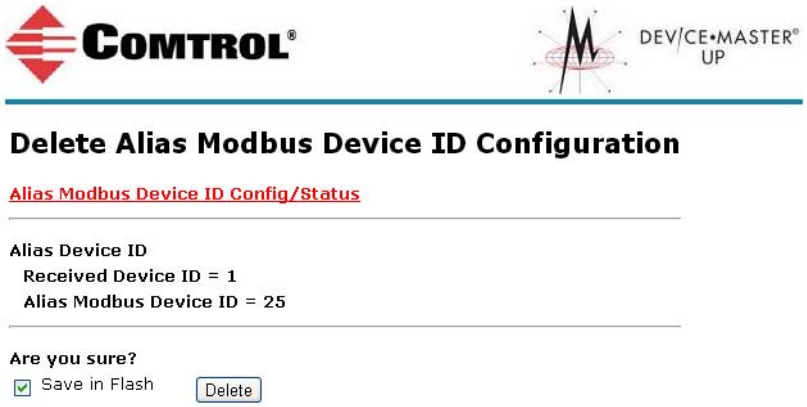
2.7.3. Edit Alias Device ID Configuration Page

This web page can be used to edit a current alias device ID configuration.



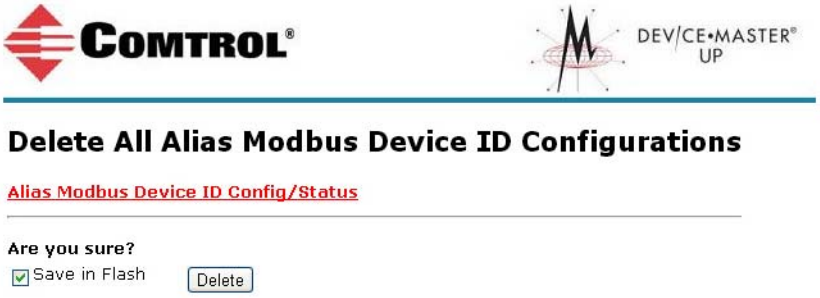
2.7.4. Delete Alias Device ID Configuration Page

This web page allows the deletion of the selected alias device ID configuration.



2.7.5. Delete All Alias Device ID Configurations Page

This web page allows deletion of all alias device ID configurations.



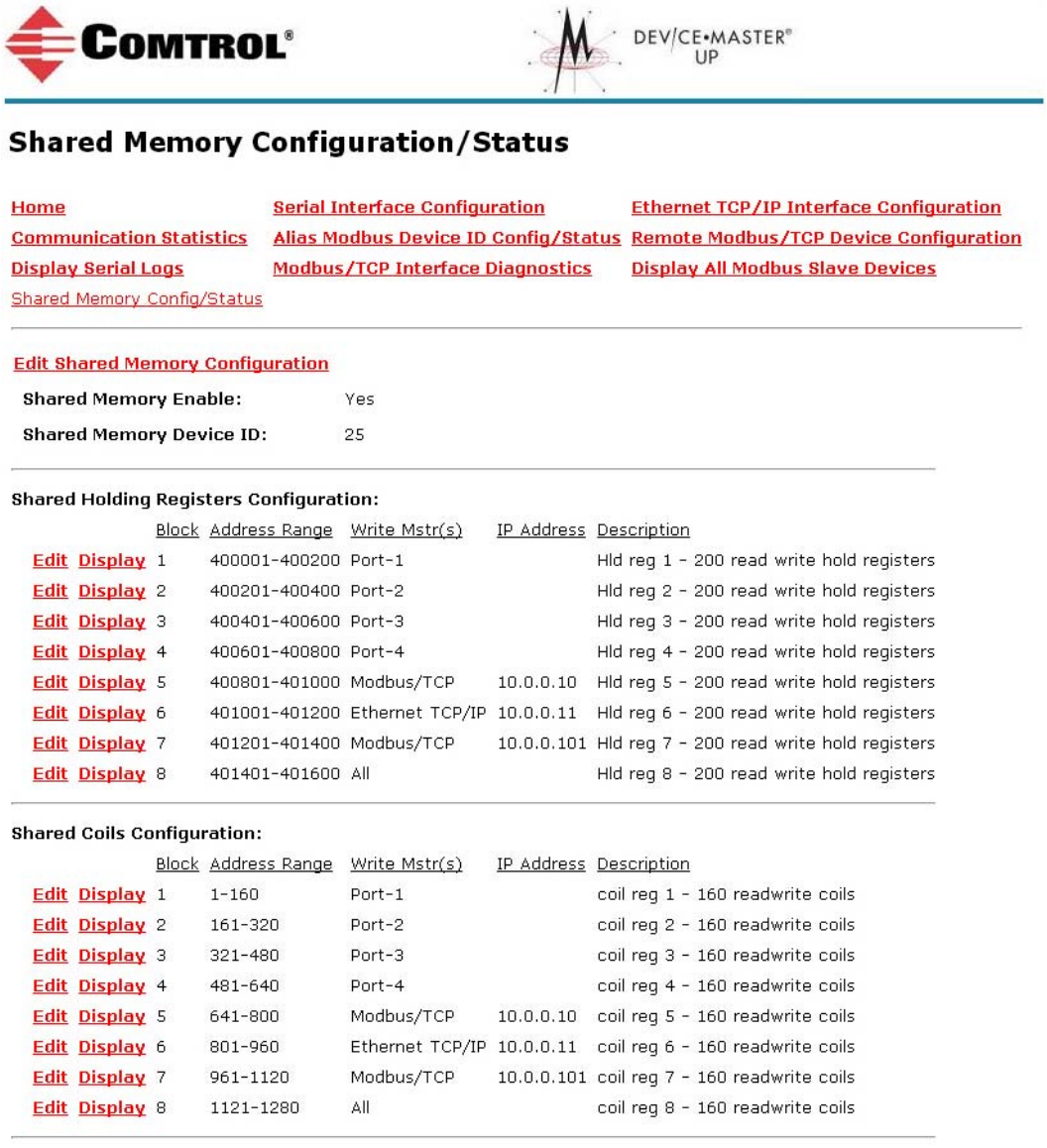
2.8. Shared Memory Configuration/Status Page

This subsection discusses the *Shared Memory Configuration / Status* page and the supporting Modbus/TCP device pages:

You can configure the entire shared memory through the *Edit Shared Memory Configuration* page.

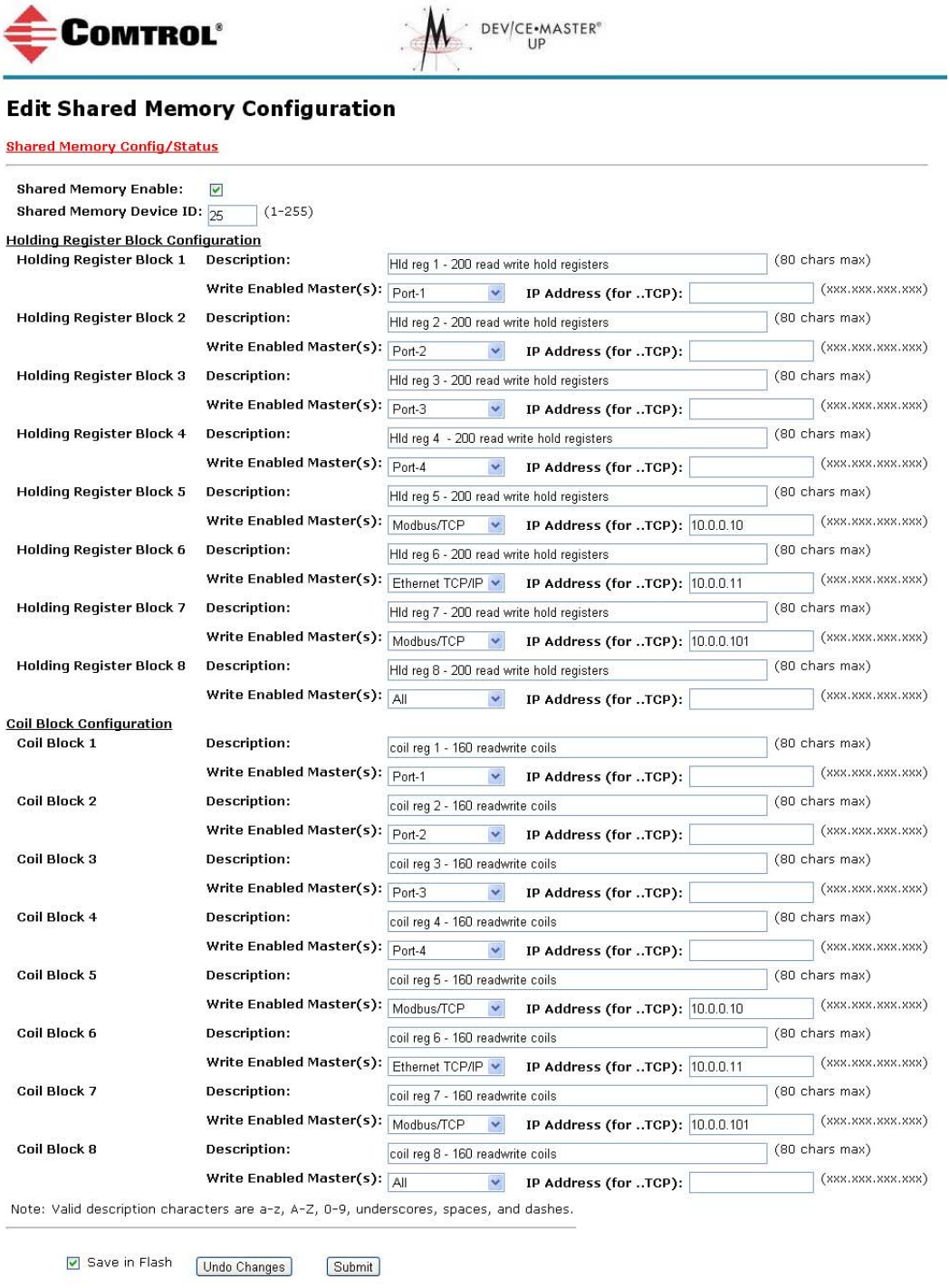
Shared Memory functionality can only be enabled and the Shared Memory Device ID can only be set through the *Edit Shared Memory Configuration* page.

After configuring the shared memory, click **Edit** to make changes to a specific Holding Register or Coil block. Click **Display** to display the contents and diagnostics for any of the Holding Register or Coil blocks.



2.8.1. Edit Shared Memory Configuration Page

Access this page from the *Shared Memory Configuration / Status* page. **Save in Flash** must be selected to make the configuration persistent.



The following table provides details about the configuration options.

Name	Value or Values	Add/Modify Remote Modbus/TCP Device Configuration Descriptions
Shared Memory Enable	On/Off (Default=Off)	If selected, enables the Shared Memory functionality.
Shared Memory Device ID	1-255 (Default=252)	The device ID (also often called the unit ID) of the Shared Memory must be unique within the public Modbus network.
Description	ASCII string	User-defined description of the Shared Memory block. Maximum of 80 characters in length.
Write Enabled Master(s)	All or Master Specific (Default = All)	Indicates which master(s) have write access to the Shared Memory block. • **All** – All masters have write access to the block • Port specific serial master: - **Port-1** - **Port-2** (2-port and 4-port models only) - **Port-3** (4-port models only) - **Port-4** (4-port models only) • Ethernet based masters: - **Modbus/TCP** - Modbus/TCP master(s) at a specified IP address - **Ethernet TCP/IP** – Ethernet TCP/IP master(s) at a specified IP address
IP Address (for …TCP)	Standard IP address format: xxx.xxx.xxx.xxx 0=No configuration	IP address of the Modbus/TCP or Ethernet TCP/IP master. **Note:** *Does not apply to All or port-specific serial masters.*

The following table displays the supported Holding Register Block function codes.

Function Code	Description
3	Read Holding Registers
6	Write Single Register
16	Write Multiple Registers
22	Write Mask Register
23	Read Write Registers

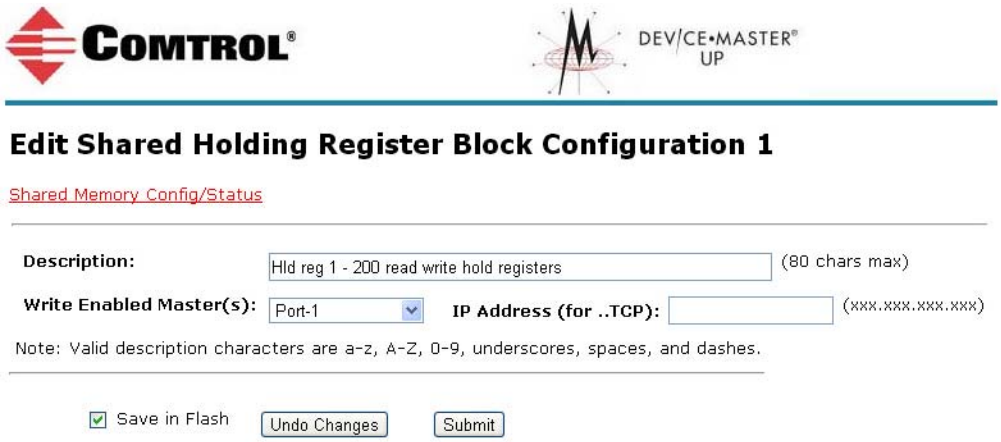
The following table shows supported the Coil Block function codes.

Function Code	Description
1	Read Coils
5	Write Single Coil
15	Write Multiple Coils

2.8.2. Edit Shared Holding Register Block Configuration Page

Use this page to edit an individual Holding Register block configuration. Access this page from the *Shared Memory Configuration/Status* page and click **Edit** next to the configuration that you want to modify.

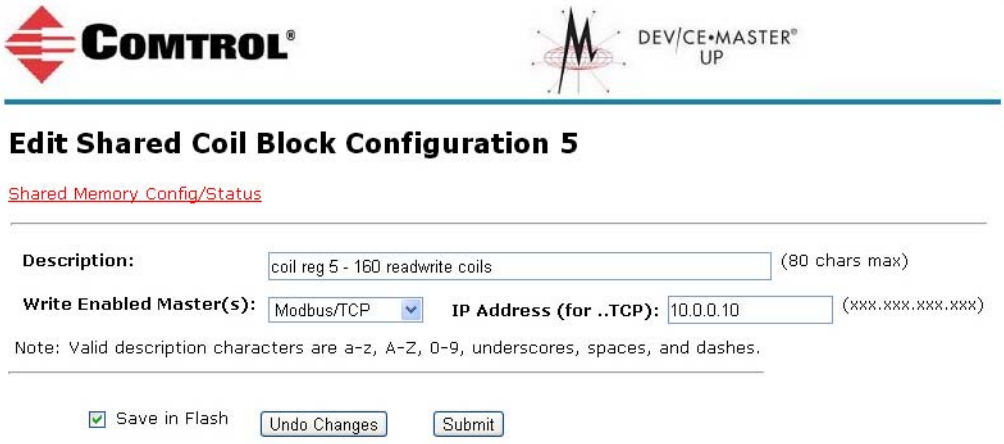
See *2.8.1. Edit Shared Memory Configuration Page* on Page 48 for information about the entries on this page.



2.8.3. Edit Shared Coil Block Configuration Page

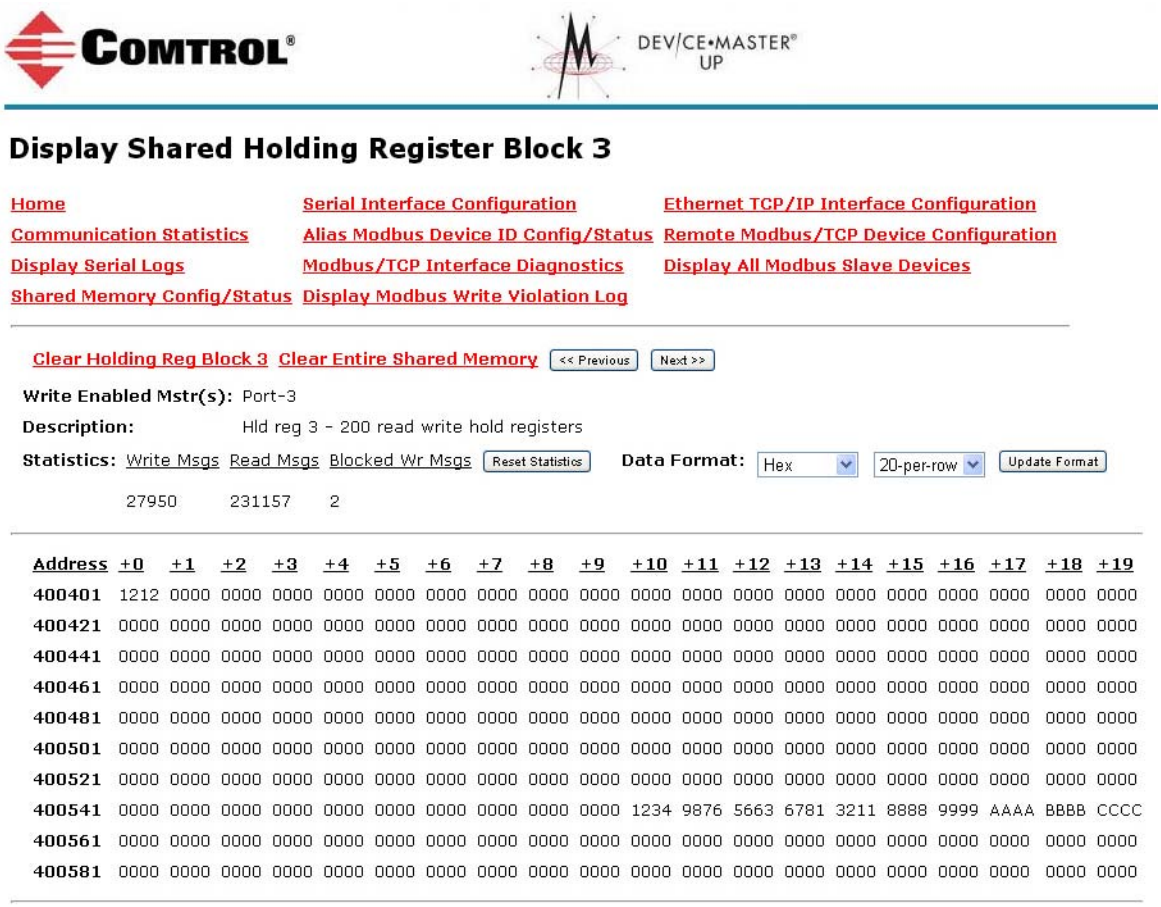
Use this page to edit an individual Coil block configuration. Access this page from the *Shared Memory Configuration/Status* page and click **Edit** next to the configuration that you want to modify.

See *2.8.1. Edit Shared Memory Configuration Page* on Page 48 for information about the entries on this page.



2.8.4. Display Shared Holding Register Block Page

This page displays the contents of a Shared Holding Register block. Access this page from the *Shared Memory Configuration / Status* page and click **Display** next to the Holding Register block that you want to display.

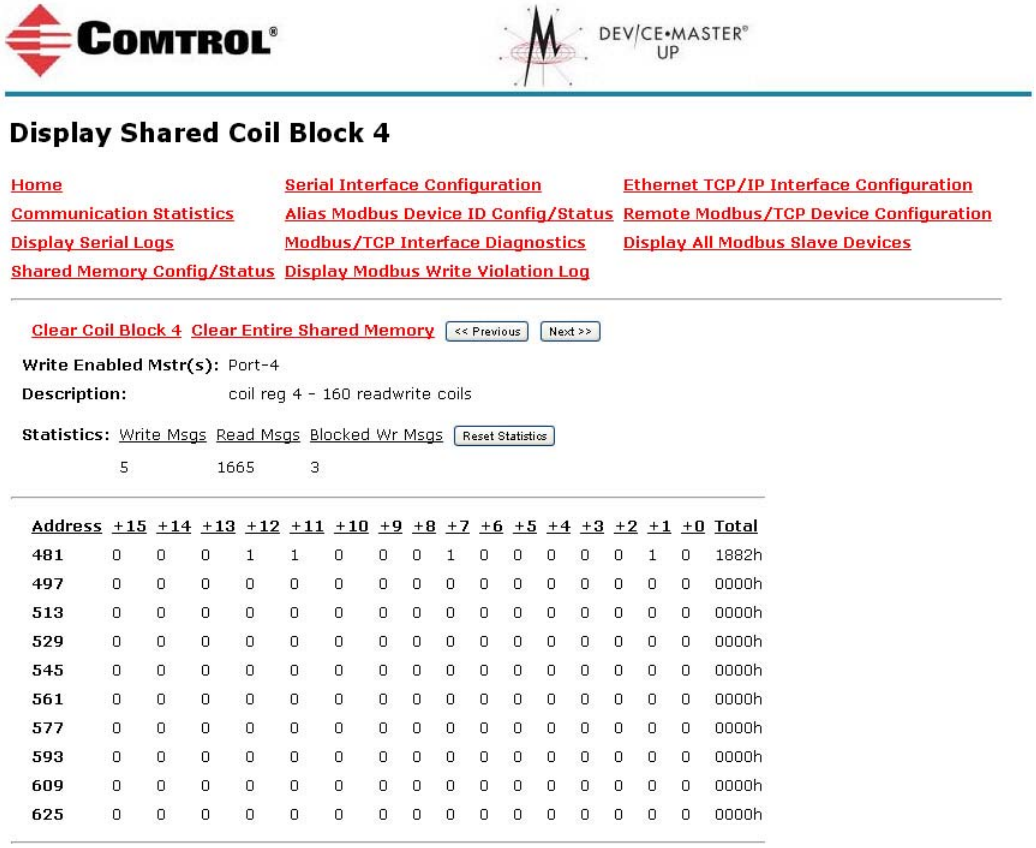


Name	Description
Clear Holding Reg Block	Click on this link to clear the diagnostics and data contents of this Holding Register block.
Clear Entire Shared Memory	Click on this link to clear the diagnostics and data contents of all Holding Register and Coil blocks.
Previous	Click on this button to display the previous Shared Memory block.
Next	Click on this button to display the next Shared Memory block.
Rx Rsp	Displays the number of Modbus responses received from this device.
Write Msgs	Displays the number of successful write messages to this Shared Memory block.
Read Msgs	Displays the number of read messages addressed to this Shared Memory block.

Name	Description
Blocked Wr Msgs	Displays the number of write messages that were blocked, or prevented, from writing to this Shared Memory block. **Note:** *Blocked writes are treated as write violations and are added to the Write Violation Log.*
Reset Statistics	Click on this button to clear the diagnostics for this Holding Register block.
Data Format	Selectable data format and entries per row to display the contents of the Holding Register block data. • Data formats: - Hex - 16 bit word (default) - word-16 - unsigned 16 bit decimal - word-32 - unsigned 32 bit decimal - string - ASCII character string • Width: - 10-per-row – ten entries display per row - 20-per-row – twenty entries displayed per row (default)
Update Format	Click on this button to update the displayed data format.

2.8.5. Display Shared Coil Block Page

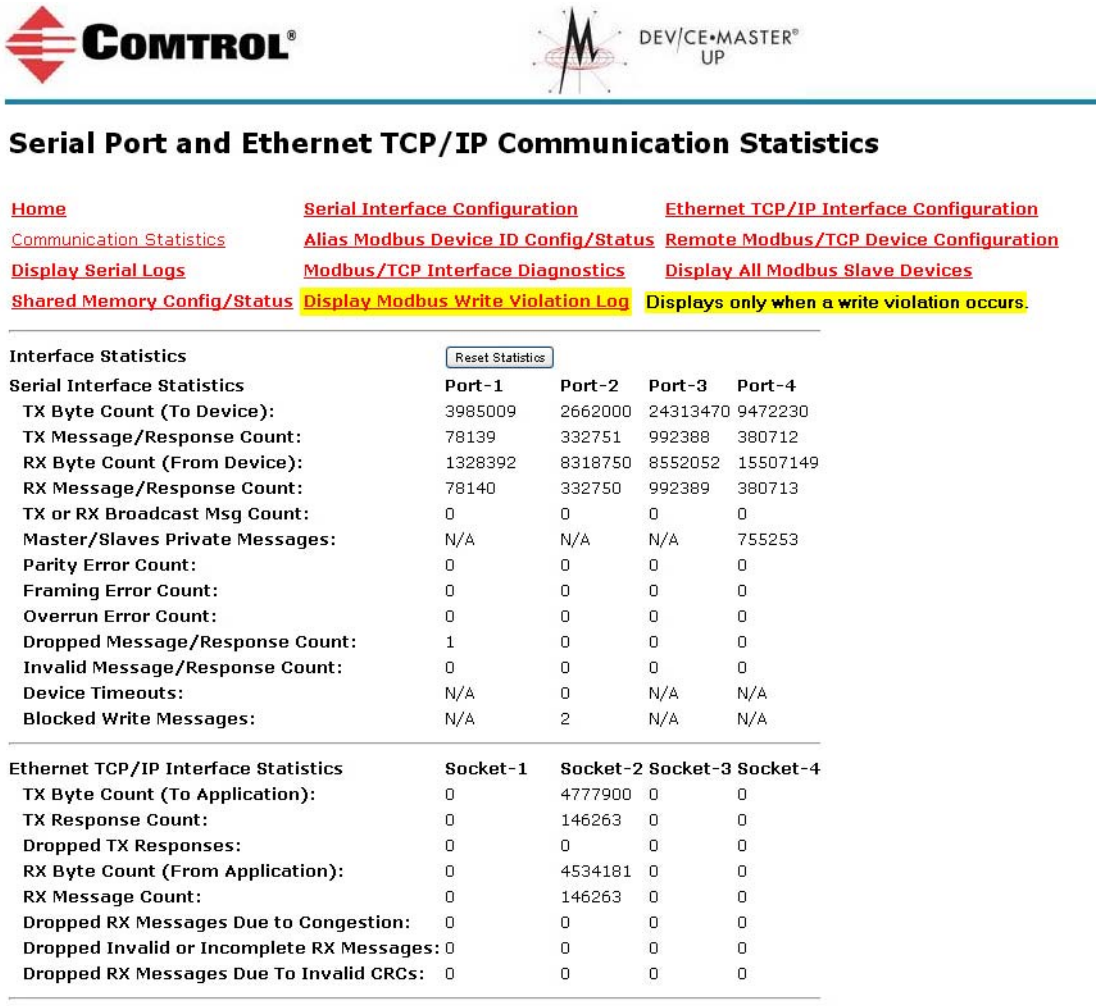
This page displays the contents of a Shared Coil block. Access this page from the *Shared Memory Configuration / Status* page and click **Display** next to the Coil block that you want to display.



Name	Description
Clear Coil Block	Click on this link to clear the diagnostics and data contents of this Coil block.
Clear Entire Shared Memory	Click on this link to clear the diagnostics and data contents of all Holding Register and Coil blocks.
Previous	Click on this button to display the previous Shared Memory block.
Next	Click on this button to display the next Shared Memory block.
Write Msgs	Displays the number of successful write messages to this Shared Memory block.
Read Msgs	Displays the number of read messages addressed to this Shared Memory block.
Blocked Wr Msgs	Displays the number of write messages that were blocked, or prevented, from writing to this Shared Memory block. **Note:** *Blocked writes are treated as write violations and are added to the Write Violation Log.*
Reset Statistics	Click on this button to clear the diagnostics for this Coil block.

2.9. Serial Port and Ethernet TCP/IP Communication Statistics Page

Click **Communication Statistics** to access the *Serial Port and Ethernet TCP/IP Communication Statistics* page. The following table provides detailed information about the *Serial Port and Ethernet TCP/IP Communications Statistics* page.



COMTROL®

DEVICE•MASTER® UP

Serial Port and Ethernet TCP/IP Communication Statistics

Home | Serial Interface Configuration | Ethernet TCP/IP Interface Configuration
Communication Statistics | Alias Modbus Device ID Config/Status | Remote Modbus/TCP Device Configuration
Display Serial Logs | Modbus/TCP Interface Diagnostics | Display All Modbus Slave Devices
Shared Memory Config/Status | Display Modbus Write Violation Log | Displays only when a write violation occurs.

Interface Statistics — Reset Statistics

Serial Interface Statistics	Port-1	Port-2	Port-3	Port-4
TX Byte Count (To Device):	3985009	2662000	24313470	9472230
TX Message/Response Count:	78139	332751	992388	380712
RX Byte Count (From Device):	1328392	8318750	8552052	15507149
RX Message/Response Count:	78140	332750	992389	380713
TX or RX Broadcast Msg Count:	0	0	0	0
Master/Slaves Private Messages:	N/A	N/A	N/A	755253
Parity Error Count:	0	0	0	0
Framing Error Count:	0	0	0	0
Overrun Error Count:	0	0	0	0
Dropped Message/Response Count:	1	0	0	0
Invalid Message/Response Count:	0	0	0	0
Device Timeouts:	N/A	0	N/A	N/A
Blocked Write Messages:	N/A	2	N/A	N/A

Ethernet TCP/IP Interface Statistics	Socket-1	Socket-2	Socket-3	Socket-4
TX Byte Count (To Application):	0	4777900	0	0
TX Response Count:	0	146263	0	0
Dropped TX Responses:	0	0	0	0
RX Byte Count (From Application):	0	4534181	0	0
RX Message Count:	0	146263	0	0
Dropped RX Messages Due to Congestion:	0	0	0	0
Dropped Invalid or Incomplete RX Messages:	0	0	0	0
Dropped RX Messages Due To Invalid CRCs:	0	0	0	0

Counter Name	Serial Port and Ethernet TCP/IP Communication Descriptions
Serial Interface Statistics	
TX Byte Count (To Device)	Displays the number of bytes transmitted out the serial port.
TX Message/Response Count	Displays the number of messages or responses transmitted out the serial port.
RX Byte Count (From Device)	Displays the number of bytes received on the serial port.
RX Message/Response Count	Displays the number of messages or responses received on the serial port.
TX or RX Broadcast Msg Count	Displays the number of broadcast messages transmitted out the serial port.

Counter Name	Serial Port and Ethernet TCP/IP Communication Descriptions
Master/Slaves Private Messages	Displays the number of private messages detected, those between a master and private slave(s), on a serial port configured in Master/Slaves mode.
Parity Error Count	Displays the number of parity errors received on the serial port. Typically occurs due to an incorrect parity setting.
Framing Error Count	Displays the number of framing errors received on the serial port. Typically occurs due to an incorrect baud rate or stop bit setting.
Overrun Error Count	Displays the number of overrun errors received on the serial port. This typically occurs to one of the following events: incorrect flow control, incorrect baud rate, incorrect data size, or incorrect stop bit setting.
Dropped Message/Response Count	Displays the number of messages or responses dropped to any of the following: • Incomplete message or response. • Did not receive valid start and/or end characters (Modbus/ASCII only).
Invalid Message/Response Count	Displays the number of invalid messages or responses received to any of the following events: • Message received after the timeout period. This may require increasing the **Device Response Timeout**. • Incorrect device ID in response message. • Incorrect function code in response message.
Device Timeouts	Displays the number of device timeouts that occurred when there was no response for a Modbus message.
Blocked Write Messages	Displays the number of Modbus write messages that were not transmitted as a result of the **Disable Writes (Read Only)** option being set.
Ethernet TCP/IP Interface Statistics	
TX Byte Count (To Application)	Displays the number of bytes transmitted out of the TCP/IP connection(s).
TX Response Count	Displays the number of responses transmitted out of the TCP/IP connection(s).
Dropped TX Responses	Displays the number of responses that were intended to be transmitted out the TCP/IP connection(s) but could not be and were dropped. This typically occurs when one or more connections close unexpectedly.
RX Byte Count (From Application)	Displays the number of bytes received on the TCP/IP connection(s).
RX Message Count	Displays the number of messages received on the TCP/IP connection(s).
Dropped RX Messages Due to Congestion	Displays the number of messages that were dropped to the gateway being overly congested. This typically occurs when the application(s) send messages faster than the slave device(s) can respond.

Counter Name	Serial Port and Ethernet TCP/IP Communication Descriptions
Dropped Invalid or Incomplete RX Messages	Displays the number of messages from the application(s) that were dropped to: • Containing an invalid Modbus message format. • Containing an incomplete Modbus message.
Dropped RX Messages Due to Invalid CRCs	Displays the number of messages from the application(s) that were dropped due to an invalid Modbus/RTU CRC or Modbus/ASCII LRC.

2.10. Modbus/TCP Diagnostics Interface Page

Click **Modbus/TCP Interface Diagnostics** to access this page. The following table provides information about this page.

Modbus/TCP Interface Diagnostics

Home | Serial Interface Configuration | Ethernet TCP/IP Interface Configuration
Communication Statistics | Alias Modbus Device ID Config/Status | Remote Modbus/TCP Device Configuration
Display Serial Logs | Modbus/TCP Interface Diagnostics | Display All Modbus Slave Devices
Shared Memory Config/Status | Display Modbus Write Violation Log | Displays only when a write violation error occurs.

Modbus/TCP Statistics	Reset Statistics
Slave Mode Specific Statistics	
Messages Received From Modbus/TCP Master(s):	1469946
Responses Sent To Modbus/TCP Master(s):	1469947
Modbus Broadcasts Received From Modbus/TCP Master:	0
Invalid Command Lengths:	0
Invalid Message Data Errors:	11
Invalid Request Protocol Types:	0
Master Mode Specific Statistics	
Messages Sent To Modbus/TCP Slave(s):	1451455
Responses Received From Modbus/TCP Slave(s):	1451455
Invalid Response Data Errors From Modbus/TCP Slave(s):	0
Remote Modbus/TCP Device Timeouts:	0
Unexpected Responses From Modbus/TCP Slave(s):	0
Error Responses From Modbus/TCP Slave(s):	0
Unexpected Response Function Codes From Modbus/TCP Slave(s):	0
Invalid Response Protocol Types From Modbus/TCP Slave(s):	0
Failed Modbus/TCP Connection Attempts To Modbus/TCP Slave(s):	0
Modbus/TCP Connection Problems:	0
Unexpected Dropped Connections:	0
Non-Mode Specific Statistics/Diagnostics	
No Available Modbus/TCP Connection Errors:	0
Improper Configuration Errors:	2
System Resource Errors:	0
First Error Description:	Shared Memory Modbus Write Request to Protected Coil, coil 450
Last Error Description:	Shared Memory Modbus Write Request to Protected Holding Register, address 1003

Note: *The Modbus/TCP interface uses the standard socket port of 502.*

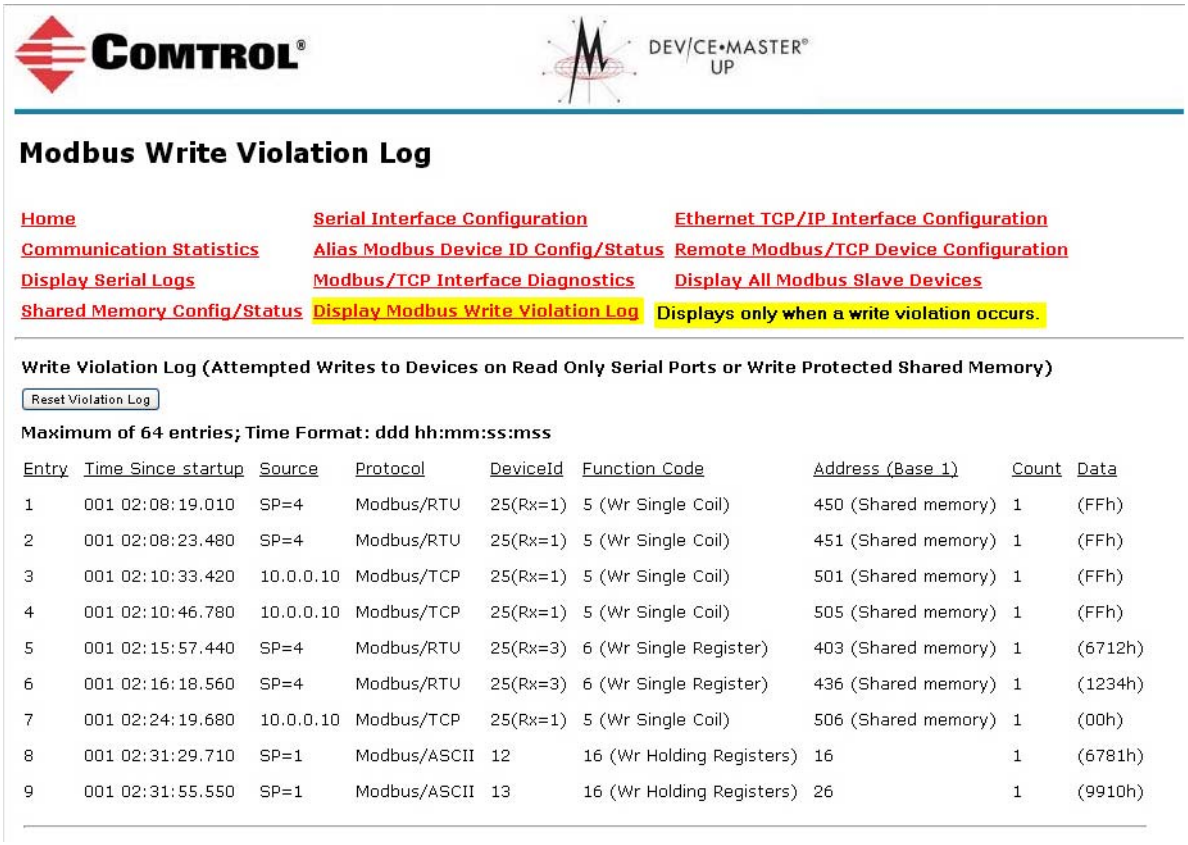
Item	Modbus/TCP Interface Diagnostics Descriptions
Slave Mode Specific Statistics	
Messages Received From Modbus/TCP Master(s)	Displays the total number of Modbus/TCP messages received from Modbus/TCP master(s).
Responses Sent to Modbus/TCP Master(s)	Displays the total number of Modbus/TCP responses sent to Modbus/TCP master(s).
Modbus Broadcasts Received From Modbus/TCP Masters	Displays the number of Modbus broadcast messages received from Modbus/TCP Masters.

Item	Modbus/TCP Interface Diagnostics Descriptions (Continued)
Slave Mode Specific Statistics (Continued)	
Invalid Command Lengths	Displays the number of messages received with invalid command lengths.
Invalid Message Data Errors	Displays the number of messages received with invalid message data errors. These errors occur when the DeviceMaster UP receives a message that cannot be processed due to improper message data.
Invalid request Protocol Types	Displays the number of messages received with invalid protocol errors. This occurs when a message is received with a protocol other than the Modbus/TCP protocol value of zero.
Master Mode Specific Statistics	
Messages Sent To Modbus/TCP Slave(s)	Displays the total number Modbus messages sent to remote Modbus/TCP slaves.
Responses Received From Modbus/TCP Slave(s)	Displays the total number of Modbus responses received from the Modbus/TCP Slave(s).
Invalid Response Data Errors From Modbus/TCP Slave(s)	Displays the number of response data errors to polling requests returned from the Modbus/TCP Slave(s). Possible causes include: • Incorrect transaction ID incorrect. • Message command length to large. • Incorrect device Id in response.
Remote Modbus/TCP Device Timeouts	Displays the number of messages to remote devices that were determined to have timed out by this gateway.
Unexpected Responses From Modbus/TCP Slaves	Displays the number of responses received when no response was expected.
Error Responses from Modbus/TCP Slave(s)	Displays the number of responses received from Modbus/TCP slaves with errors indicated. This may be caused by such things as: • Device timeouts detected by slave Modbus/TCP device, such a gateway. • Invalid device address. • Invalid device ID. • Invalid message data.
Unexpected Response Function Codes From Modbus/TCP Slave(s)	Displays the number of unexpected response function codes from Modbus/TCP slaves. This occurs when a response was received with a different function code than what was sent.
Invalid Response Protocol Types From Modbus/TCP Slave(s)	Displays the number of responses with invalid protocol errors. This occurs when a response is returned with a protocol other than the Modbus/TCP protocol value of zero.
Failed Modbus/TCP Connection Attempts to Modbus/TCP Slave(s)	Displays the number of failed Modbus/TCP connection attempts to the specified PLC IP address.

Item	Modbus/TCP Interface Diagnostics Descriptions (Continued)
Master Mode Specific Statistics (continued)	
Modbus/TCP Connection Problems	Displays the number of Modbus/TCP connection attempt problems. This occurs when the device responds and the connection is made, but there are problems setting up the connection options. The possible option problems include: • Setting the TCP connection to **TCP_NODELAY**. • Setting the socket connection to **SO_OOBINLINE**. • Setting the socket connection to **SO_KEEPALIVE**.
Unexpected Dropped Connections	Displays the number of Modbus/TCP connections that were unexpectedly dropped.
Non-Mode Specific Statistics/Diagnostics	
No Available Modbus/TCP Connection Errors	Displays the number of connections aborted when there are no available Modbus/TCP connections. This error occurs when the maximum number of Modbus/TCP connections has been reached and the DeviceMaster UP is attempting to form another Modbus/TCP connection.
Improper Configuration Errors	Displays the number of errors that were caused by improper configuration errors.
System Resource Errors	Displays the numbers of system resource errors. These errors are typically caused by congestion and/or non-responding devices.
First Error Description	Displays the first error detected.
Last Error Description	Displays the last or most recent error detected.

2.11. Modbus Write Violation Page

This web page provides a detailed log of attempts to write to Modbus slave device attached to serial ports that have been configured with the **Disable Writes (Read Only)** option selected

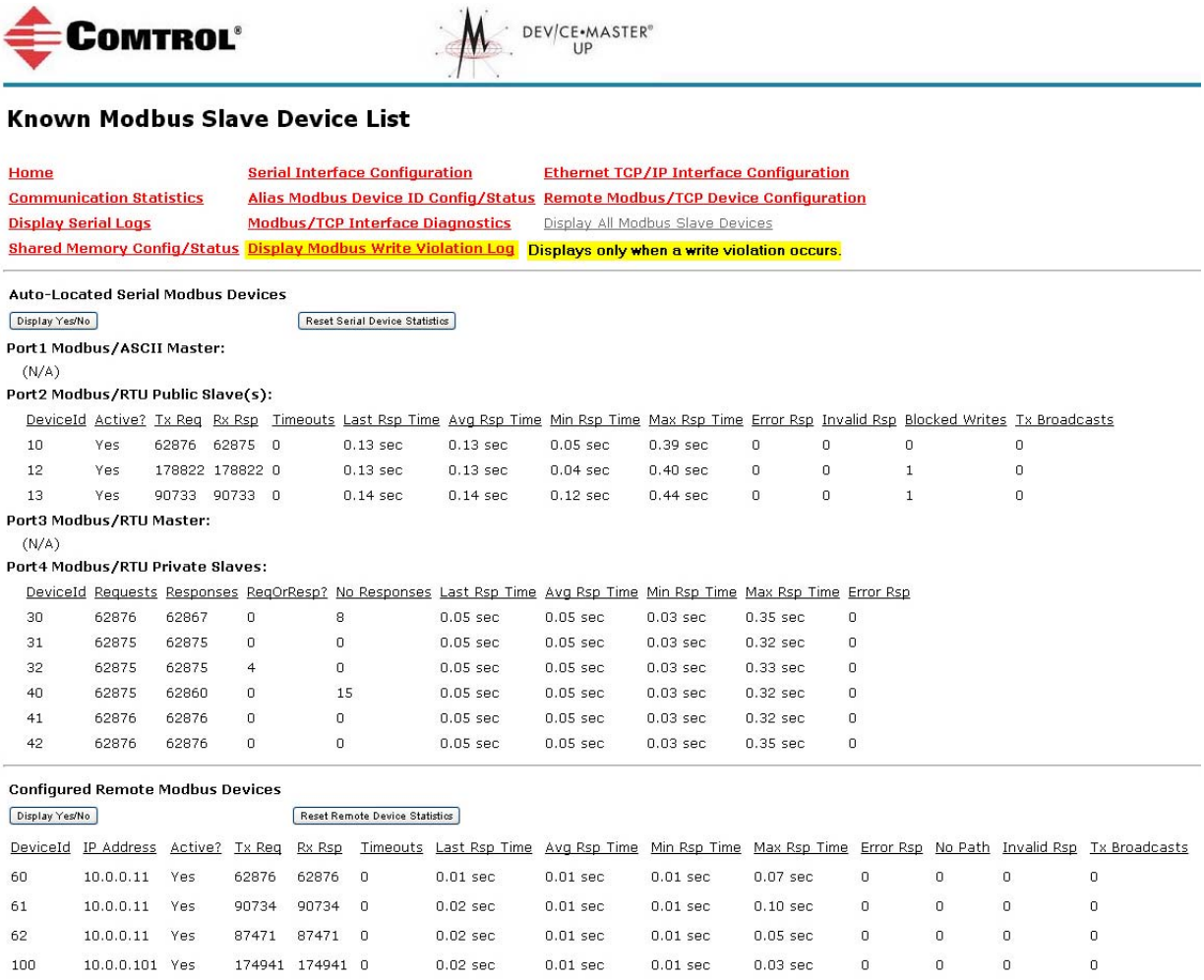


Item	Modbus Write Violation Log Descriptions
Entry	The log entry number.
Time Since startup	The time since the DeviceMaster UP was started up due to a power on/cycle or a system reset. The format displays in the **ddd hh:mm:ss:mss** format where: • ddd = days • hh = hours • mm = minutes • ss = seconds • mss = milliseconds
Source	The source or sender, of the Modbus write message: • The IP address in standard xxx.xxx.xxx.xxx format will be displayed for all Modbus/TCP and Modbus over Ethernet TCP/IP masters. • Will be displayed in SP=<port number> for serial Modbus masters.

Item	Modbus Write Violation Log Descriptions (Continued)
Protocol	The protocol of the received Modbus message. Possible values are: • Modbus/TCP • Modbus/RTU (serial or Ethernet) • Modbus/ASCII (serial or Ethernet)
Device ID	Displays the device ID associated with the received write message. **Note:** *If this message underwent a Device ID Alias operation, the originally received device Id will be displayed as (Rx=xxx).*
Function Code	Displays the Modbus write function code.
Address	Display the address of the intended write message in base 1 format. **Note:** *If this message was addressed to shared memory, the shared memory address will be displayed followed by (Shared memory).*
Count	The number of items the write message intended to modify. • For write register messages, this is the number of 16 bit registers. • For write coil messages, this is the number of coils. • For write file record messages, this is the number of records.
Data	The data the write message intended to write to the specified address. • For write register messages, the data will be displayed in 16 bit hex word format. • For all other write messages, the data will be displayed in 8 bit, or byte, hex format.

2.12. Display All Modbus Slave Devices Page

The *Known Modbus Slave Device List* page provides device specific status and statistics for each device connected locally to one or more of the serial ports or remotely through a remote Modbus/TCP device configuration. You can click *Display All Modbus Slave Devices* on any Modbus Router page to access this information. The following table provides detailed information about this page.



COMTROL® DEVICE•MASTER® UP

Known Modbus Slave Device List

Home | Serial Interface Configuration | Ethernet TCP/IP Interface Configuration
Communication Statistics | Alias Modbus Device ID Config/Status | Remote Modbus/TCP Device Configuration
Display Serial Logs | Modbus/TCP Interface Diagnostics | Display All Modbus Slave Devices
Shared Memory Config/Status | Display Modbus Write Violation Log | Displays only when a write violation occurs.

Auto-Located Serial Modbus Devices

Display Yes/No | Reset Serial Device Statistics

Port1 Modbus/ASCII Master:

(N/A)

Port2 Modbus/RTU Public Slave(s):

DeviceId	Active?	Tx Req	Rx Rsp	Timeouts	Last Rsp Time	Avg Rsp Time	Min Rsp Time	Max Rsp Time	Error Rsp	Invalid Rsp	Blocked Writes	Tx Broadcasts
10	Yes	62876	62875	0	0.13 sec	0.13 sec	0.05 sec	0.39 sec	0	0	0	0
12	Yes	178822	178822	0	0.13 sec	0.13 sec	0.04 sec	0.40 sec	0	0	1	0
13	Yes	90733	90733	0	0.14 sec	0.14 sec	0.12 sec	0.44 sec	0	0	1	0

Port3 Modbus/RTU Master:

(N/A)

Port4 Modbus/RTU Private Slaves:

DeviceId	Requests	Responses	ReqOrResp?	No Responses	Last Rsp Time	Avg Rsp Time	Min Rsp Time	Max Rsp Time	Error Rsp
30	62876	62867	0	8	0.05 sec	0.05 sec	0.03 sec	0.35 sec	0
31	62875	62875	0	0	0.05 sec	0.05 sec	0.03 sec	0.32 sec	0
32	62875	62875	4	0	0.05 sec	0.05 sec	0.03 sec	0.33 sec	0
40	62875	62860	0	15	0.05 sec	0.05 sec	0.03 sec	0.32 sec	0
41	62876	62876	0	0	0.05 sec	0.05 sec	0.03 sec	0.32 sec	0
42	62876	62876	0	0	0.05 sec	0.05 sec	0.03 sec	0.35 sec	0

Configured Remote Modbus Devices

Display Yes/No | Reset Remote Device Statistics

DeviceId	IP Address	Active?	Tx Req	Rx Rsp	Timeouts	Last Rsp Time	Avg Rsp Time	Min Rsp Time	Max Rsp Time	Error Rsp	No Path	Invalid Rsp	Tx Broadcasts
60	10.0.0.11	Yes	62876	62876	0	0.01 sec	0.01 sec	0.01 sec	0.07 sec	0	0	0	0
61	10.0.0.11	Yes	90734	90734	0	0.02 sec	0.01 sec	0.01 sec	0.10 sec	0	0	0	0
62	10.0.0.11	Yes	87471	87471	0	0.02 sec	0.01 sec	0.01 sec	0.05 sec	0	0	0	0
100	10.0.0.101	Yes	174941	174941	0	0.02 sec	0.01 sec	0.01 sec	0.03 sec	0	0	0	0

Public Devices

Column Name	Description for Public Devices
Device ID	Displays the device ID associated with this device. **Note:** *If **Device ID Offset Mode** is enabled, the actual device ID transmitted out the serial port displays as (SP=xxx).*
Active?	Displays the status of device: • **Yes** means that the last request received a valid response and did not time out. • **No** means that the last request timed out or the device has not yet received a message.

Column Name	Description for Public Devices (Continued)
IP Address	Displays the IP address associated with the local device under **Configured Remote Modbus Devices**.
Tx Req	Displays the number of Modbus messages transmitted to this device.
Rx Rsp	Displays the number of Modbus responses received from this device.
Timeouts	Displays the number of response timeouts associated with this device.
Last Rsp Time	Displays the last response time from the Modbus device.
Avg Rsp Time	Displays the average response time from the Modbus device.
Min Rsp Time	Displays the minimum response time from the Modbus device.
Max Rsp Time	Displays the maximum response time from the Modbus device.
Error Rsp	Displays the number of responses with Modbus error indications.
No Path	This displays under **Configured Remote Modbus Devices** the number of times the network path could not be connected. This could be a result of: • Out of Modbus/TCP connections • Modbus/TCP device not responding • Incorrect IP address
Invalid Responses	Displays the number of invalid messages or responses received to any of the following: • Message received after the timeout period. This may require increasing the **Device Response Timeout**. • Incorrect device ID in response message. • Incorrect function code in response message.
Blocked Writes	Displays the number of Modbus write messages that were not transmitted for this device. This only occurs when the **Disable Writes (Read Only)** serial port option is selected.
Tx Broadcasts	Displays the number of Modbus broadcast messages transmitted to this device.

Private Devices

Column Name	Description for Private Devices
Device ID	Displays the device ID associated with this device. **Note:** *If Device ID Offset Mode is enabled, the actual device ID transmitted out the serial port displays as (SP=xxx).*
Requests	Displays the number of private requests addressed to this Modbus device.
Responses	Displays the number of private responses from this Modbus device.
ReqOrResp?	Displays the number of private requests/responses addressed to/from this Modbus device that could not be identified specifically as either a request or a response.

Column Name	Description for Private Devices (Continued)
No Responses	Displays the number of requests that that this Modbus device did not respond to.
Last Rsp Time	Displays the last response time from the Modbus device.
Avg Rsp Time	Displays the average response time from the Modbus device.
Min Rsp Time	Displays the minimum response time from the Modbus device.
Max Rsp Time	Displays the maximum response time from the Modbus device.
Error Rsp	Displays the number of responses with Modbus error indications.

2.13. Serial Interface Logs Page

This page displays the serial messages transmitted and received during normal operation. Click **Display Serial Logs** to access this page.

The format is as follows:

Pkt(N): ddd:hh:mm:ss.mss Rx/Tx>(data packet)

Where:

ddd – days since last system restart

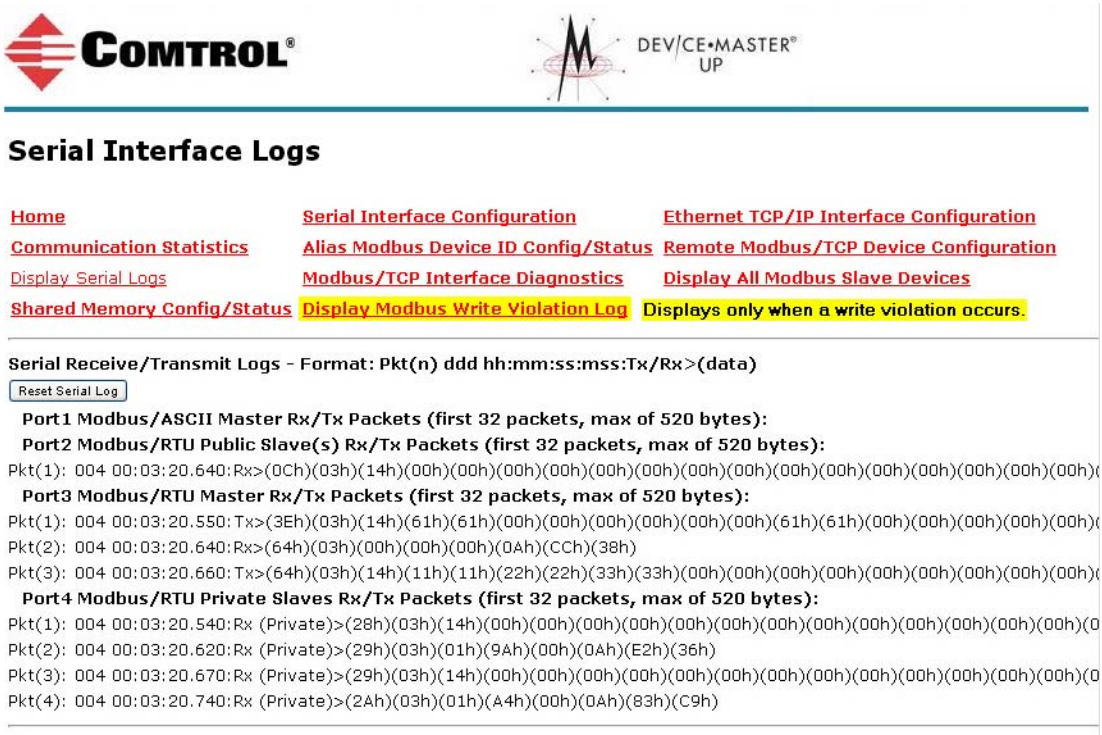
hh – hours since last system restart

mm – minutes since last system restart

ss – seconds since last system restart

mss – milliseconds since last system restart

(**data**) – Modbus/RTU or Modbus/ASCII message data.

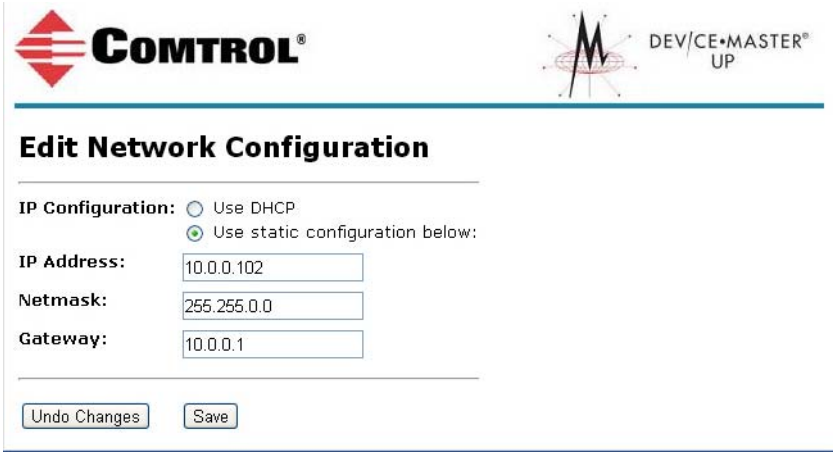


- Private messages between the master and slaves on the Master/Slaves ports are indicated with "(**Private**)".
- All other messages are those routed through the Modbus gateway to the Modbus network or Shared Memory.

2.14. Configure Network Page

You can use the *Edit Network Configuration* page to change the DeviceMaster UP network configuration after using PortVision DX for initial network configuration. Click **Configure Network** to access this page.

Use the following procedure to change the network configuration.



1. Select the IP configuration type (DHCP or Static).
2. If you select **Static**, enter a valid IP address, subnet mask, and IP gateway for your network. The network information is programmed into the DeviceMaster UP after applying the changes and rebooting the device. If necessary, see your network administrator for a valid IP address.

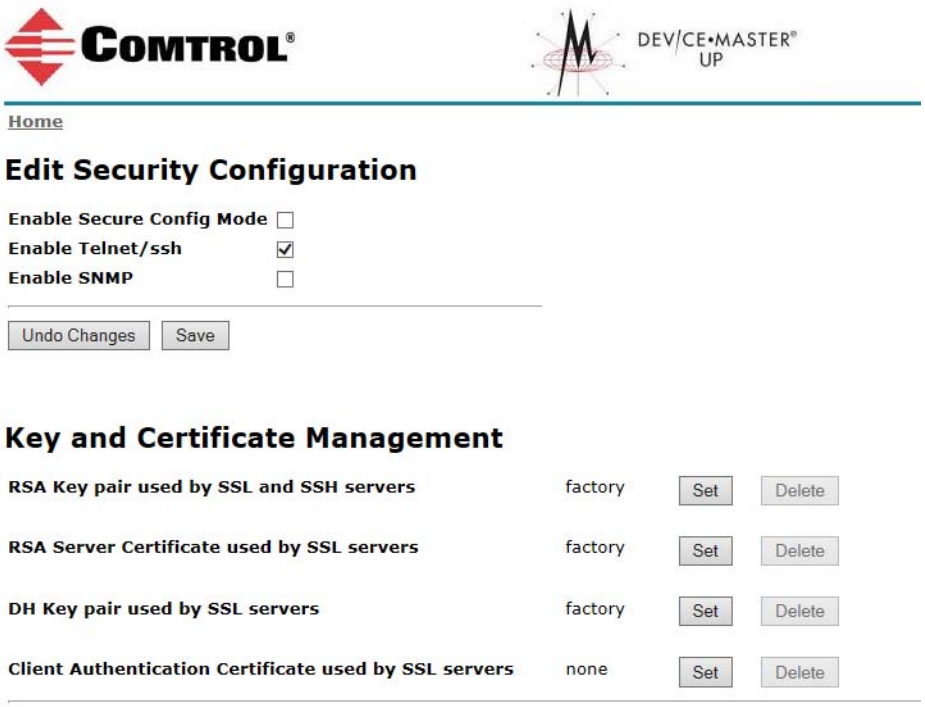
   The default IP address is 192.168.250.250, default subnet mask is 255.255.0.0, and the default IP gateway is 192.168.250.1.
3. Select **Save** or **Undo Changes** to close the page.

   If you selected **Save**, select **Reboot** to program the network information into the DeviceMaster UP or **Continue** if you want to reboot later.

Note: *Changed network settings will not take affect until the DeviceMaster UP is rebooted.*

2.15. Edit Security Configuration Page

To configure security on the DeviceMaster UP, use the *Edit Security Configuration* page.



Option	Description
Enable Secure Data Mode (Default = Disabled)	If **Secure Data Mode** is enabled, TCP connections that carry data to/from the serial ports are encrypted using SSL or TLS security protocols. This includes the following: • TCP connections to the per-serial-port TCP ports (default is 8000, 8001, 8002, ...) are encrypted using SSL/TLS. • TCP connections to TCP port 4606 on which the DeviceMaster UP implements the Comtrol proprietary protocol are encrypted using SSL/TLS. • In addition to encrypting the data streams, it is possible to configure the DeviceMaster UP so that only authorized client applications can connect using SSL/TLS. *2.15.1. Client Authentication* on Page 69 for more information.
Enable Telnet/ssh (Default = Enabled)	This option enables or disables the telnet security feature after you click **Save** and the DeviceMaster UP has been rebooted.
Enable SNMP (Default = Disabled)	This option enables or disables the SNMP security feature after you click **Save** and the DeviceMaster UP has been rebooted.

Option	Description
RSA Key pair used by SSL and SSH servers	This is a private/public key pair that is used for two purposes: • It is used by some cipher suites to encrypt the SSL/TLS handshaking messages. Possession of the private portion of this key pair allows an eavesdropper to both decrypt traffic on SSL/TLS connections that use RSA encryption during handshaking. • It is used to sign the Server RSA Certificate in order to verify that the DeviceMaster UP is authorized to use the server RSA identity certificate. • Possession of the private portion of this key pair allows somebody to pose as the DeviceMaster UP. If the Server RSA Key is to be replaced, a corresponding RSA identity certificate must also be generated and uploaded or clients are not able to verify the identity certificate.
RSA Server Certificate used by SSL servers	This is the RSA identity certificate that the DeviceMaster UP uses during SSL/TLS handshaking to identify itself. It is used most frequently by SSL server code in the DeviceMaster UP when clients open connections to the DeviceMaster's secure web server or other secure TCP ports. If a DeviceMaster UP serial port configuration is set up to open (as a client) a TCP connection to another server device, the DeviceMaster UP also uses this certificate to identify itself as an SSL client if requested by the server. In order to function properly, this certificate must be signed using the Server RSA Key. This means that the server RSA certificate and server RSA key must be replaced as a pair.
DH Key pair used by SSL servers	This is a private/public key pair that is used by some cipher suites to encrypt the SSL/TLS handshaking messages. Possession of the private portion of the key pair allows an eavesdropper to decrypt traffic on SSL/TLS connections that use DH encryption during handshaking. Client Authentication Certificate used by SSL serversIf configured with a CA certificate, the DeviceMaster UP requires all SSL/TLS clients to present an RSA identity certificate that has been signed by the configured CA certificate. As shipped, the DeviceMaster UP is not configured with a CA certificate and all SSL/TLS clients are allowed. See *2.15.1. Client Authentication* on Page 69 for more detailed information.
Client Authentication Certificate used by SSL servers	If desired, controlled access to SSL/TLS protected features can be configured by uploading a client authentication certificate to the DeviceMaster UP. By default, the DeviceMaster UP is shipped without a CA (Certificate Authority) and therefore allows connections from any SSL/TLS client.

2.15.1. Client Authentication

If a CA certificate is uploaded, the DeviceMaster UP only allows SSL/TLS connections from client applications that provide to the DeviceMaster UP an identity certificate that has been signed by the CA certificate that was uploaded to the DeviceMaster UP.

This uploaded CA certificate that is used to validate a client's identity is sometimes referred to as a trusted root certificate, a trusted authority certificate, or a trusted CA certificate. This CA certificate might be that of a trusted commercial certificate authority or it may be a privately generated certificate that an organization creates internally to provide a mechanism to control access to resources that are protected by the SSL/TLS protocols.

To control access to the DeviceMaster UP's SSL/TLS protected resources you should create your own custom CA certificate and then configure authorized client applications with identity certificates signed by the custom CA certificate.

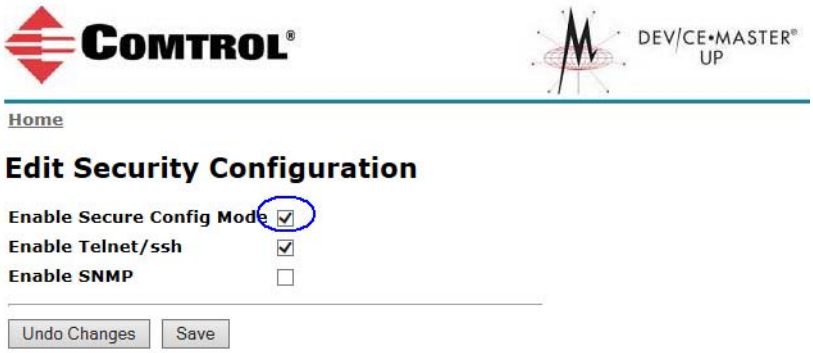
2.15.2. Configuring Security

Use the following procedure to configure DeviceMaster UP security.

Note: *All DeviceMaster UP units are shipped from the factory with identical configurations. They all have the identical, self-signed, Comtrol Server RSA Certificates, Server RSA Keys, Server DH Keys, and no Client Authentication Certificates.*

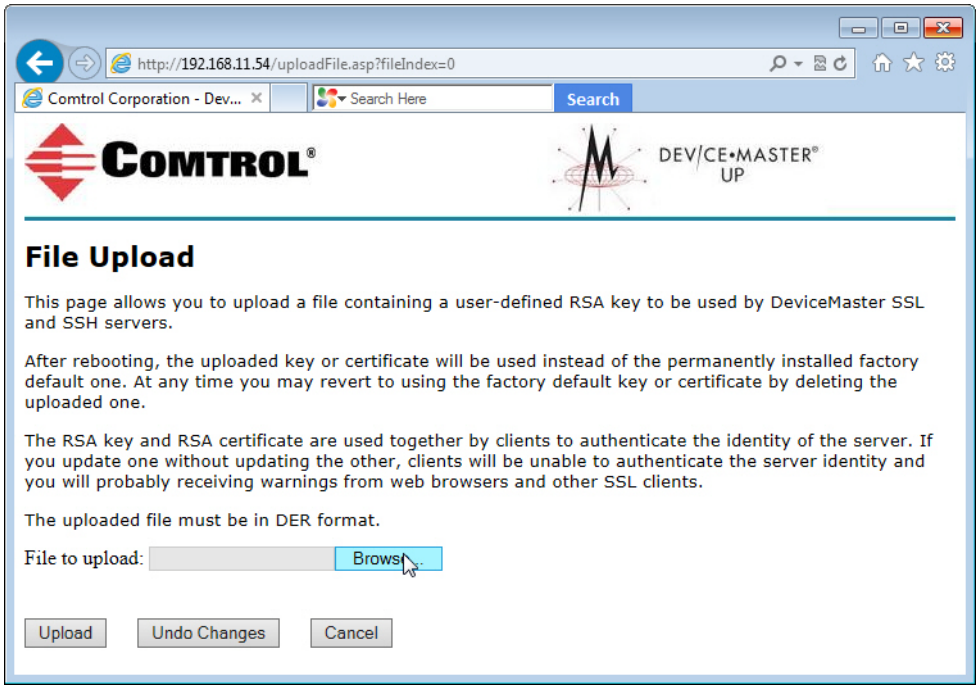
For maximum data and access security, you should configure all DeviceMaster units with custom certificates and keys.

1. If necessary, access the *Router Configuration* web page by entering the DeviceMaster UP IP address in your web browser or by highlighting the DeviceMaster UP in PortVision DX and clicking **Webpage**.
2. To enable secure configuration mode, click **Secure Config Mode**.



3. If you are not going to upload a key or certificate, go to *Step 7.*
4. Click **Set** for the appropriate key or certificate option in the *Keys and Certificate Management* area to configure security keys and certificates.

5. Click **Browse** to locate the key or certificate file, highlight the file, and click **Open**.

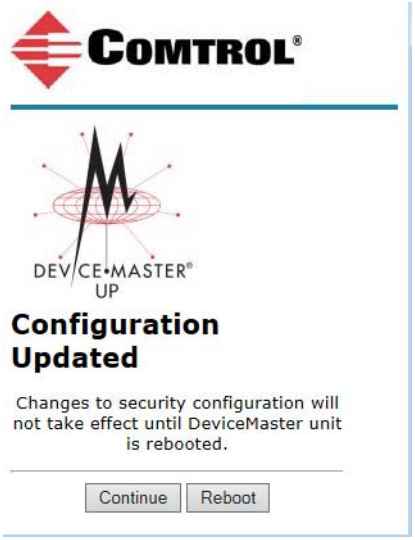


6. Click **Upload** when you return to the *Key and Certificate Management* area.

   The key or certificate notation changes from factory or none to User when the DeviceMaster UP is secure.

7. Click **Save** so that you can access the *Configuration Updated* page, click the **Reboot** button.

   Note: *Changes do not take effect until the DeviceMaster UP is rebooted.*



Section 3. Troubleshooting and Technical Support

This section contains troubleshooting information for your DeviceMaster UP. You should review the following subsections before calling Technical Support because they will request that you perform many of the procedures or verifications before they will be able to help you diagnose a problem.

If you cannot diagnose the problem, you can contact *3.4. Technical Support* on Page 74.

3.1. Troubleshooting Checklist

The following checklist may help you diagnose your problem:

- Verify that you are using the correct types of cables on the correct connectors and that all cables are connected securely.

  Note: *Most customer problems reported to Comtrol Technical Support are eventually traced to cabling or network problems.*

- Isolate the DeviceMaster UP from the network by connecting the device directly to a NIC in a host system.

Model	Connected to	Ethernet Cable	Connector Name
1-Port	Ethernet hub or NIC	Standard	10/100 ETHERNET
1-Port Embedded	Ethernet hub or NIC	Standard	RJ45 port (not labeled)
2-Port - 1E (All models)	Ethernet hub or NIC	Standard	10/100
2-Port - 2E (All dual Ethernet ports)	NIC or Ethernet hub	Standard	10/100 - E1/E2
4-Port	NIC	Standard	DOWN
	Ethernet hub	Standard	UP

- Verify that the Ethernet hub and any other network devices between the system and the DeviceMaster UP are powered up and operating.

- Reset the power on the DeviceMaster UP and watch the **PWR** or **Status** light activity.

PWR or Status LED	Description
5 sec. off, 3 flashes, 5 sec. off, 3 flashes ...	Redboot™ checksum failure.
5 sec. off, 4 flashes, 5 sec. off, 4 flashes ...	SREC load failure.
5 quick flashes	The default application is starting up.
10 sec. on, .1 sec. off, 10 sec. on .1 sec. off ...	The default application is running.

- If the device has a power switch, turn the device's power switch off and on, while watching the LED diagnostics.
- If the DeviceMaster UP does not have a power switch, disconnect and reconnect the power cord.
- Verify that the network IP address, subnet mask, and gateway is correct and appropriate for the network. If IP addressing is being used, the system should be able to ping the DeviceMaster UP.
- Verify that the IP address programmed into the DeviceMaster UP matches the unique reserved IP configured address assigned by the system administrator.
- If using DHCP, the host system needs to provide the subnet mask and gateway.
- Reboot the system and the DeviceMaster UP.
- If you have a spare DeviceMaster UP, try replacing the device.

3.2. General Troubleshooting

This table illustrates some general troubleshooting tips.

Note: *Make sure that you have reviewed the Troubleshooting Checklist (Page 71).*

General Condition	Explanation/Action
PWR or **Status** LED flashing	Indicates that boot program has not downloaded to the unit. 1. Reboot the system. 2. Make sure that you have downloaded the most current firmware for *Modbus Router*. **Note:** *If the PWR or Status LED is still flashing, contact Technical Support.*
PWR or **Status** LED not lit	Indicates that power has not been applied or there is a hardware failure. Contact Technical Support.
Cannot ping the device through Ethernet hub	Isolate the DeviceMaster UP from the network. Connect the device directly to the NIC in the host system (see Page 71).
Cannot ping or connect to the DeviceMaster UP	The default IP address is often not accessible due to the subnet masking from another network unless **192.168** is used in the network. In most cases, it will be necessary to program in an address that conforms to your network.

General Condition	Explanation/Action
DeviceMaster UP continuously reboots when connected to some Ethernet switches or routers	Invalid IP information may also cause the switch or router to check for a gateway address. Lack of a gateway address is a common cause.

3.3. Daisy-Chaining DeviceMaster UP 2E/4-Port Units

The DeviceMaster UP 2E/4-port models with external power supplies follow the IEEE specifications for standard Ethernet topologies.

When using the **UP** and **DOWN** ports, the DeviceMaster UP 2E/4 is classified as a switch. When using the **UP** port only, it is a simple end node device.

The maximum number of daisy-chained DeviceMaster UP 2E/4 units, and the maximum distance between units is based on the Ethernet standards and will be determined by your own environment and the conformity of your network to these standards.

Comtrol has tested with seven DeviceMaster UP 2E/4 units daisy-chained together using 10 foot CAT5 cables, but this is not the theoretical limit. You may experience a performance hit on the devices at the end of the chain, so it is recommended that you overload and test for performance in your environment. The OS and the application may also limit the total number of ports that may be installed.

Following are some quick guidelines and URLs of additional information. Please note that standards and URLs do change.

- Ethernet 10BASE-T Rules
  - The maximum number of repeater hops is four.
  - You can use Category 3 or 5 twisted-pair 10BASE-T cables.
  - The maximum length of each cable is 100m (328ft).

    Note: *Category 3 or 5 twisted pair cables look the same as telephone cables but they are not the same. The network will not work if telephone cables are used to connect the equipment.*
- Fast Ethernet 100BASE-TX rules
  - The maximum number of repeater hops is two (for a Class II hub). A Class II hub can be connected directly to one other Class II Fast Ethernet hub. A Class I hub cannot be connected directly to another Fast Ethernet hub.
  - You must use Category 5 twisted-pair 100BASE-TX cables.
  - The maximum length of each twisted-pair cable is 100m (328ft).
  - The total length of twisted-pair cabling (across directly connected hubs) must not exceed 205m (672ft).

    Note: *Category 5 twisted pair cables look the same as telephone cables but they are not the same. The network will not work if telephone cables are used to connect the equipment.*
- IEEE 802.3 specification: A network using repeaters between communicating stations (PCs) is subject to the “5-4-3” rule of repeater placement on the network:
  - Five segments connected on the network.
  - Four repeaters.
  - Three segments of the 5 segments can have stations connected. The other two segments must be inter-repeater link segments with no stations connected.

    See *http://www.optronics.gr/Tutorials/ethernet.htm* for more specific information.

    Additional information may be found at *http://compnetworking.about.com/cs/ethernet1/* or by searching the web.

3.4. Technical Support

It contains troubleshooting procedures that you should perform before contacting Technical Support since they will request that you perform, some or all of the procedures before they will be able to help you diagnose your problem. If you need technical support, use one of the following methods.

Comtrol Contact Information	
Downloads	*ftp://ftp.comtrol.com/html/up_main.htm*
Web site	*http://www.comtrol.com*
Phone	763.957.6000