Serial MBX Driver Help

Serial MBX Driver for
Modbus Networks

Version 9

SERIAL MBX DRIVER HELP

Serial MBX® Driver for Modbus Networks

Version 9

Document last revision date October 20, 2017

TABLE OF CONTENTS

INTRODUCTION

The Serial MBX Driver provides connectivity through standard serial COM ports to Modbus-compatible programmable controllers and other devices. It supports both master and slave node communications, in both ASCII and RTU formats. Because the Serial MBX devices emulate Modbus Plus devices, standard Modbus Plus compatible applications can run without modification.

This driver is part of the MBX Driver Suite, MBX OPC Server Suite, MBX OPC Premier Suite, MBX OPC Enterprise Suite and MBX Bridge Suite, providing serial Modbus connectivity for these products.

Remote Connectivity

The Serial MBX Driver includes the MBX Gateway Server. When enabled, the MBX Gateway Server allows other computers on your network access to the Serial MBX devices on your system. The remote system, which can be any Windows node running the MBX Gateway Driver, will then have full Serial MBX Driver functionality. For more information on this capability, refer to the MBX Gateway Driver section.

The Serial MBX Driver can also be used with the Mbx.Net Server. The Mbx.Net Server along with the Mbx.Net Gateway Driver can provide secure remote connectivity to MBX devices on a local network or over the Internet. Just like with the MBX Gateway Server, you will have full Serial MBX Driver functionality on the remote system.

Running 16-Bit Software

The Virtual MBX Driver, which is included with all MBX products, allows all 16-bit NETLIB/NetBIOS-compatible applications, such as Modsoft, to run concurrently with all 32-bit applications in the same computer. For more information, refer to the Virtual MBX Driver section.

Compatibility

The Serial MBX Driver is implemented as part of the Cyberlogic MBX architecture, which is the foundation used in other Cyberlogic drivers such as the MBX Driver, the Ethernet MBX Driver, the MBX Gateway Driver, and the Mbx.Net Gateway Driver. Consequently, these drivers consistently support identical programming interfaces: MBXAPI, MBXAPI.Net and NETLIB. Supporting these existing standards protects the software and R&D investments of end-users and OEMs.

Software developers can use the MBX Software Development Kit (MBX SDK) to obtain connectivity to Modbus, Modbus Plus and Modbus TCP networks for their applications. Applications developed with the MBX SDK can be used with all MBX family drivers and can execute under all current Windows operating systems.

Blending MBX-Supported Networks

The MBX driver family provides support for all Modicon networks through a common architecture, with identical programming interfaces. This means that an application that operates with one of the MBX family drivers, such as the Ethernet MBX Driver, will work with the rest of them as well. Thus, virtually all Modbus Plus compatible software programs can operate over all Modicon-supported networks with no code modifications. You will find a complete description of the MBX family in the Appendix: MBX Architecture and Companion Products.

Migration of existing installations to new hardware products does not require the user to discard working, proven software solutions. As depicted in the diagram below, a user can mix Modbus, Modbus Plus and Modbus TCP based hardware products in existing installations without losing software, network or integration investment.

The MBX family of products includes:

- MBX Driver is Cyberlogic's device driver for Modbus Plus interface adapters.

- Ethernet MBX Driver provides Modbus TCP communication.

- Serial MBX Driver provides Modbus RTU/ASCII communication.

- MBX Gateway Driver works with the other MBX drivers, giving access to Modbus, Modbus Plus and Modbus TCP networks from remote locations.

- Mbx.Net Gateway Driver works with the other MBX drivers, giving secure access to Modbus, Modbus Plus and Modbus TCP networks from remote locations, including over the Internet.

- Virtual MBX Driver works with the other MBX drivers to permit 16-bit legacy software to run in current 32-bit Windows operating systems.

- MBX Bridge allows you to bridge any combination of Modicon networks by routing messages between MBX devices.

- MBX OPC Server connects OPC-compliant client software applications to data sources over all Modicon networks.

- MbpStat.Net is the Modbus Plus network diagnostic utility. It provides the same functionality as the original DOS-based MBPSTAT application.

- Mbx.Net Server a WCF server that provides secure remote access to all MBX devices.

- MBX SDK is a software development kit for MBXAPI, MBXAPI.Net and NETLIB compliant development.

MBX-Enabled System Development

New hardware solutions blend into
Existing installations without software
Or network modifications.
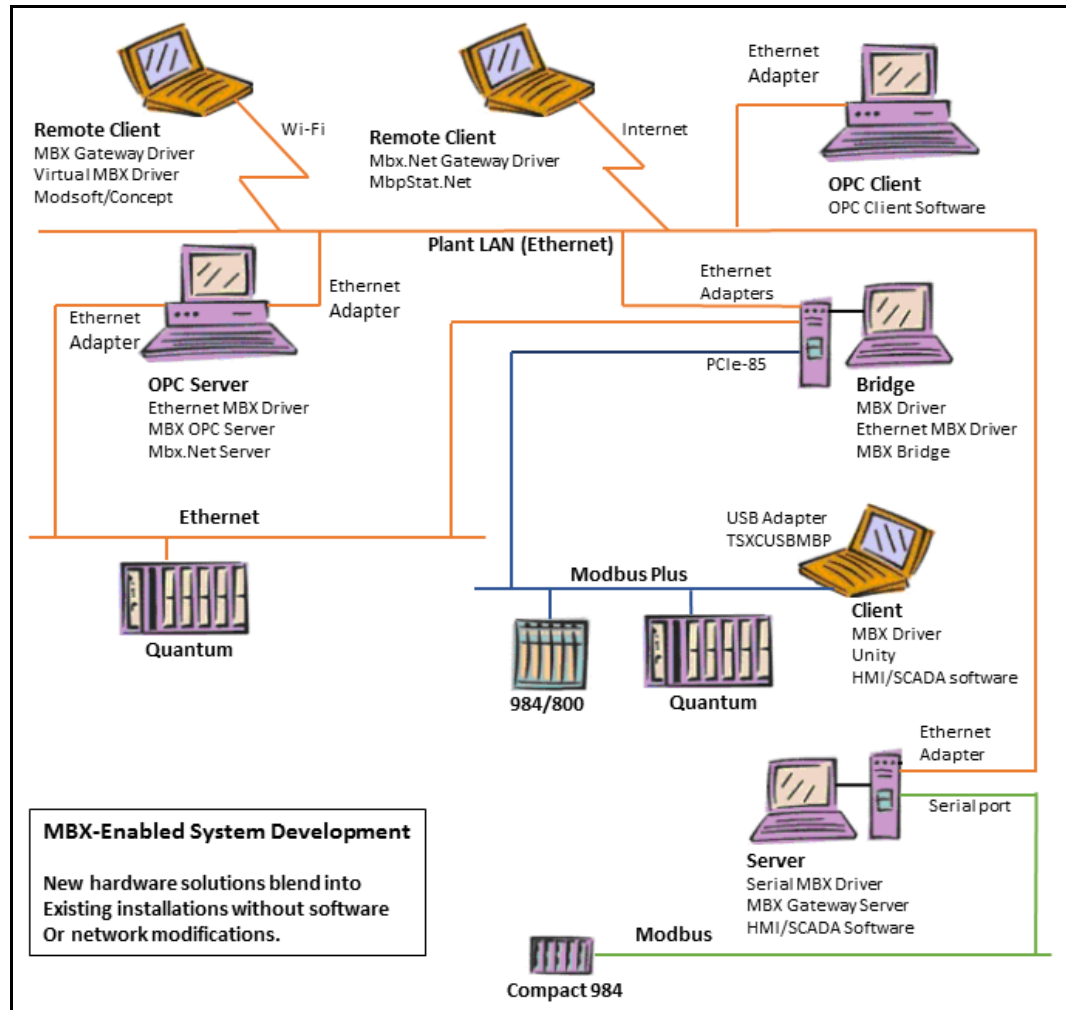
WHAT SHOULD I DO NEXT?

The links below will take you directly to the section of this manual that contains the information you need to configure, use and troubleshoot the Serial MBX Driver.

Learn How the Driver Works

If you are not familiar with the way that the Serial MBX Driver handles communication, including the differences between Serial MBX and standard Modbus, you should begin by reading Communication Using the Serial MBX Driver.

Read a Quick-Start Guide

First-time users of the Serial MBX Driver will want to read the Quick-Start Guide, which walks through a typical configuration session, step-by-step.

Get Detailed Information on the Configuration Editors

Experienced users who want specific information on features of the configuration editors will find it in the Configuration Editor Reference section.

Verify That It's Working or Troubleshoot a Problem

If you have already configured the driver, you should verify that it operates as expected. Refer to the Validation & Troubleshooting section for assistance. In case of communication problems, this section also provides problem-solving hints.

Get Information on Related Products

The MBX family consists of several well-integrated products, which provide connectivity for Modicon networks in distributed environments. For more information about these products, refer to the Appendix: MBX Architecture and Companion Products section.

Print a Copy of This Document

The content of this document is also provided in PDF format. PDF files can be viewed using the Adobe® Reader program, and can also be used to print the entire document.

Contact Technical Support

To obtain support information, open the Windows **Start** menu and go to **Cyberlogic Suites**, and then select **Product Information**.

COMMUNICATION USING THE SERIAL MBX DRIVER

Modbus is a master/slave network that allows a master node (typically a host computer) to communicate to one of several slave nodes (typically Modicon PLCs). The master node supports only a solicited mode of operation, while the slave nodes can only reply to unsolicited message requests from the master node. The message structure supports only 1-byte destination node addressing. Although its serial communications are relatively slow, Modbus is still commonly used in legacy as well as new installations.

The Serial MBX Driver allows existing Modbus Plus software applications, without modification, to communicate over serial Modbus. This is possible because the MBX architecture is modeled on the architecture of Modbus Plus. However, Modbus does not support all of the communication concepts implemented in Modbus Plus. To overcome this, the Serial MBX Driver hides the architectural differences by mapping Modbus communications into appropriate Modbus Plus types of communications. This section describes how you can configure your Serial MBX devices to take advantage of this compatibility.

Serial MBX Device Types

Modbus is a master/slave network that allows a master node to communicate to one of several slave nodes. The master node initiates all communications while the slave nodes can only reply to requests from the master node.

Each COM port associated with a Serial MBX device can operate as either a master node or as one or many slave nodes. Therefore, the Serial MBX Driver implements two types of MBX devices: a Serial MBX Master device and a Serial MBX Slave device.

Serial MBX Master Device

Each Serial MBX Master device has an associated serial COM port and implements the functionality of a Master node. The assigned COM port belongs exclusively to a single device and is not shared with other Serial MBX devices or other applications in the system. Communication requests can be initiated from multiple applications and are processed in first-in-first-out order.

The master node functionality is mapped into the Data/Program Master Path communications of the MBX architecture. Each device allows up to 65,535 Data Master (DM) and Program Master (PM) paths. The following describes the specific behavior of the Data Master and the Program Master path communications.

Data Master Path Communication

Data Master Path communication allows data requests to be sent to destination nodes. Each DM path can be used to send requests to any destination node, but command messages are restricted to data requests only. Programming requests are not allowed over the DM paths.

Program Master Path Communication

Program Master Path communication allows programming requests to be sent to destination nodes. In general, each PM path can send requests to any destination node. However, to protect the integrity of the nodes being programmed, only one application at a time is allowed to send programming requests to a given node. For another application to gain programming rights to that node, all PM paths that already have programming rights to that node must be closed.

For better performance, an application with the programming rights can launch several simultaneous programming requests to the same slave node.

Serial MBX Slave Device

Each Serial MBX Slave device can be configured to respond to multiple slave addresses, thus functioning as a set of virtual Modbus slaves. Unlike the master device, one COM port can be shared by multiple slave Serial MBX devices. The port cannot be shared with any other application in the system, however.

The slave addresses assigned to the Serial MBX Slave device are mapped into the Data/Program Slave Paths of the MBX architecture. Each slave address is mapped to a pair of DS/PS paths sharing the same path number. Note, however, that more than one address can be mapped to the same DS/PS pair. (Refer to the Serial MBX Slave Configuration Editor section for details on how to edit this mapping.) Data requests are routed into a DS path while programming requests are routed into a PS path.

Since the maximum number of slave nodes that a single Serial MBX device can handle is 247, the maximum number of DS/PS paths is also 247. In comparison, Modbus Plus supports a maximum of eight DS/PS paths. As a result, some Modbus Plus applications may not work with more than 8 DS/PS paths. In addition, applications that use the NETLIB library cannot use more than 63 slave paths. If any of your applications impose any of the above restrictions, and you need to assign more slave addresses to the device than the application can handle, you can create multiple Serial MBX Slave devices sharing the same COM port, each with a smaller range of assigned addresses.
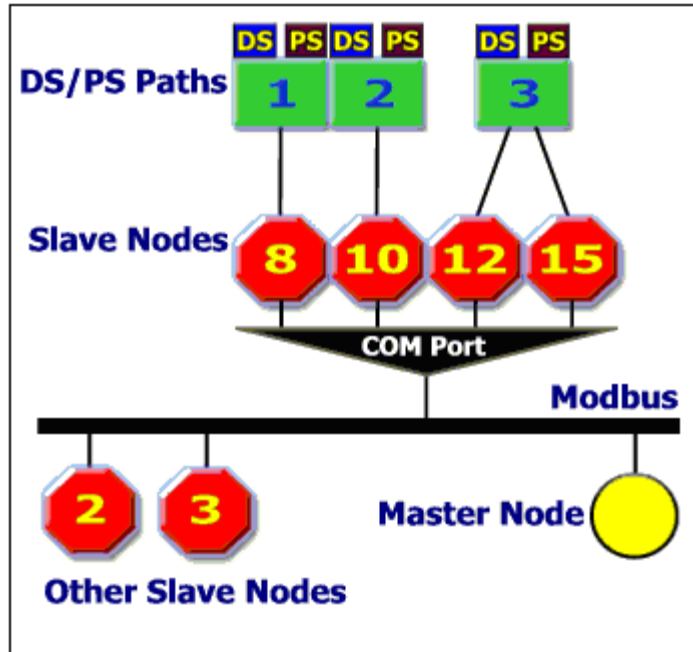
Each Serial MBX Slave device can respond to multiple slave addresses. However, from the MBX architecture perspective, it always appears as a single network node at address 1. In addition, a Modbus network's master node does not have a node address, but for compatibility with the MBX architecture, the master node is always identified as node address 2. As a result, all request messages received from the master always identify the source node address as 2.

In summary, from the Serial MBX Slave device perspective, the entire network consists of only two nodes: its own (address 1) and the master (address 2). There may be other slave nodes on the Modbus network, but they are inaccessible and therefore do not exist as far as the Serial MBX Slave device is concerned.

Broadcast Messages

In addition to regular communication, DM path communication allows broadcast messages to be sent to all slave nodes on the Modbus network. A broadcast message must specify a destination address of either 0 or 255. The Serial MBX Master device always transmits these messages with the destination address of 0. Since no reply messages are expected, the Serial MBX Driver internally generates successful reply messages. This allows applications to handle broadcast messages in the same way as regular DM path messages.

The Serial MBX Driver enforces the use of valid broadcast functions. Only the following Modbus functions support broadcast.

Code	Hex	Function Name
05	05	Force Single Coil
06	06	Preset Single Register
15	0F	Force Multiple Coils
16	10	Preset Multiple Registers

Broadcast messages received by a Serial MBX Slave device are sent to all configured DS paths. To complete slave transactions, client applications must always reply to these messages. However, because the master device sending the broadcast messages does not expect replies, the Serial MBX Driver ignores them, and does not transmit them.

Serial MBX Device Addressing Modes

Modbus addresses consist of a single-byte node address, while Modbus Plus uses a five-byte routing array. In addition, while Modbus addresses may be in the range of 1-247, some Modbus Plus applications restrict their routing byte values to the range of 1-64. To accommodate these differences, the Serial MBX Driver implements two types of addressing: direct addressing and indirect addressing.

Direct Addressing

In direct addressing, the first Modbus Plus routing array byte simply contains the Modbus address (1-247), and the remaining four bytes are set to zero.

Permitted Routing Array Byte Values – Direct Addressing				
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
1 – 247	0	0	0	0

Example

The application specifies a routing array of 97.0.0.0.0. The Serial MBX Driver will route this message to Modbus node 97.

Recommended Use

The direct addressing technique is simpler to use than the indirect addressing method, so it is preferred for applications that can specify routing array bytes over the full Modbus address range of 1 – 247. It is also preferred for all applications if all of the node addresses on your network are no higher than 64.

Some applications permit values only in the range of 0 – 64. These applications must use the indirect addressing approach if you need to communicate with nodes that have addresses higher than 64.

Configuration

No configuration is needed for Serial Master devices. When a Serial Master device sees 0 as the second routing array byte, it simply uses the first byte as the Modbus node address when it sends the message.

For Serial Slave devices, you must select the desired mode in the Routing Path Mode group on the Settings tab. This selection determines how the address for an incoming message is reported to the application. When you select Direct mode, the address will be specified as the routing array MB.0.0.0.0, where MB is the Modbus node address that was actually used.

Indirect Addressing

In indirect addressing mode, the first Modbus Plus routing array byte is in the range of 1-64, and the second byte contains a multiplier in the range of 0-3. The remaining three bytes are set to zero.

To convert to a Modbus node address, the Serial MBX Driver multiplies the second byte value by 64 and adds the first byte value.

Permitted Routing Array Byte Values – Indirect Addressing				
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
1 – 64	0 – 3	0	0	0

Example

The application specifies a routing array of 33.1.0.0.0. The driver calculates the required Modbus node address as $(1 * 64) + 33 = 97$.

Recommended Use

This addressing mode is provided for compatibility with some legacy Modbus Plus software. Those applications that restrict the value of routing array bytes to the 0-64 range may not permit you to specify directly all of the node addresses you require. In such cases, you must use indirect addressing.

If, on the other hand, your application allows you to directly specify all of the node addresses you will need, then direct addressing is preferred.

Configuration

No configuration is needed for Serial Master devices. When a Serial Master device sees a value other than 0 as the second routing array byte, it multiplies that value by 64 and adds that to the first byte. The resulting value is used as the Modbus node address when it sends the message.

For Serial Slave devices, you must select the desired mode in the Routing Path Mode group on the Settings tab. This selection determines how the address for an incoming

message is reported to the application. When you select Indirect mode, the address will be specified according to the following table. In the table, MB is the Modbus node address that was actually used to route the message.

MB	Reported Routing Array
1 – 64	MB.0.0.0.0
65 – 128	MB-64.1.0.0.0
129 – 192	MB-128.2.0.0.0
193 – 247	MB-192.3.0.0.0

Active Node Selection and Mapping

The MBX family drivers support all of the Modbus Plus interface functionality, including active node table calls. However, Modbus does not support an active node table. Therefore, by default, the active node table returned for a Serial MBX Master device shows all configured nodes as active. This means that you must manually deactivate all unused nodes, which you can do from the Active Nodes tab of the Serial MBX Master Configuration editor. A Serial MBX Slave device shows only two active nodes: node address 1 for itself and node address 2 for the master node, so it is not necessary to deactivate any nodes.

Serial MBX Master devices can be configured to map the node address specified by the application into a different physical Modbus node. You do this by entering the desired mapping on the Active Nodes tab of the Serial MBX Master Configuration editor.
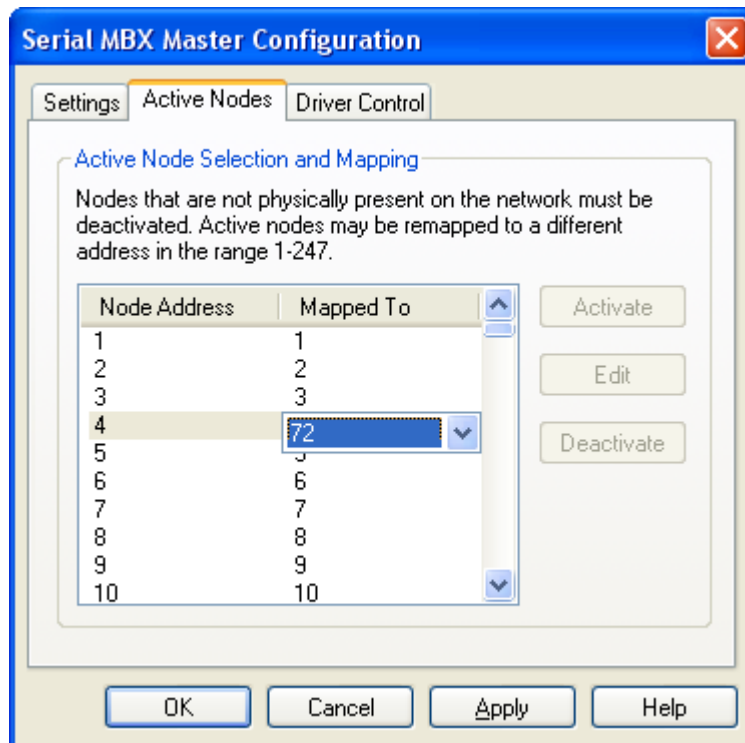
The default mapping has every node address mapped to the same value as itself. To remap a node address, double-click on it and a drop-down box will open, allowing you to select a different value for the mapping.

Example

In the mapping shown above, node address 4 has been mapped to 72. Any message that the application sends to node 4 will now be sent instead to node 72.

Recommended Use

There are two suggested uses for this capability. The first would be a situation in which you have no more than 64 nodes to communicate with, but some have addresses higher than 64 and your software application does not support addresses higher than 64. You could use indirect addressing as explained in the Serial MBX Device Addressing Modes discussion, or you could map some of the unused nodes 1 – 64 to the nodes above 64. This would allow the application to use the simpler direct addressing method, while still keeping the node addresses within the acceptable range.

A second possible use would arise when you have two installations that are nearly identical, differing only in the Modbus node addresses that are used. By remapping the nodes on one installation, you can keep the software identical between the two—with the exception of the node mapping table itself.

The disadvantage of node mapping is that, because the address used by the application is not the same as the actual address for the node, it is much more likely that a user will accidentally reference the wrong address. If you use node mapping, you must take precautions to avoid this.

Hardware Flow Control

The Serial MBX Driver supports the use of hardware flow control or "handshaking". The two output signals, Data Terminal Ready (DTR) and Request To Send (RTS), are always used as follows:

- DTR is always on.
- RTS is always on for master devices. For slave devices, RTS is turned on when the device has data it wishes to send, and is turned off when all of the data has been sent. Thus, slave devices can use RTS for tri-state control of devices such as RS-485 converters.

In addition, when hardware flow control is enabled on a device, the driver uses the two input signals, Data Set Ready (DSR) and Clear To Send (CTS) as follows:

- DSR must be on to permit the device to send or receive data. If DSR is off, no data will be sent and any received data will be ignored.
- CTS must be on to permit the device to send data.

In general, hardware flow control is not needed when you are simply connecting two devices directly using an RS-232 cable. Handshaking should be used if you are using

modems to connect several devices in a multi-drop configuration. By default, hardware flow control is enabled.

Caution!	Not all serial ports support handshaking. When using hardware flow control, always verify that the communication module you are connecting to supports hardware flow control.

Cables

The specific cable requirements for your installation will depend on the type of programmable controller or other equipment you are connecting to. The figures below show some of the more common Modbus cables.
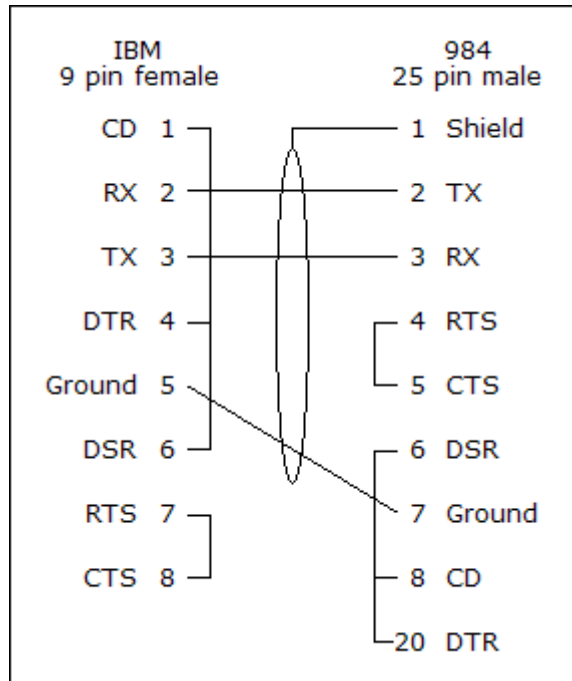
Notice that, for cables used in one-to-one direct connections, some or all of the handshaking pins are jumpered back, whereas they are connected through for cables that are used with modems. For a discussion of the reasons for this, refer to the preceding section, Hardware Flow Control.
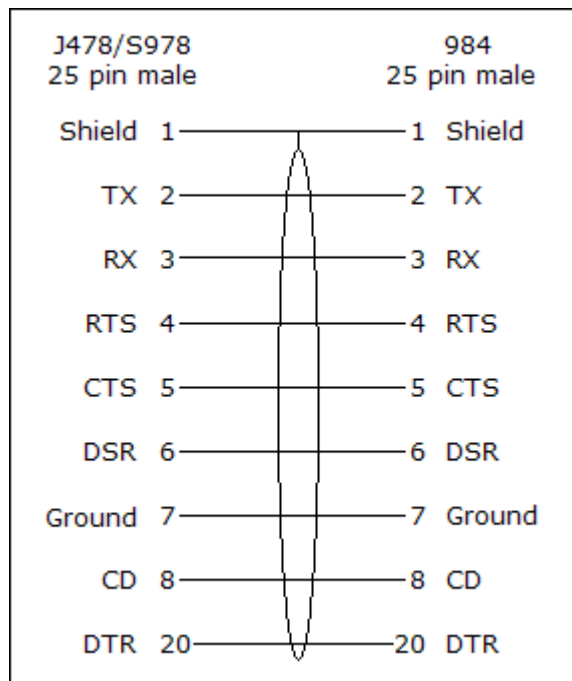
Direct PC to DB-9 Modbus connection
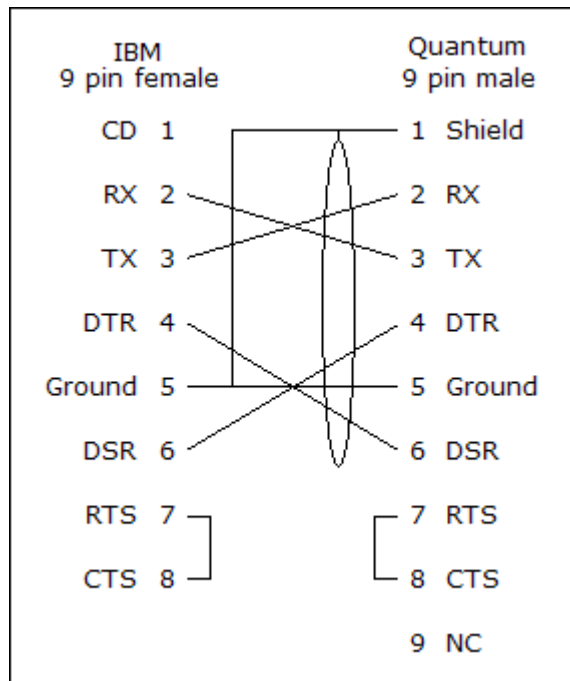
Above is a typical cable for use in connecting a PC's serial port directly to a 9-pin Modbus port as found on Quantum and other controllers.

Direct PC to DB-25 Modbus Connection

```
        IBM                        984
     9 pin female             25 pin male

      CD  1 ────┐      ┌──────  1  Shield

      RX  2 ───┐│      │  ┌────  2  TX

      TX  3 ──┐││      │  │ ───  3  RX

     DTR  4 ─┐│││      │  │  ┌─  4  RTS

  Ground  5  ││││      │  │  └─  5  CTS

     DSR  6  ││││      │  │  ┌─  6  DSR

     RTS  7 ─┐ ││      │  │  │   7  Ground

     CTS  8 ─┘ ││      │  │  └─  8  CD

                              └ 20  DTR
```

This is a typical cable that would be used to connect a PC serial port to a 25-pin Modbus connector, such as would be found on a 984 or 584 controller.

Modem to DB-25 Modbus Connection for Multi-Drop Applications

```
     J478/S978                    984
    25 pin male              25 pin male

  Shield  1 ───────────────── 1  Shield

      TX  2 ───────────────── 2  TX

      RX  3 ───────────────── 3  RX

     RTS  4 ───────────────── 4  RTS

     CTS  5 ───────────────── 5  CTS

     DSR  6 ───────────────── 6  DSR

  Ground  7 ───────────────── 7  Ground

      CD  8 ───────────────── 8  CD

     DTR 20 ───────────────── 20 DTR
```

This is a cable of the type used to connect a 984 or similar controller's 25-pin Modbus port to a Modicon J478/S978 modem. A similar cable would be used for other types of modems that support the full RS-232 handshaking capability.

Modem to DB-9 Modbus Connection for Multi-Drop Applications
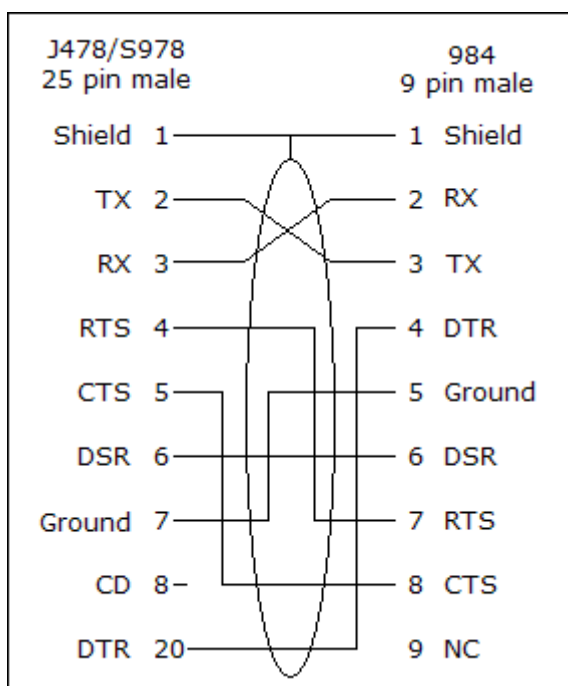
This cable connects a 9-pin Modbus port to a Modicon J478/S978 or similar modem.

Message Routing

Modbus Plus supports the use of message routing, allowing messages to be sent through bridges spanning multiple Modbus Plus networks. Modbus does not support this capability. However, Cyberlogic's **Error! Reference source not found.** allows you to route messages across all MBX-compatible networks.

Global Data

Modbus does not support Global Data communication. Therefore, each Serial MBX device ignores Global Data writes and returns no data for Global Data reads. Both operations complete successfully.

Peer Cop

Modbus does not support Peer Cop communication. As a result, Serial MBX devices do not provide Peer Cop functionality. All Peer Cop related requests return the Not Implemented completion status.

QUICK-START GUIDE

Before the Serial MBX Driver can be used, it must be properly configured. The configuration procedure involves creating one or more Serial MBX devices and configuring them to work on your network. A Serial MBX device is a logical device that emulates the behavior of a Modbus Plus network adapter card. Your software applications will then use this logical device to communicate over the network.

To accomplish this, you must run the MBX Driver Configuration Editor after you install the software. The MBX Driver Configuration Editor is a common component of all drivers in the MBX family. When configuring a Serial MBX device, the MBX Driver Configuration Editor automatically dispatches the Serial MBX Configuration Editor.

The following steps show a typical configuration session. Use it only as a guideline of how to configure the most common features. For detailed descriptions of all of the available features, refer to the Configuration Editor Reference section.

The procedure is broken into several short segments:

- Creating a Serial MBX Device

- Configuring a Serial MBX Master Device

- Configuring a Serial MBX Slave Device

- Configuring the Serial MBX Driver Control

- Configuring the MBX Gateway Server

- Verifying Your Driver Configuration

- Backing Up Your Configuration

After completing this procedure, you will have a fully-configured Serial MBX device and will be able to confirm that the driver is running and communicating with other nodes on your network.

To begin, go to Creating a Serial MBX Device.
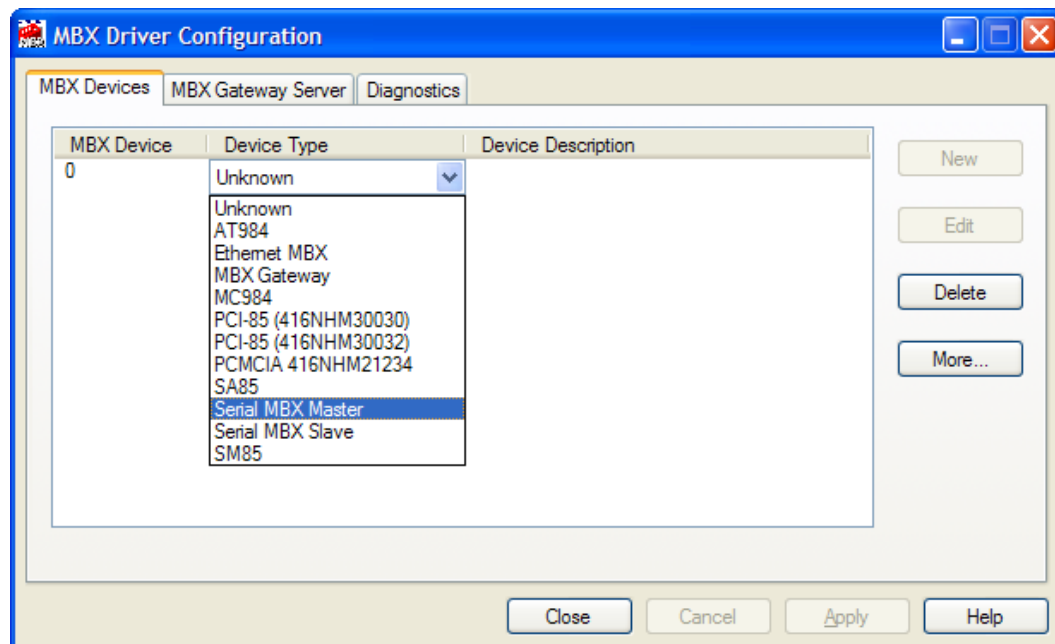
Creating a Serial MBX Device

The first step in configuring the Serial MBX Driver is to create a Serial MBX device. This is a logical device that emulates a Modbus Plus interface adapter, such as a PCIe-85 card. It is not related to an actual, physical network card.

There are two types of Serial MBX devices: Master and Slave. Before you proceed with this section, you must decide which type of device you want to create. Refer to the Communication Using the Serial MBX Driver section for more information.

1. From the Windows **Start** menu, go to **Cyberlogic Suites**, then open the **Configuration** sub-menu, and then select **MBX Device Drivers**.
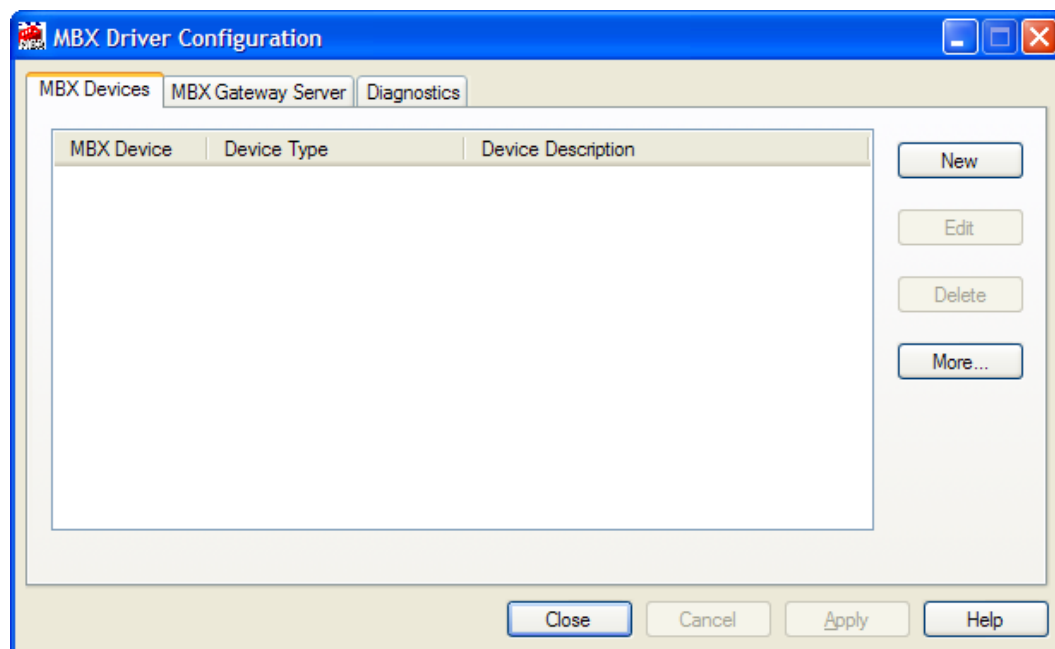
Running the editor for the first time displays the above screen.

2. Click the **New** button and select **Serial MBX Master** or **Serial MBX Slave** from the drop-down list.

This creates a Serial MBX device and launches the Serial MBX Master or Serial MBX Slave Configuration Editor.

If you selected Serial MBX Master, continue with Configuring a Serial MBX Master Device. Otherwise, go to Configuring a Serial MBX Slave Device to continue.

Configuring a Serial MBX Master Device

The Serial MBX Master Configuration Editor consists of three tabs. In this section, we will configure the parameters on the Settings and Active Nodes tabs.

Note	This section is only for configurations that use a Serial MBX Master device. For slave device configuration, go to the Configuring a Serial MBX Slave Device section.
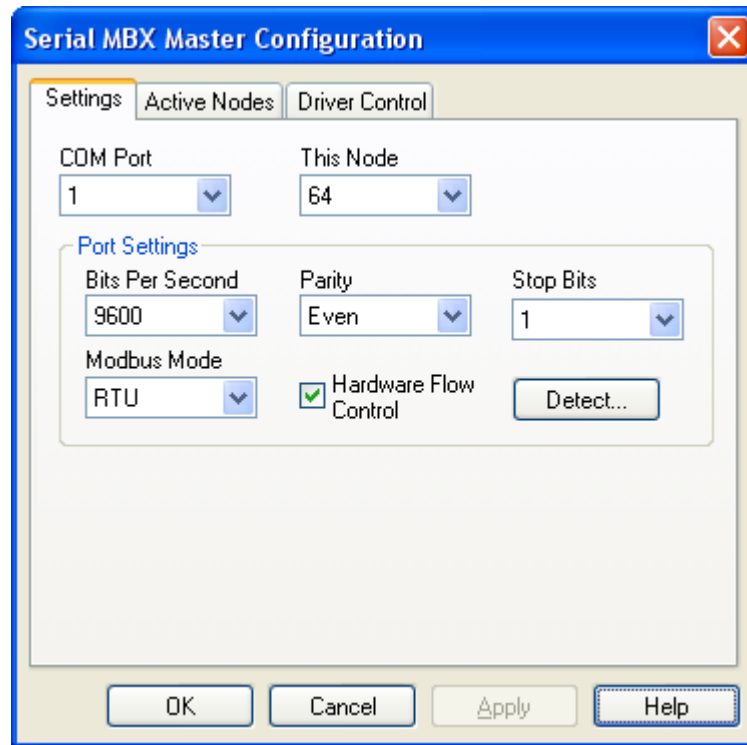
1. Select the serial **COM Port** associated with your Serial MBX device.

   Each Serial MBX Master device has an associated serial COM port. The assigned COM port belongs exclusively to a single master device and is not shared with other Serial MBX devices or other applications in the system.

2. Assign a Modbus Plus equivalent node address to this device by selecting a value for **This Node**.

3. If you know the correct Port Settings parameters, select the desired values for **Bits Per Second**, **Parity**, **Stop Bits**, **Hardware Flow Control** and **Modbus Mode**.

   If you do not know the correct Port Settings, connect the serial cable to the network and click the **Detect...** button. You will be asked to narrow down the range of possible settings and node addresses, if you can, and the remaining combinations will be tried until a working set of values is found.

| Note | If you do not narrow down the possibilities at all, every possible combination of port settings will be tried on all 247 node addresses. This could require over 38,000 tests and may take hours to complete. |
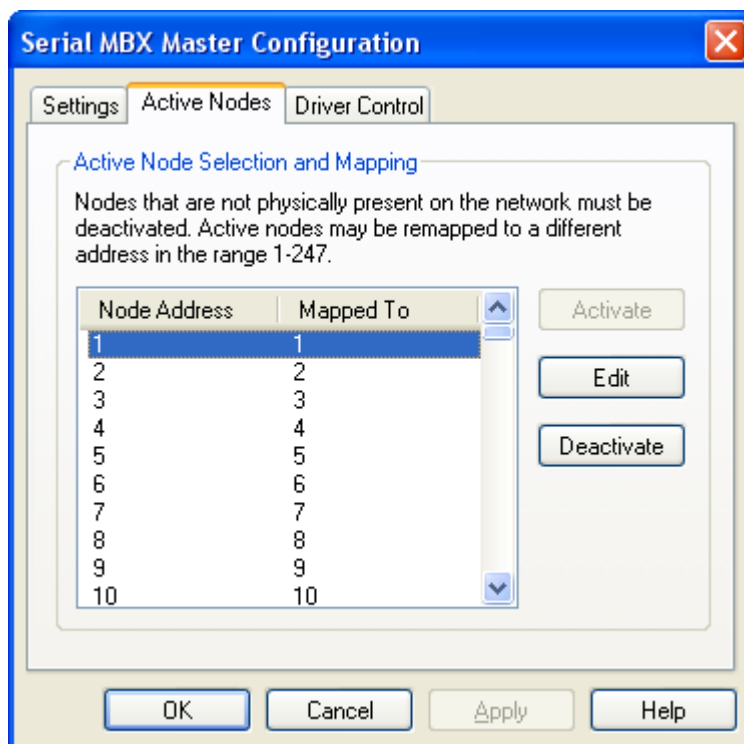
4. Click on the **Active Nodes** tab.

| Caution! | Some applications will not function properly if there are active nodes that are not physically present. If you are sure that your applications will not have trouble with missing active nodes, you may skip this part of the procedure. |

5. Select the **Node Addresses** that are not present on the network and click the **Deactivate** button.

    You may use the Shift-Click and Ctrl-Click functions to select multiple nodes.
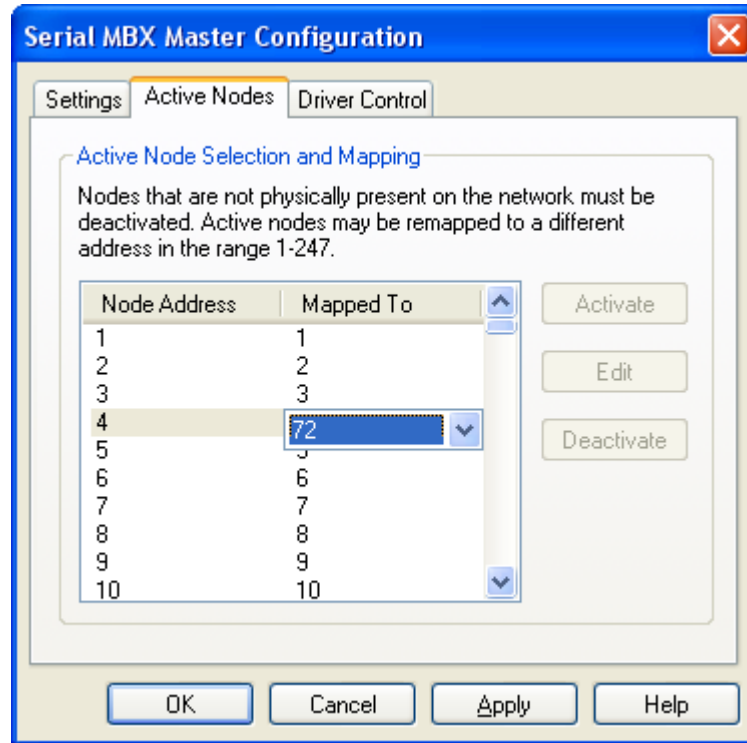
6. Select a node and click the **Edit** button. A list box opens allowing you to map the node to a different address, if you wish.
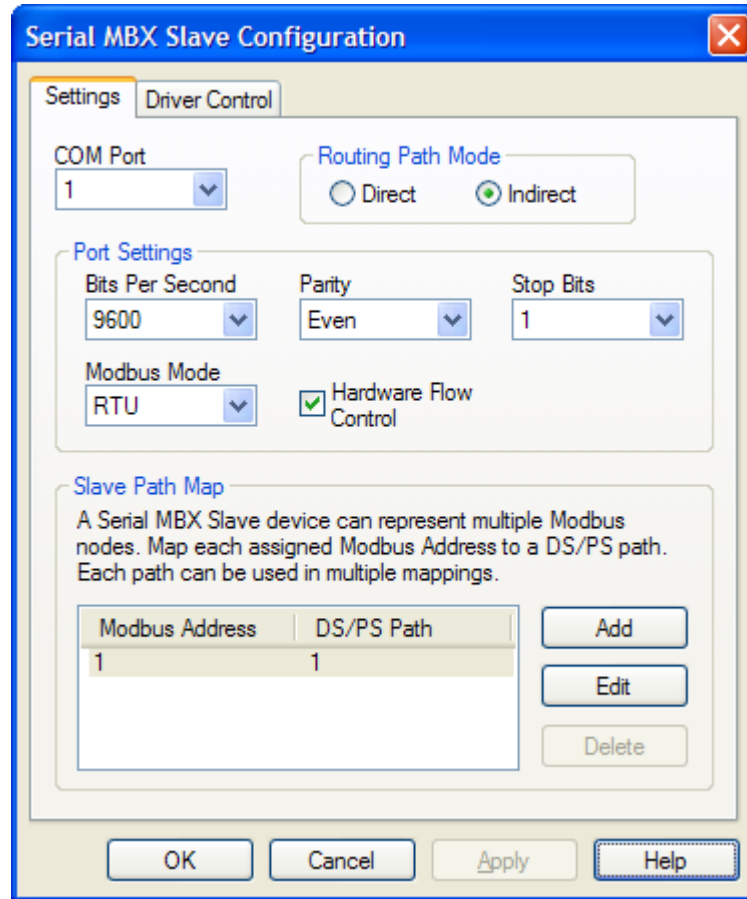
   Refer to Communication Using the Serial MBX Driver for more information on node mapping.

   Now go to Configuring the Serial MBX Driver Control to continue.

Configuring a Serial MBX Slave Device

The Serial MBX Slave Configuration Editor consists of two tabs. In this section, we will configure the parameters on the Settings tab.

Note	This section is only for configurations that use a Serial MBX Slave device. For master device configuration, go to the Configuring a Serial MBX Master Device section.

1. Select the serial **COM Port** associated with your Serial MBX device.

   Each Serial MBX Slave device has an associated serial COM port. One COM port can be shared by multiple Serial MBX Slave devices.

2. Select the desired **Bits Per Second** value.

3. Select the desired **Parity** type.

4. Select the desired number of **Stop Bits**.

5. If your hardware uses hardware flow control, check the **Hardware Flow Control** box. Otherwise, clear the box.

6. Select either **RTU** or **ASCII** for the Modbus Mode.

7. In the Routing Path Mode box, select **Direct** or **Indirect**.

   Refer to Serial MBX Device Addressing Modes for a discussion of when to use each mode.

> **Note** Each Serial MBX Slave device can answer to multiple slave addresses, thus functioning as a set of virtual Modbus slaves. Each assigned Modbus node address can be mapped to a pair of DS/PS paths sharing the same path number. Refer to Communication Using the Serial MBX Driver for details.
>
> In the next step, you will configure this mapping.

8. In the Slave Path Map area, select a Modbus Address to map and click **Edit**, then select a **DS/PS Path** from the list box that will open.

   To add more mappings, click the **Add** button, and then select the **Modbus Address** and its corresponding **DS/PS Path** number.

   To delete a Modbus Address entry, select it and click **Delete**.

   Repeat this step until all node addresses are configured.

   Proceed to Configuring the Serial MBX Driver Control to continue.

Configuring the Serial MBX Driver Control

Here, you will determine how the Serial MBX Driver will be started. Most users should select the Automatic startup type. In this mode, the driver starts automatically when the operating system boots.

This configuration is done on the Driver Control tab. This example uses the Serial MBX Master Configuration Editor, but the tab on the Serial MBX Slave Configuration Editor is identical.

> **Caution!** These settings are global, and are common to all Serial MBX Master and Serial MBX Slave devices.
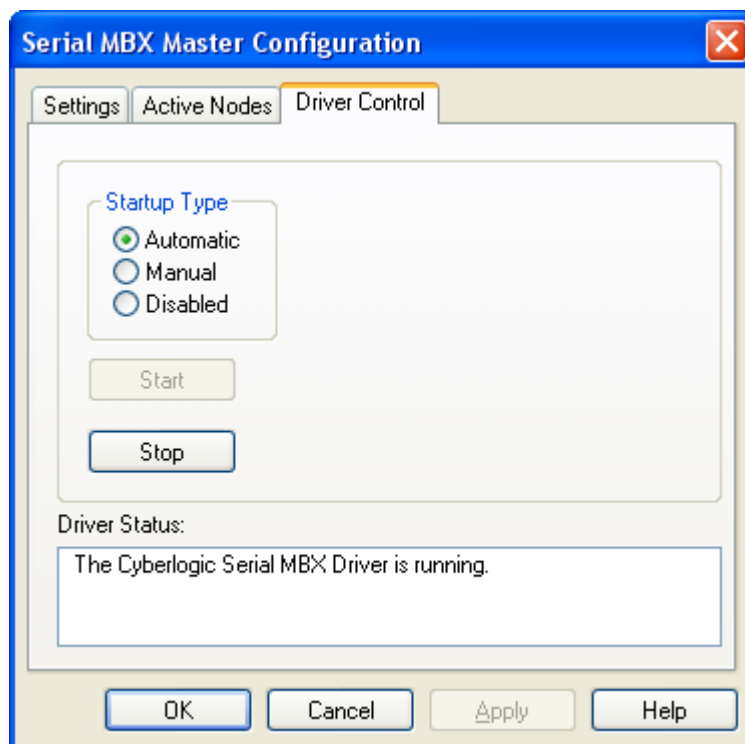
1. Choose the **Driver Control** tab.

2. Select the desired mode of operation among the **Startup Type** choices.

   If you want the driver to start automatically when Windows boots, select **Automatic**. This is the recommended setting.

   If you want to control the driver manually, select **Manual**.

   To prevent the Serial MBX Driver from running, select **Disabled**.

3. Click **Close** to return to the MBX Driver Configuration Editor.

Your Serial MBX device is now fully configured.

Next, go to Configuring the MBX Gateway Server.

Configuring the MBX Gateway Server

The Serial MBX Driver comes with the MBX Gateway Server. The MBX Gateway Server allows remote nodes to access all configured MBX devices present on the system that is running the MBX Gateway Server. Refer to the MBX Gateway Driver section for more information on this capability.

You must enable and configure the MBX Gateway Server if you plan to use the MBX Gateway Driver on other systems on your network and you want them to be able to access the MBX devices on this system. Otherwise, you should disable the MBX Gateway Server.
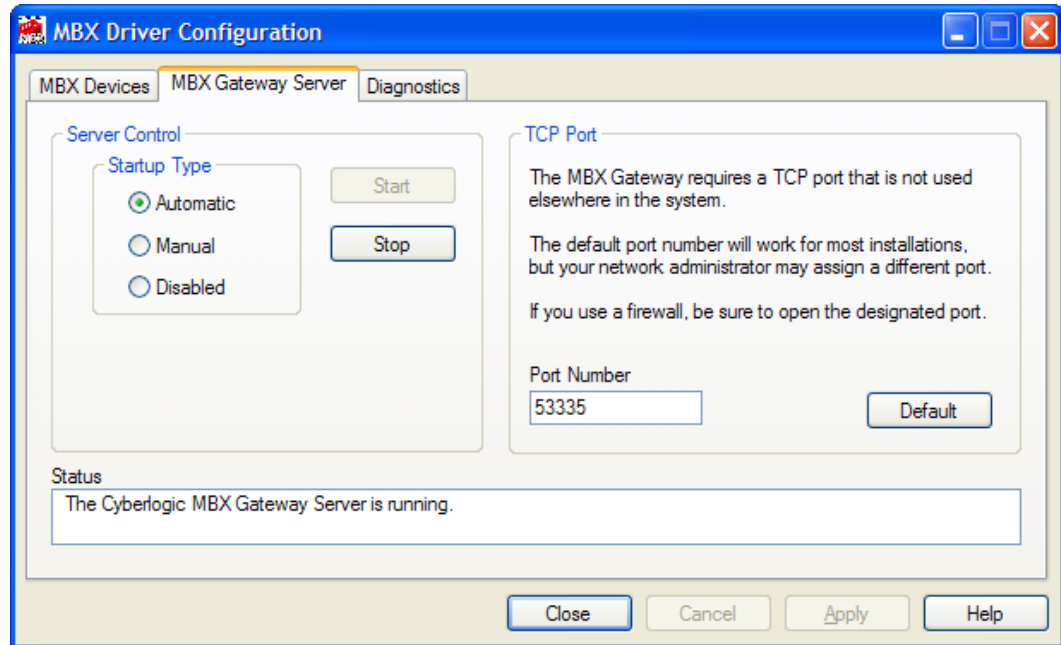
1. Select the **MBX Gateway Server** tab.

2. Select the desired mode of operation among the Startup Type choices.

    If you want to use the MBX Gateway Server and you want it to start whenever the system is booted, select **Automatic**. This is the recommended setting for systems that will use the Gateway Server.

    If you want to use the MBX Gateway Server and want to control it manually, choose **Manual**.

    If you do not want to use the MBX Gateway Server, choose **Disabled**. You can then skip the rest of this section and go directly to Verifying Your Driver Configuration.

3. You must enter a TCP port that is not used elsewhere in the system. The default, 53335, will work for most installations, but this port may be taken in some unusual cases. If that applies to your system, the system administrator will assign a different port value that you must enter in the **Port Number** field.

4. If your system uses a firewall, you must configure it to permit MBX Gateway communication. The procedure will depend upon the firewall you are using. Refer to the MBX Gateway Server Tab discussion in the MBX Driver Configuration Editor section for more information.

Now go to the Verifying Your Driver Configuration section, which will introduce you to the diagnostic features of the product.

Verifying Your Driver Configuration

The Diagnostics tab features will help you to confirm that the driver is running and is properly configured. They will also provide important help in case troubleshooting or technical support is needed.
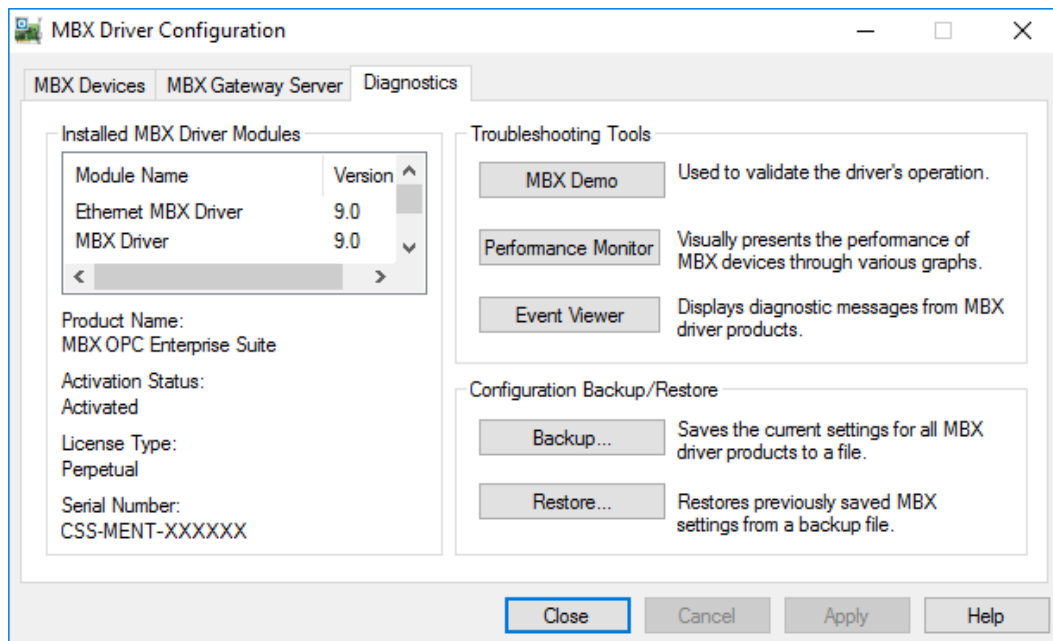
1. Select the **_Diagnostics_** tab.

2. The left pane of this screen shows all MBX software components installed on your system. This information, including the version numbers, may be requested if you call for technical support.

    This screen also tells you if the software has been activated or if it is running in the two-hour demo mode.

Caution!	If you are running in demo mode, the MBX drivers will stop after two hours of operation and will resume after the system is restarted.

3. The right pane of the screen provides shortcuts to troubleshooting and backup/restore tools. Run the **_MBX Demo_** program after configuring the Serial MBX Driver to verify that the driver is configured and running properly. Detailed instructions for running this utility are included in the Validation & Troubleshooting section.
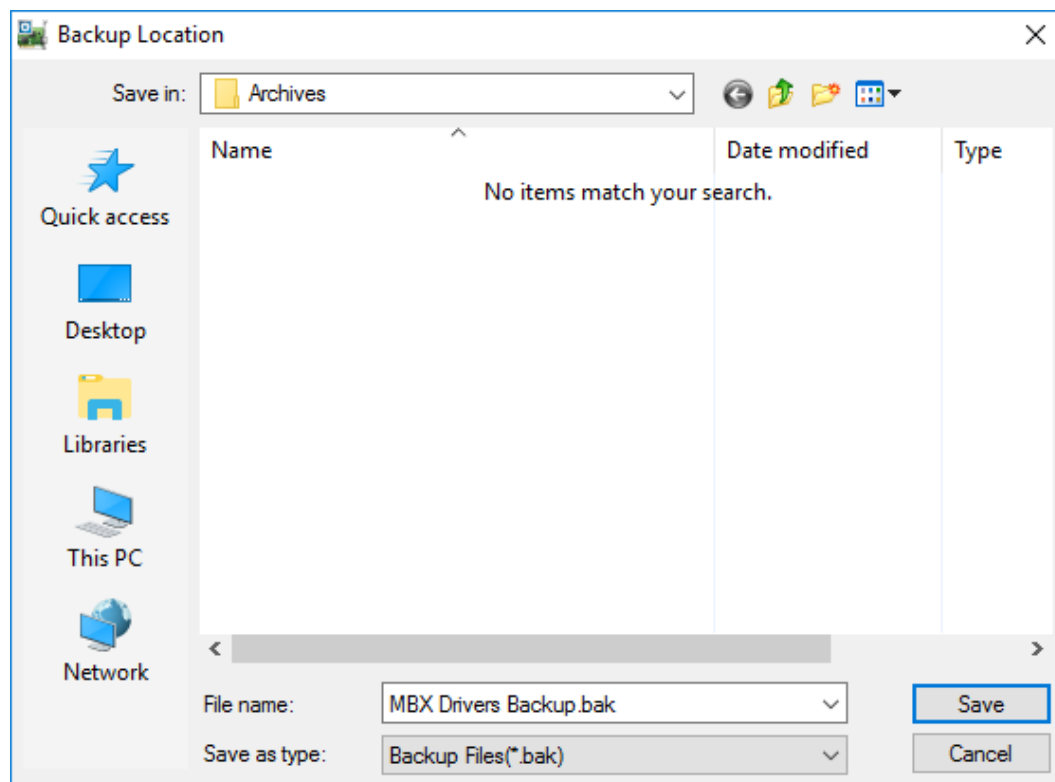
    When you are satisfied that the driver is correctly configured, proceed to Backing Up Your Configuration.

Backing Up Your Configuration

To protect the work that you put into configuring and testing the driver, we strongly recommend that you back up the configuration.
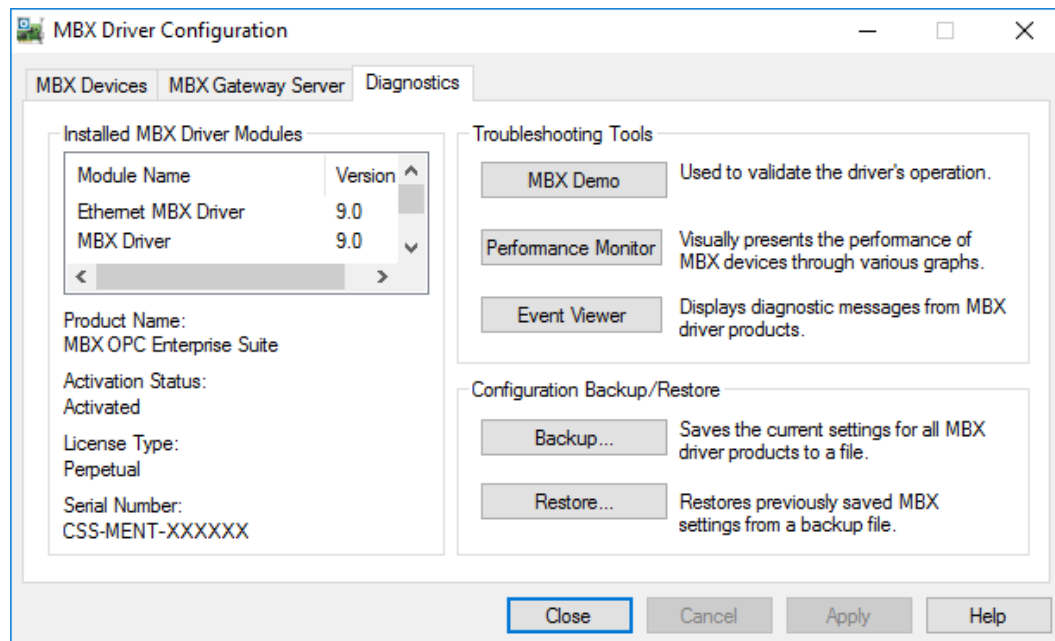
1. Select the *Diagnostics* tab of the MBX Driver Configuration editor.

2. Click the *Backup...* button.

3. Browse for the desired backup directory. By default, the last-used directory will be selected.

4. Enter the **File name** you want to use for your configuration backup file, and then click the **Save** button to complete the backup operation.

CONFIGURATION EDITOR REFERENCE

Before the Serial MBX Driver can be used, it must be properly configured. The configuration procedure involves creating one or more Serial MBX devices and configuring them to work with your network's addressing scheme. A Serial MBX device is a logical device that emulates the behavior of a Modbus Plus network adapter card. Your software applications will then use this logical device to communicate over the network.

This section provides a detailed description of each of the configuration editor features. If you are a new user and want a procedure to guide you through a typical configuration session, refer to the Quick-Start Guide.

To create a Serial MBX device, you must run the MBX Driver Configuration Editor after you install the software. The MBX Driver Configuration Editor is a common component of all drivers in the MBX family.

When configuring a Serial MBX device, the MBX Driver Configuration Editor automatically dispatches either the Serial MBX Master Configuration Editor or the Serial MBX Slave Configuration Editor, depending on the type of device you are editing.

MBX Driver Configuration Editor

The MBX Driver Configuration Editor is a common component of all drivers in the MBX family. It is used to create MBX devices, configure the MBX Gateway Server and provide access to diagnostic information and utilities. When you create or edit a Serial MBX device, the MBX Driver Configuration Editor automatically dispatches the Serial MBX Configuration Editor.

To start the MBX Driver Configuration Editor, open the Windows Start menu and go to the product suite you have installed. Next, open the Configuration submenu and select MBX Device Drivers.

The MBX Driver Configuration Editor consists of three tabs:

- MBX Devices Tab
- MBX Gateway Server Tab
- Diagnostics Tab

The following sections provide complete descriptions of these tabs.

MBX Devices Tab

Every MBX device must be configured on the MBX Devices tab before it can be used by client applications. The MBX Devices tab lists all currently configured MBX devices in your system. The information is provided in three columns: MBX Device, Device Type and Device Description.
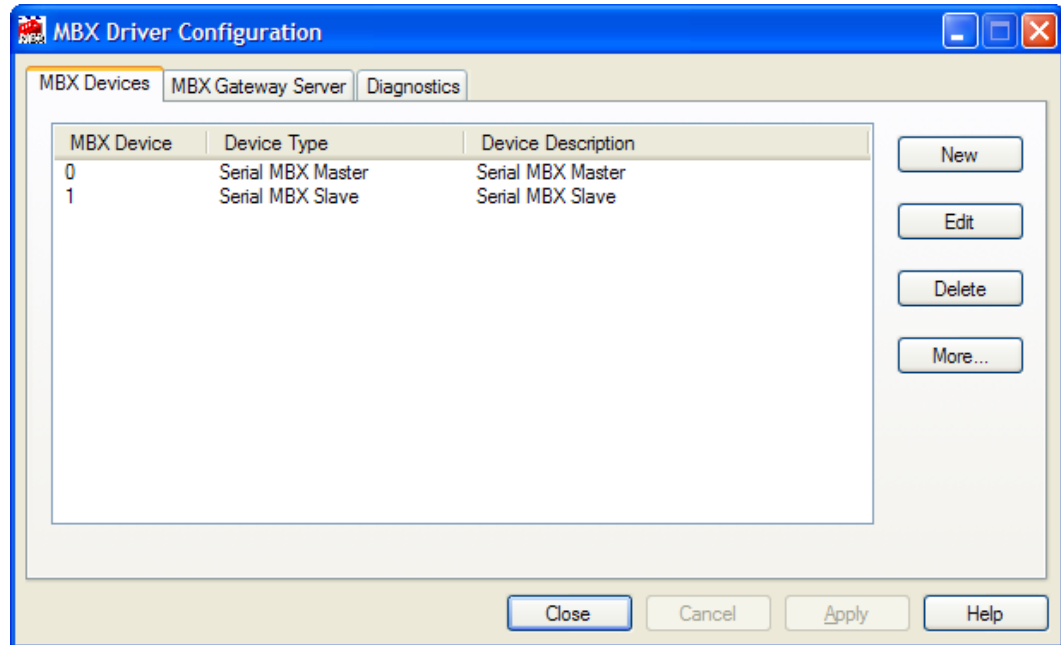
MBX Device

This column contains a device number that the editor assigns to every MBX device installed in the system. This is not the Modbus node address. By default, the editor will try to use consecutive numbers for the devices starting from 0. However, this is not a requirement.

Device Type

This column identifies the type of the MBX device, such as PCIe-85, Ethernet MBX or Serial MBX Master.

Device Description

This is a user-assigned text for device description. During device creation, a default description text will be assigned. Refer to the Changing the Device Description section, below, for information on how to modify this text.

The device description text has no effect on the MBX device operation. However, some applications using this device may be able to show this text.

New

Click this button to create a new MBX device.

Edit

Select an MBX device and click this button to edit it.

Delete

Select an MBX device and click this button to delete it.

More…

Select an MBX device and click this button for additional editing features. You can change the device type or edit the Device Description field.

Creating a New MBX Device

Click the **New** button or right-click inside the list window and select **New** from the context menu. Then select a host interface adapter or other device type from the drop-down list.

Upon selecting the device type, the MBX Driver Configuration Editor will automatically dispatch the configuration editor that is appropriate for that device.

Deleting an Existing MBX Device

Select the device and click the **Delete** button or right-click and select **Delete** from the context menu.

Editing an Existing MBX Device configuration

Select the device and click the **Edit** button or right-click and select **Edit** from the context menu. The MBX Driver Configuration Editor will automatically dispatch the appropriate device configuration editor. The screen that follows will depend on the type of device you selected.

Changing the Device Description

Select the device and click the **More…** button or right-click and select **Edit Description** from the context menu. Modify the device description and press the **Enter** key when you are done.

Changing the Device Type

Select the device and click the **More…** button or right-click and select **Change Type** from the context menu. From the drop-down list select the new device type for your MBX device. Upon selecting the new device type, the MBX Driver Configuration Editor will automatically dispatch the appropriate device configuration editor. The screen that follows will depend on the device type selected.

MBX Gateway Server Tab

The Serial MBX Driver comes with the MBX Gateway Server, a remote connectivity component of the MBX family. The MBX Gateway Server allows remote nodes to access

all configured MBX devices present on the system that is running the MBX Gateway Server. Refer to the MBX Gateway Driver section for more information on this capability.
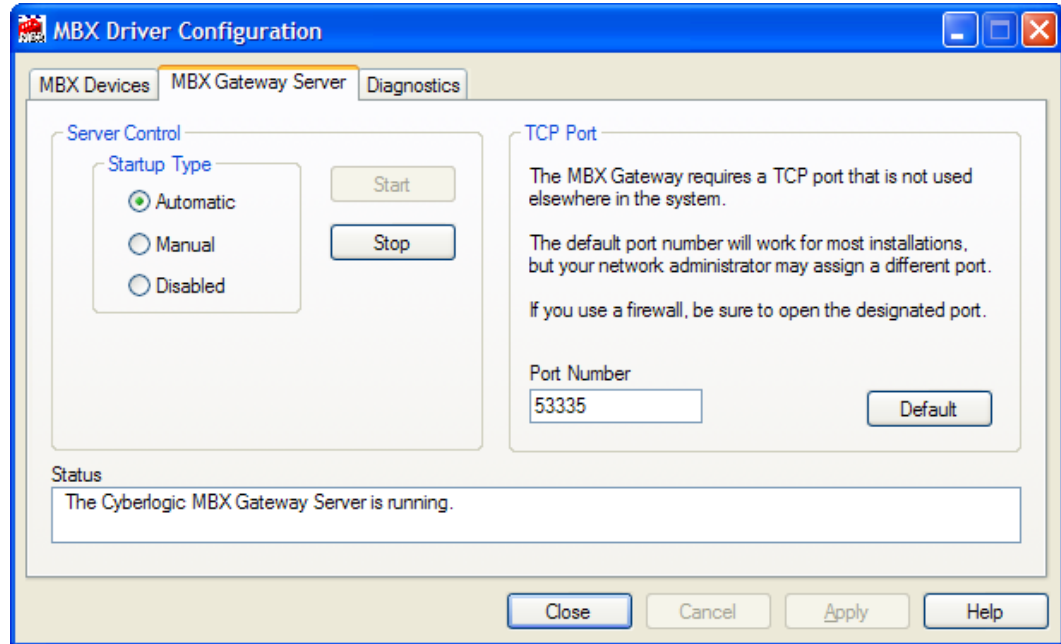
Server Control

This section allows you to designate if and how you want the MBX Gateway Server to start.

Automatic

When this option is selected, the MBX Gateway Server will start when Windows boots.

Manual

When this option is selected, the MBX Gateway Server will not start when Windows boots, but you can control it manually using the Start and Stop buttons.

Disabled

When this option is selected, the MBX Gateway Server will not run.

Start

In Automatic or Manual mode, click this button to start the MBX Gateway Server.

Stop

In Automatic or Manual mode, click this button to stop the MBX Gateway Server.

Status

This tells you whether the MBX Gateway Server is running, stopped, starting or stopping.

TCP Port

The port used here must not be used elsewhere in the system. If your system uses a firewall, the port must be opened in the firewall configuration. For more information, refer to the Configuring the Firewall section.

Port Number

Enter the number of the TCP port you wish to use.

Default

Click this button to restore the TCP port value to its default setting of 53335.

Selecting the Startup Type

If you want to use the MBX Gateway Server and want it to start whenever the system is booted, select **Automatic**. This is the recommended setting for systems that will use the Gateway Server.

If you want to use the MBX Gateway Server and want to control it manually, choose **Manual**. The Server will not start on boot-up; instead you must use the Start and Stop buttons to control it.

If you do not want to use the MBX Gateway Server, choose **Disabled**.

Start/Stop the Gateway Server

Click the **Start** or **Stop** button.

Selecting the TCP Port

Enter the desired port number in the Port Number field.

You must enter a TCP port that is not used elsewhere in the system. The default, 53335, will work for most installations, but this port may be taken in some unusual cases. If that applies to your system, the system administrator will assign a different port.

Configuring the Firewall

If your system uses a firewall, you must configure it to permit MBX Gateway communication. The procedure shown here is for the Windows 7 firewall. The exact procedure for your system will depend upon the firewall you are using, but the issues are the same for all firewall types.

1. To configure Windows 7's firewall, go to **Control Panel** and open **Windows Firewall** and select **Advanced Settings**.
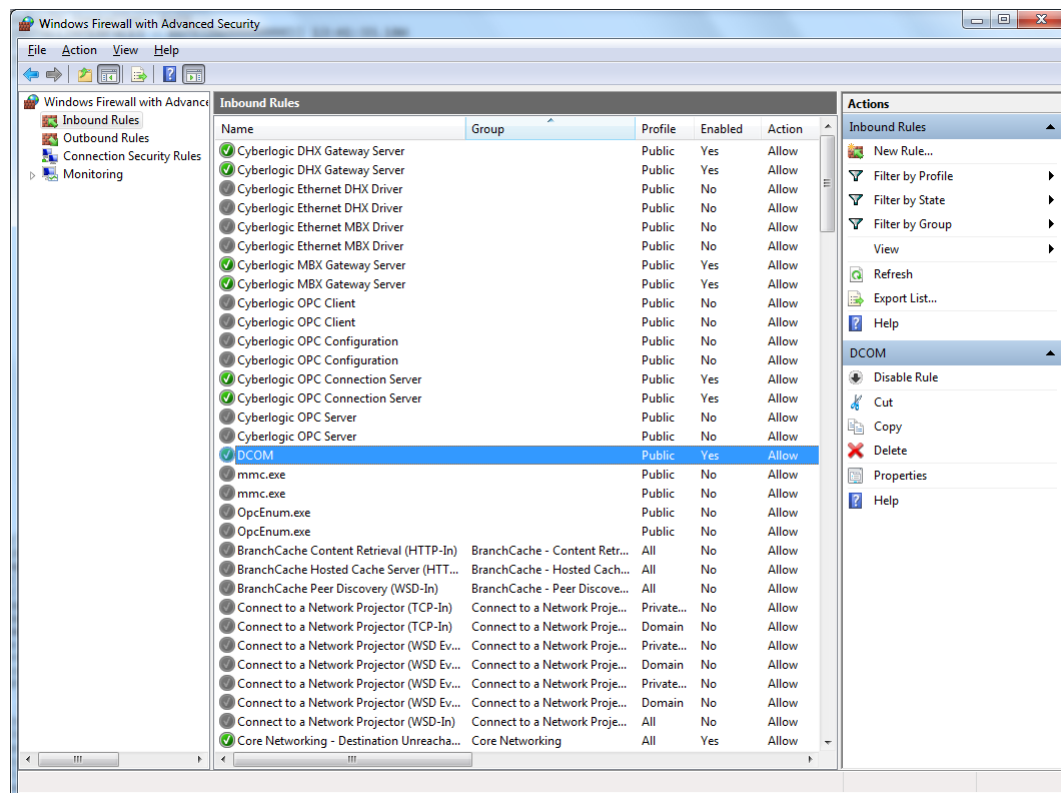
2. Select **Inbound Rules**.

3. Verify that the rules **Cyberlogic MBX Gateway Server** and **DCOM** are enabled. If not, enable them.

4. Close the window to exit.

Diagnostics Tab

The diagnostic features will help you to confirm that the driver is running and is properly configured. They will also provide important help if troubleshooting or technical support is needed.
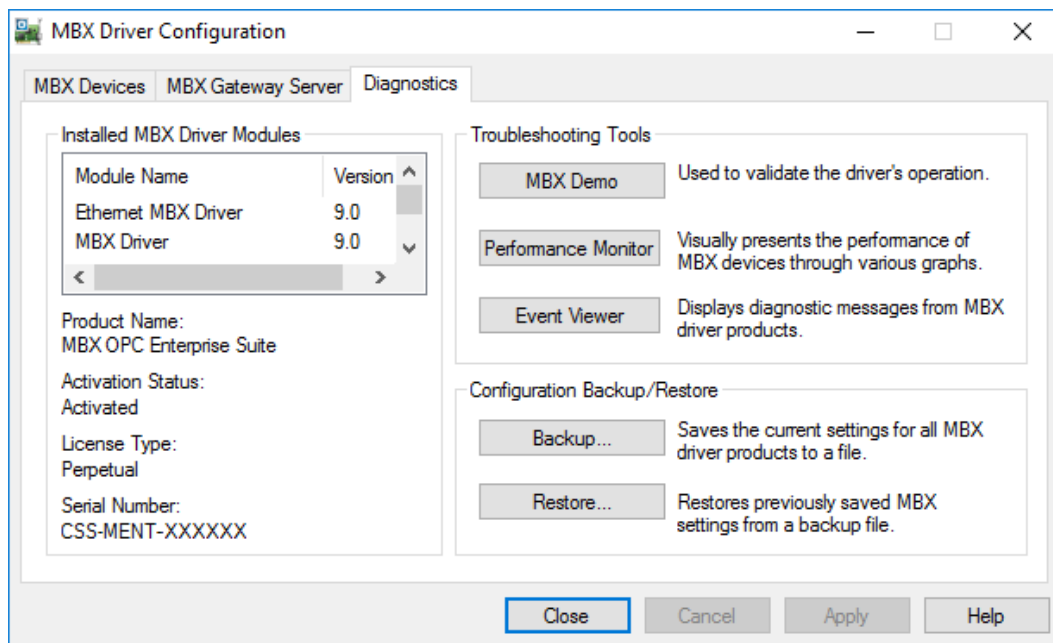
Installed MBX Driver Modules

This area shows all MBX product components installed on your system, along with their version numbers. This information may be requested if you call for technical support. This screen also tells you if the software has been activated or if it is running in demo mode.

Product Name

MBX products are sold and installed as packaged suites, such as the MBX Driver Suite and MBX OPC Server Suite. This field indicates one of the MBX family suites that is installed on your system. For complete information on all installed Cyberlogic products go to the **Product Information** app accessable from the Windows **Start** menu.

Activation Status

Most Cyberlogic software products operate in a time-limited demonstration mode until they are activated. This field tells you whether or not the installed product has been activated.

If your product requires activation, run the **Activation** wizard, which you will find in the Windows **Start** menu under **Cyberlogic Suites**. You will need the serial number and password that were assigned when you purchased your license for the software.

Note	Some OEM versions of MBX products are pre-activated and do not require you to take any additional activation steps.

License Type

This field shows the licensing mode that the software is operating under. If the type displayed is 2 Hour Demo, the software will run for only two hours at a time, after which you must restart the system to obtain another two hours of use. To enable continuous, uninterrupted operation, you must activate the software.

Serial Number

If you have activated the software by entering the serial number and password, the serial number used will be shown here. This will help you to determine which license goes with which of your systems.

Troubleshooting Tools

The Troubleshooting Tools group provides shortcuts to diagnostic tools that will help you to verify that your drivers are operating as expected. In case of communication problems, these tools will help in the diagnosis.

For details on how to use these tools, refer to the Validation & Troubleshooting section.

MBX Demo

Run this program after configuring the driver to confirm that it is configured correctly and running properly.

Performance Monitor

Click this button to launch the Windows Performance Monitor, which will allow you to observe numerous performance parameters in graphical form.

Event Viewer

In case of communication difficulties, the Windows Event Viewer may provide error messages to guide you in troubleshooting problems.

Configuration Backup/Restore

The Backup… and Restore… buttons in this group can be used to backup and restore configurations of all MBX family drivers on your system.
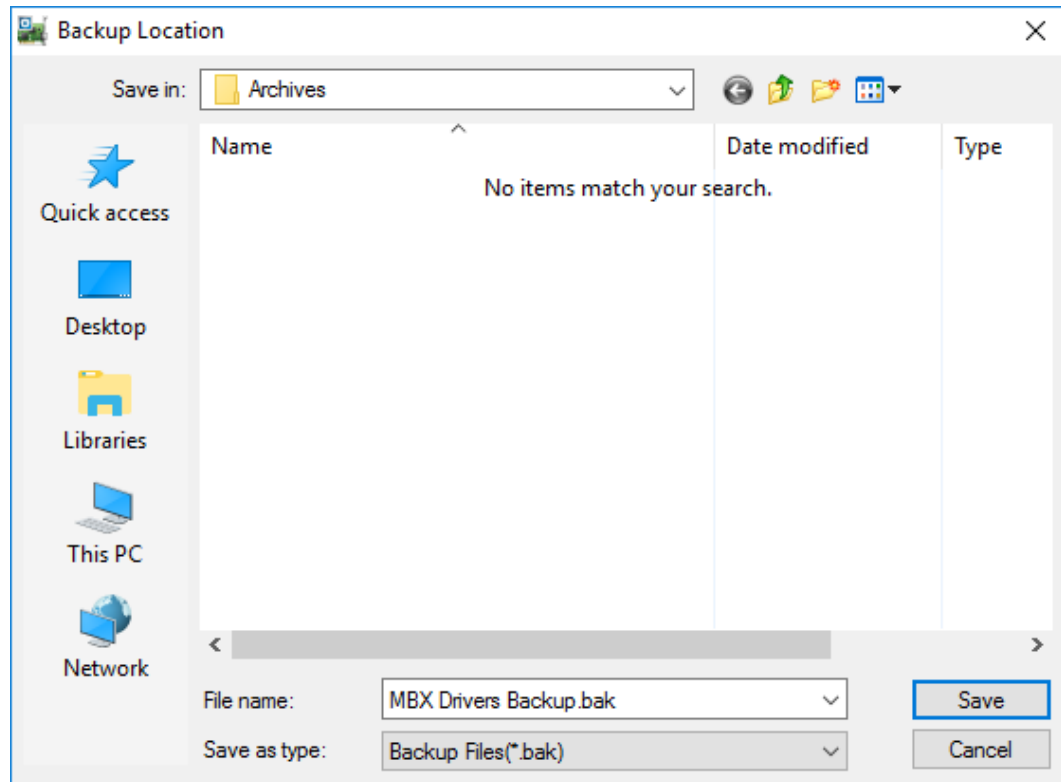
Note	We strongly recommend that you backup your configuration data after the initial configuration and that you maintain up-to-date backups after every configuration change.

Backup Configuration

Use this procedure to backup your configuration.
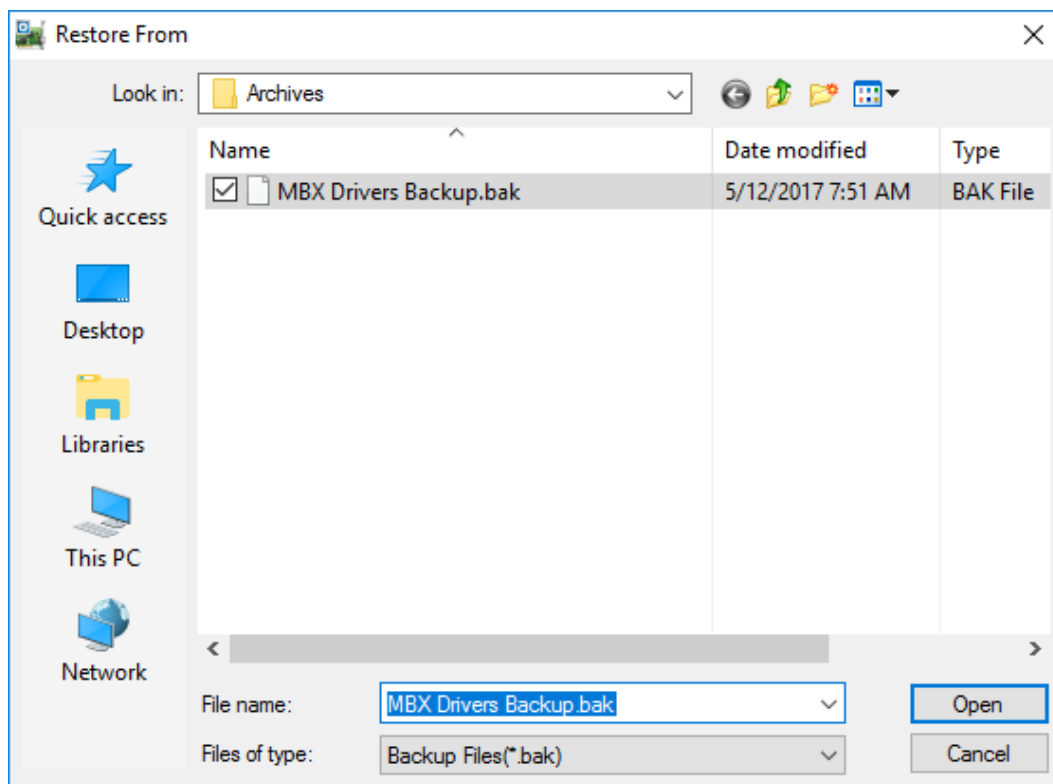
1. Click the **Backup...** button.

2. Browse for the backup directory. By default, the last-used directory will be selected.

3. Enter the **File name** you want to use for your configuration backup file, and then click the **Save** button to complete the backup operation.

Restore Configuration

To restore a configuration that was previously backed up, use this procedure.

1. Click the **Restore...** button.

2. Browse for your configuration backup file. By default, the last-used directory will be selected.

3. Select the backup file and click the **Open** button to complete the restore operation.

Caution!	After you finish restoring the configuration, restart the system to ensure proper operation of the restored devices.

Configuration Backup/Restore Utility

The MBX products include a utility program, ClMbxCfg.exe, that you can use to backup and restore MBX device configurations. The program is located in the \Program Files\Common Files\Cyberlogic Shared\ directory.

The utility accepts the following command line switches:

/Save *FileName*	Save configuration
/Restore *FileName*	Restore configuration
/Q	Quiet operation (No error or warning messages)
/?	Help
/H	Help

For example, to backup the configuration of all MBX devices to a file named MbxCfg.bak, located in the directory C:\Program Files\Common Files\Cyberlogic Shared\, use the following command line:

>ClMbxCfg /Save C:\Program Files\Common Files\Cyberlogic Shared\MbxCfg.bak

To restore the configuration that the previous command saved, use the following command:

>ClMbxCfg /Restore C:\Program Files\Common Files\Cyberlogic Shared\MbxCfg.bak

You can use different file names to maintain different versions of your backups. However, for most users, a single backup is sufficient.

Serial MBX Master Configuration Editor

When editing Serial MBX Master devices, the MBX Driver Configuration Editor dispatches the Serial MBX Master Configuration Editor.

Note	This section describes the editor for serial master devices. If you are editing a serial slave device, go to the reference for the Serial MBX Slave Configuration Editor.

The Serial MBX Master Configuration Editor consists of three property tabs:

- Settings Tab
- Active Nodes Tab
- Driver Control Tab

The following sections provide complete descriptions of these tabs.

Settings Tab

The Settings tab allows you to configure the serial port settings, to choose whether you want to use RTU or ASCII mode, and to select the Modbus Plus equivalent node address for the device.
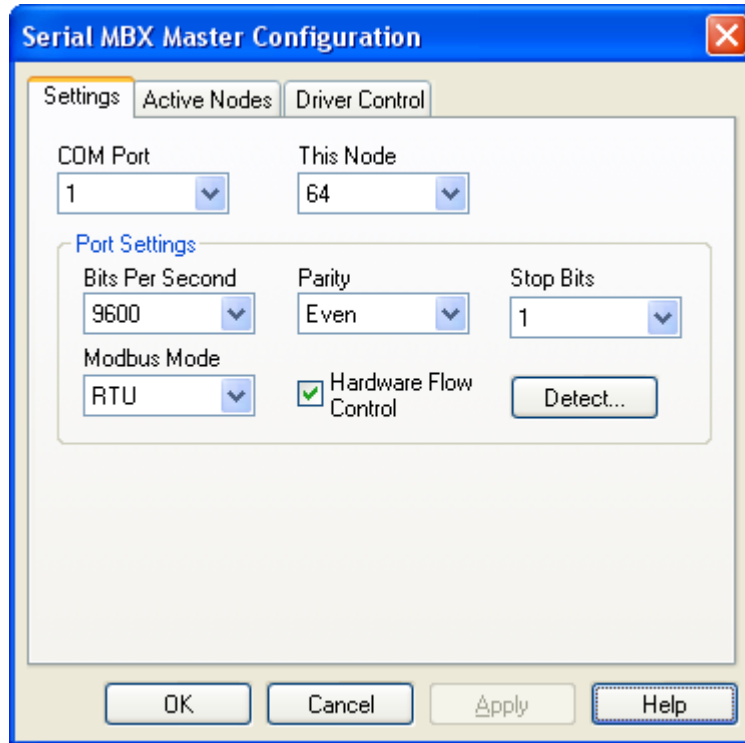
COM Port

This setting selects the serial COM port for this device.

Each Serial MBX Master device has an associated serial COM port. The assigned COM port belongs exclusively to a single master device and is not shared with other Serial MBX devices or other applications in the system.

This Node

This allows you to assign a Modbus Plus equivalent node address to the device by selecting it from the drop-down list. By default, node address 64 is used.

Bits Per Second

This setting specifies how fast the data bytes are transmitted by a COM port. All Modbus nodes must have the same Bits Per Second setting. The default is 9600.

Parity

The Parity setting controls how the COM port checks for bit errors, which may occur in transmission of the data. Selecting Even or Odd parity allows detection of these errors. Selecting None disables parity checking. All Modbus nodes must have the same Parity setting. The default is Even.

Stop Bits

Stop bits frame data bytes in asynchronous communication. They tell the receiving COM port that a byte has been sent. The Stop Bits field allows 1 or 2 stop bits to be selected. All Modbus nodes must have the same Stop Bits setting. The default setting is 1.

Modbus Mode

Modbus protocol allows for either an RTU or ASCII mode of operation. For better performance use the RTU mode. However, all devices connected to a Modbus network must use the same communication mode. The default setting is RTU.

Hardware Flow Control

Hardware flow control or "handshaking" is enabled by checking this box. By default, this box is checked and hardware flow control is enabled.

For detailed information on this option, refer to the Hardware Flow Control discussion in the Communication Using the Serial MBX Driver section.

Caution!	Not all serial ports support handshaking. When using hardware flow control, always verify that the communication module you are connecting to supports hardware flow control, and that it has the same settings as the Serial MBX device you have configured.

Detecting Port Settings

If you do not know the correct Port Settings, connect the serial cable to the network and click the **Detect...** button. You will be asked to narrow down the range of possible settings and node addresses, if you can, and the remaining combinations will be tried until a working set of values is found.

Note	If you do not narrow down the possibilities at all, every possible combination of port settings will be tried on all 247 node addresses. This could require over 38,000 tests and may take hours to complete.

The detection works by sending a test message to each of the nodes in the address range you specify. If the node fails to respond, a different combination of port settings is tried on the same node. If all combinations fail, the process is repeated on another node. This continues until communication is established with a node or all possibilities have been exhausted.

Caution!	For the port settings detection to work, your system must be connected to a network that has at least one other working node.

Active Nodes Tab

This tab allows you to deactivate network nodes for which there is no device present on the network. It also allows you to re-map node addresses.

Caution!	Some application programs will attempt to communicate with every active node on the network. If the corresponding devices are not physically present, it will be unable to do so, resulting in errors and performance problems.
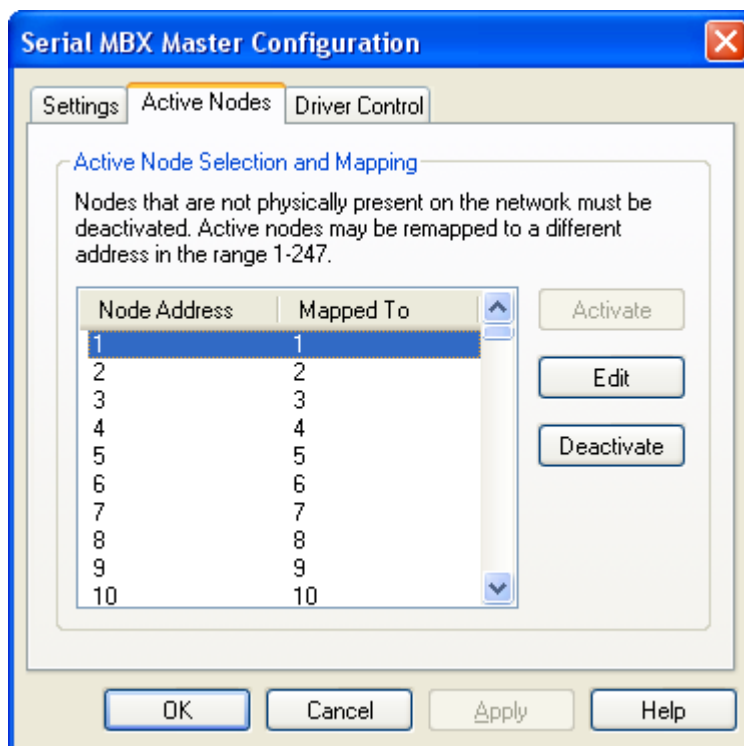
Node Address

This is the address that you want to activate, deactivate or map into a different address. It is the address that the application will use.

Mapped To

This is the physical node on the network that the Node Address is mapped to.

Activate

Click this button to activate the selected node.

Edit

Click this button to change the mapping of the selected node.

<u>*Deactivate*</u>

Click this button to deactivate the selected node.

Deactivating a Node

To deactivate a node, select it in the **Node Address** column, and then click the **Deactivate** button. To deactivate multiple nodes, you can use the Ctrl-Click and Shift-Click functions to select more than one node.

Activating a Node

To activate a node, select it in the **Node Address** column, and then click the **Activate** button. To activate multiple nodes, you can use the Ctrl-Click and Shift-Click functions to select more than one node.
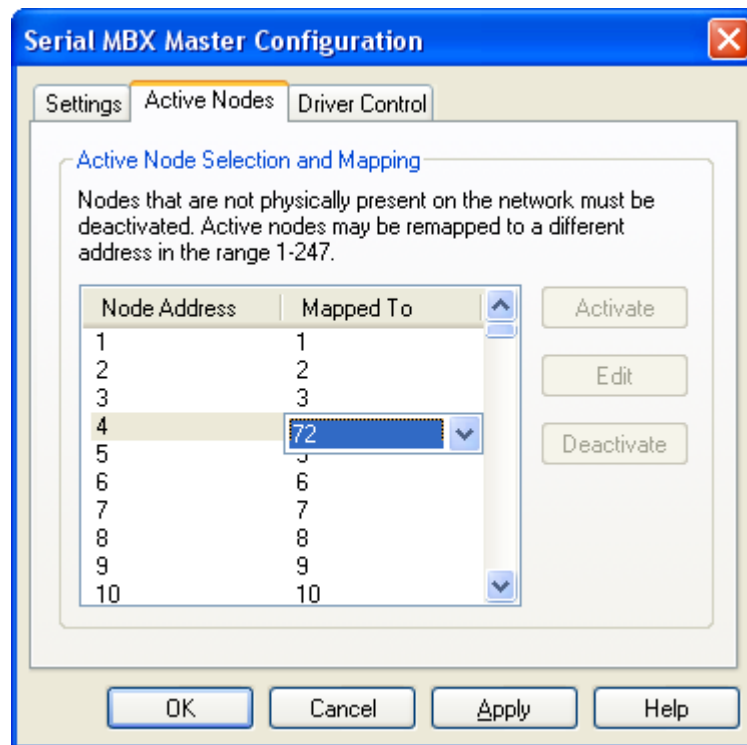
Editing a Node

Select a device and click the **Edit** button. A drop-box opens in the Mapped To column, permitting you to map the selected node address to a different address. Select the desired mapping from the selection box.

For a discussion of node mapping, refer to the <u>Active Node Selection and Mapping</u> section.

Driver Control Tab

The Driver Control tab allows you to set the startup type and monitor the current driver status.

Caution!	These settings are global and are common to all Serial MBX Master and Serial MBX Slave devices.

Automatic

When this option is selected, the Serial MBX Driver will start when Windows boots.

Manual

When this option is selected, the Serial MBX Driver will not start when Windows boots, but you can control it manually using the Start and Stop buttons.

Disabled

When this option is selected, the Serial MBX Driver will not run.

Start

In Automatic or Manual mode, click this button to start the Serial MBX Driver.

Stop

In Automatic or Manual mode, click this button to stop the Serial MBX Driver.

Driver Status

This tells you whether or not the Serial MBX Driver is running, stopped, starting or stopping.

Selecting the Startup Type

Select the desired mode among the Startup Type choices.

If you want the Serial MBX Driver to start whenever the system is booted, select **Automatic**. This is the recommended setting for systems that will use the Serial MBX Driver.

If you want to use the Serial MBX Driver and want to control it manually, choose **Manual**. The driver will not start on boot-up; instead you must use the Start and Stop buttons to control it.

If you do not want to use the Serial MBX Driver, choose **Disabled**.

Start/Stop the Ethernet MBX Driver

Click the **Start** or **Stop** button.

Serial MBX Slave Configuration Editor

When editing Serial MBX Slave devices, the MBX Driver Configuration Editor dispatches the Serial MBX Slave Configuration Editor.

Note   This section describes the editor for serial slave devices. If you are editing a serial master device, go to the reference for the Serial MBX Master Configuration Editor.
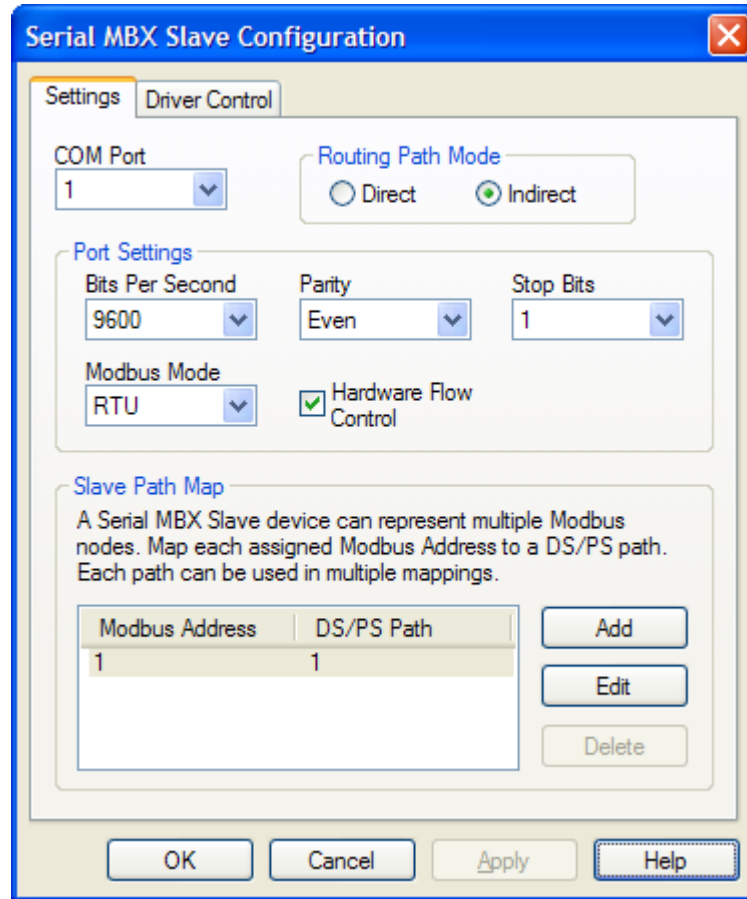
The Serial MBX Slave Configuration Editor consists of two property tabs:

- Settings Tab
- Driver Control Tab

The following sections provide complete descriptions of these tabs.

Settings Tab

The Settings tab allows you to configure the serial port settings, to choose whether you want to use RTU or ASCII mode, and to select the Modbus Plus equivalent node address for the device.

COM Port

This setting selects the serial COM port for this device.

Each Serial MBX Slave device has an associated serial COM port. Unlike the master device, one COM port can be shared by multiple Serial MBX Slave nodes. However, the port cannot be shared with any other application in the system. Refer to the Communication Using the Serial MBX Driver section for additional details.

Routing Path Mode

This selection allows you to choose either Direct or Indirect Addressing, depending on the needs of your device and software.

In general, applications that can accept routing array bytes in the range of 1-247 will use Direct mode, and those that can accept routing array bytes in the range of only 1-64 will use Indirect mode. Refer to the Serial MBX Device Addressing Modes section for a complete discussion of these modes.

Bits Per Second

This setting indicates how fast the data bytes are transmitted by a COM port. All Modbus nodes must have the same Bits Per Second setting. The default is 9600.

Parity

The Parity setting controls how the COM port checks for bit errors, which may occur in transmission of the data. Selecting Even or Odd parity allows detection of these errors. Selecting None disables parity checking. All Modbus nodes must have the same Parity setting. The default is Even.

Stop Bits

Stop bits frame data bytes in asynchronous communication. They tell the receiving COM port that a byte has been sent. The Stop Bits field allows 1 or 2 stop bits to be selected. All Modbus nodes must have the same Stop Bits setting. The default setting is *1*.

Modbus Mode

Modbus protocol allows for either an RTU or ASCII mode of operation. For better performance use the RTU mode. However, all devices connected to a Modbus network must use the same communication mode. The default setting is RTU.

Hardware Flow Control

Hardware flow control or "handshaking" is enabled by checking this box. By default, this box is checked and hardware flow control is enabled.

For detailed information on this option, refer to the Hardware Flow Control discussion in the Communication Using the Serial MBX Driver section.

Caution!	Not all serial ports support handshaking. When using hardware flow control, always verify that the communication module you are connecting to supports hardware flow control, and that it has the same settings as the Serial MBX device you have configured.

Slave Path Map

The Slave Path Map allows you to map each Modbus node address into a pair of DS/PS paths that share the same path number.

This permits the Serial MBX Slave device to answer to multiple slave addresses, thus functioning as a set of virtual Modbus slaves. Refer to the Communication Using the Serial MBX Driver section for additional details. At least one entry in this map is required for a Serial MBX Slave device to function.

Note	A Serial MBX Slave device emulates the behavior of a Modbus Plus adapter card, therefore it must have its own, single node address. Even if multiple Modbus addresses are assigned in the Slave Path Map, this device will always report its node address as 1. The master node will always be identified as node address 2.

Adding a New Map Entry

Click the **Add** button, or right-click inside the list window and select **Add** from the context menu. Select the **Modbus Address** and the corresponding **DS/PS Path** number.

Deleting an Existing Map Entry

Select the map entry in the list and click the **Delete** button, or right-click and select **Delete** from the context menu.
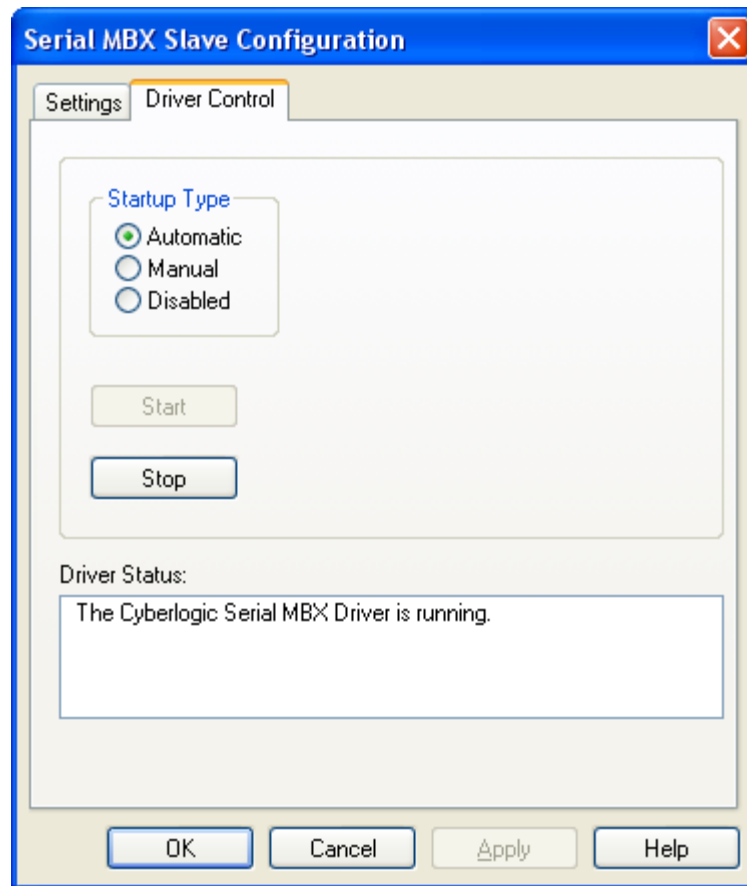
Editing an Existing Map Entry

Select the map entry in the list and double-click it, or click the **Edit** button, or right-click on the entry and select **Edit** from the pop-up menu. You can change only the DS/PS Path number assigned to the selected Modbus node address.

Driver Control Tab

The Driver Control tab allows you to set the startup type and monitor the current driver status.

Caution!	These settings are global and are common to all Serial MBX Master and Serial MBX Slave devices.

Automatic

When this option is selected, the Serial MBX Driver will start when Windows boots.

Manual

When this option is selected, the Serial MBX Driver will not start when Windows boots, but you can control it manually using the Start and Stop buttons.

Disabled

When this option is selected, the Serial MBX Driver will not run.

Start

In Automatic or Manual mode, click this button to start the Serial MBX Driver.

Stop

In Automatic or Manual mode, click this button to stop the Serial MBX Driver.

Driver Status

This tells you whether or not the Serial MBX Driver is running, stopped, starting or stopping.

Selecting the Startup Type

Select the desired mode among the Startup Type choices.

If you want the Serial MBX Driver to start whenever the system is booted, select **Automatic**. This is the recommended setting for systems that will use the Serial MBX Driver.

If you want to use the Serial MBX Driver and want to control it manually, choose **Manual**. The driver will not start on boot-up; instead you must use the Start and Stop buttons to control it.

If you do not want to use the Serial MBX Driver, choose **Disabled**.

Start/Stop the Ethernet MBX Driver

Click the **Start** or **Stop** button.

VALIDATION & TROUBLESHOOTING

The following sections describe how the MBX Demo and Performance Monitor are used to verify that the Serial MBX devices are configured correctly.

If you are having difficulties communicating through a Serial MBX device, the troubleshooting sections can help you determine the nature of the problem. Included is a description of the Event Viewer, a list of Serial MBX Driver Messages and a Frequently Asked Questions section.

MBX Demo
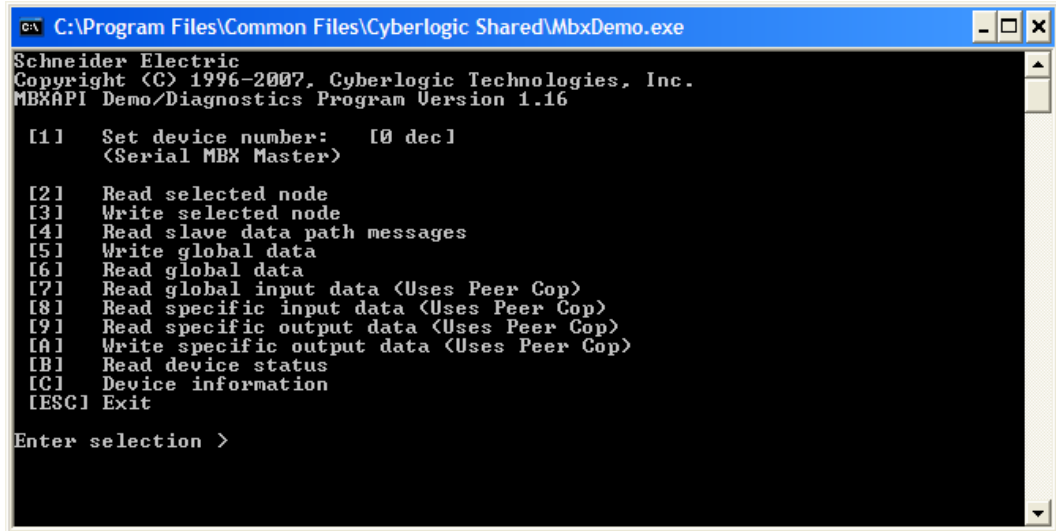
The MBX Demo program can be used to test all configured MBX devices in a system for proper operation. To run the program, open the Windows **Start** menu and locate the **Cyberlogic Suites** sub-menu. From there, go to **Diagnostics** and select **MBX Demo**.

Alternatively, open the **MBX Driver Configuration Editor**, go to the **Diagnostics** tab and click **MBX Demo**.
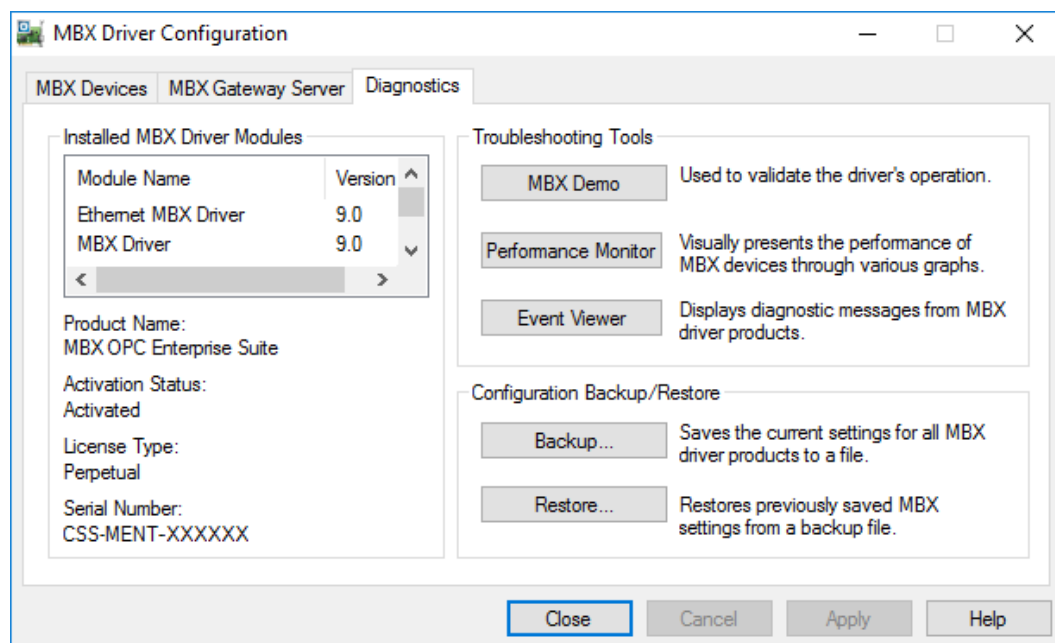
Main Menu

The MBX Demo will quickly access all available features of the configured MBX devices in your system, allowing you to verify their operation.
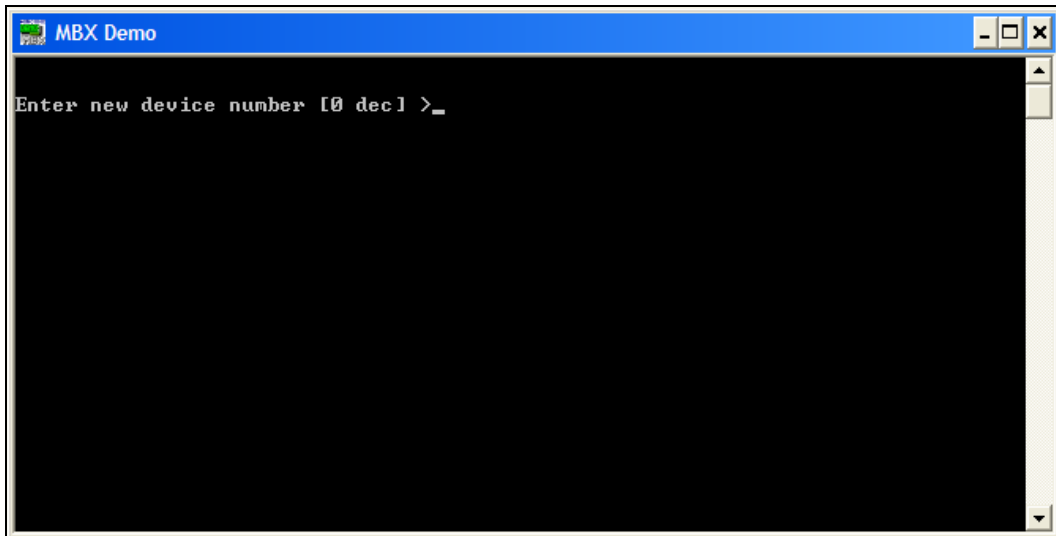
The simple command-line interface mimics earlier tools familiar to most users. It displays menu choices that take the user to secondary level screens.

Press **Esc** at any screen to return to the main menu shown above.

Press **Esc** in the main window to exit the program.
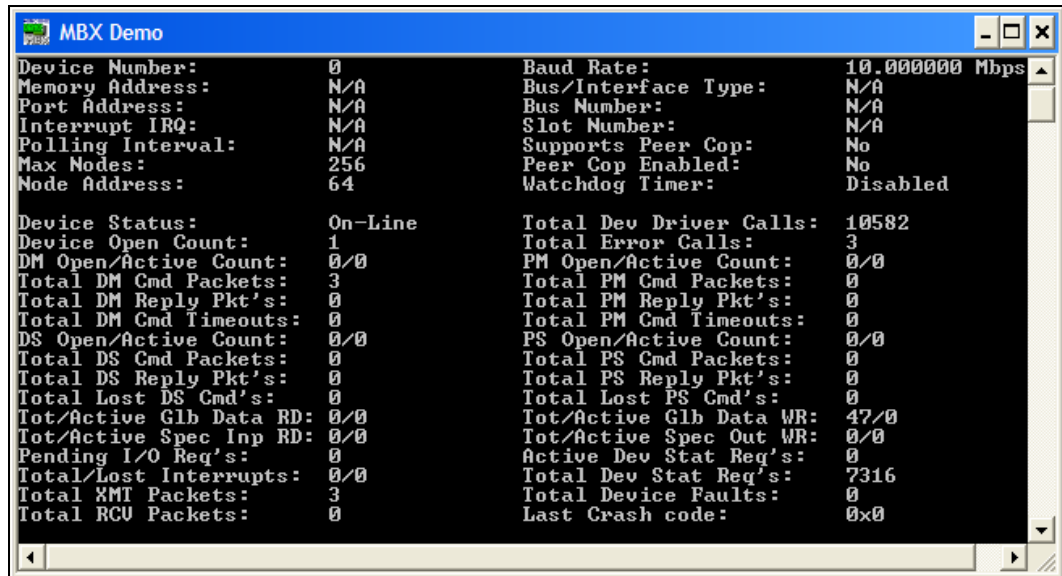
[1] Set Device Number

When the MBX Demo program starts, the device number defaults to 0. To change it, press **1**.

At the prompt, enter the desired device number and press the **Enter** key to change the selected device and return to the main menu.

Verify that the device type shown on the main menu matches the type you configured for the selected device number.

[B] Read Device Status

From the main menu, press **B**. This launches the device status screen, which shows all active nodes on the network.

In the Active Node List grid, the letter I designates the node you are working from and the letter U designates other nodes found on the network.

Note	Only nodes that are configured as active will be shown here.

Verify that all expected nodes are shown, and that their node addresses are correct. The node address for I should be the value you chose for This Node on the Settings tab for a Serial MBX Master device, or 1 for a Serial MBX Slave device.

Press **Esc** to return to the main menu.

[C] Device Information

From the main menu, press **C** to launch the Device information screen.

This screen shows configuration, statistical and diagnostic information about the driver, the device and the network.

After viewing the information, press **Esc** to return to the main menu.

[2] Read Selected Node

To read data from registers on a specific node, press **2**.

Enter the **routing path** of the node you want to read from. For a Serial MBX device, this is simply the node address.

The **reference value** is the register address of the data you want to read. If you want to read from more than a single register, enter the first register's reference value.

The **length** is the number of consecutive data items you want to read.

In the example shown, we want to read from the device at node 7, and will read five registers beginning with 40001, that is, registers 40001 – 40005.

Press **Enter** to initiate the read. The requested data will be displayed on the screen. Press **Esc** to return to the main menu.

Other Operations

The rest of the functions available on the main menu will perform read and write operations in a manner similar to the Read Selected Node function.

Performance Monitor

Microsoft provides a diagnostic tool, the Performance Monitor, as part of the Windows operating system. Applications supporting the Performance Monitor, including the MBX driver family, allow users to monitor relevant performance information. Multiple devices can be monitored simultaneously for comparison.

To run the program, open the Windows **Start** menu and locate the **Cyberlogic Suites** sub-menu. From there, go to **Diagnostics** and select **Performance Monitor**.

Alternatively, go to the Diagnostics tab of the MBX Driver Configuration Editor and click the **Performance Monitor** button.
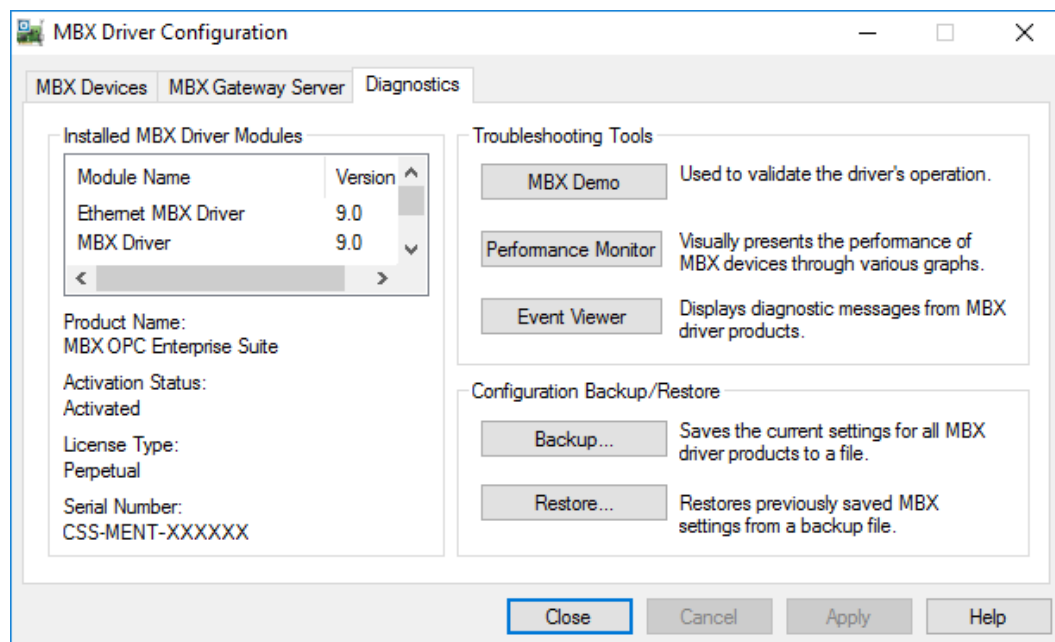
How to Use the Performance Monitor

Since extensive help is provided for this program by Microsoft, only a few points relevant to the MBX Driver family are shown here.
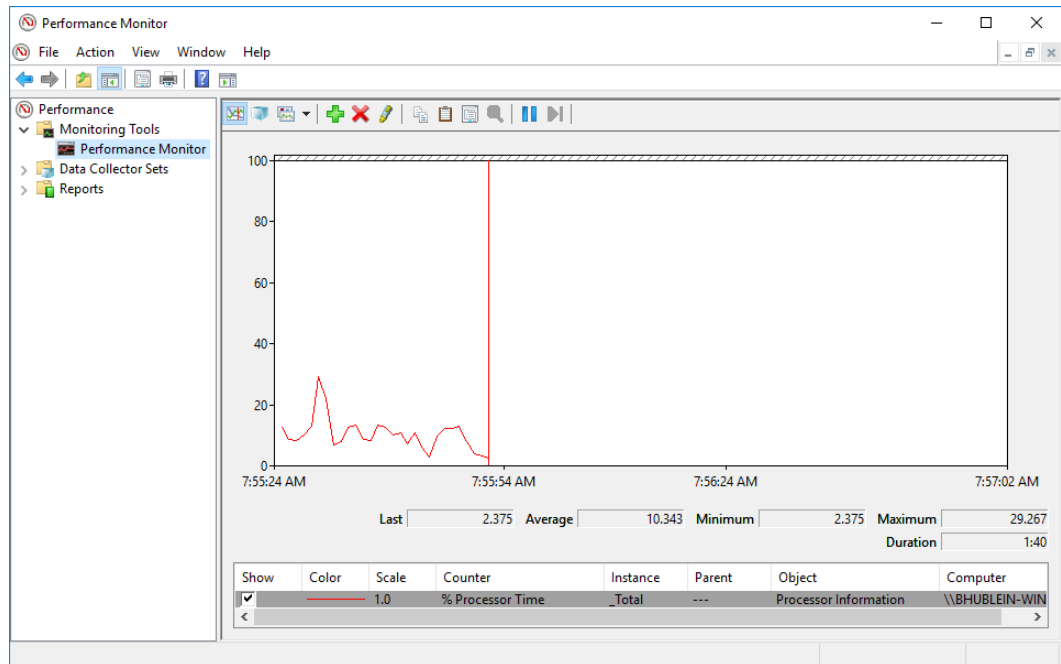
1. When the Performance Monitor program starts, click the **+** button on the tool bar.
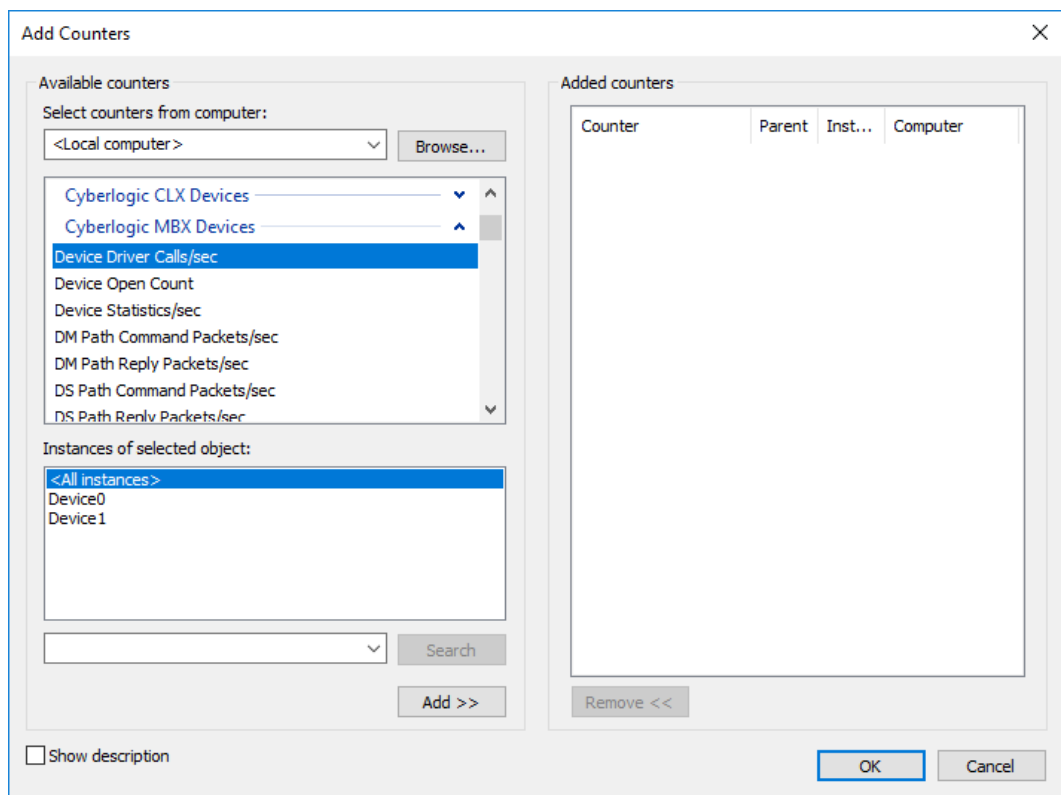
2. Select **Cyberlogic MBX Devices** from the **Available counters** list.
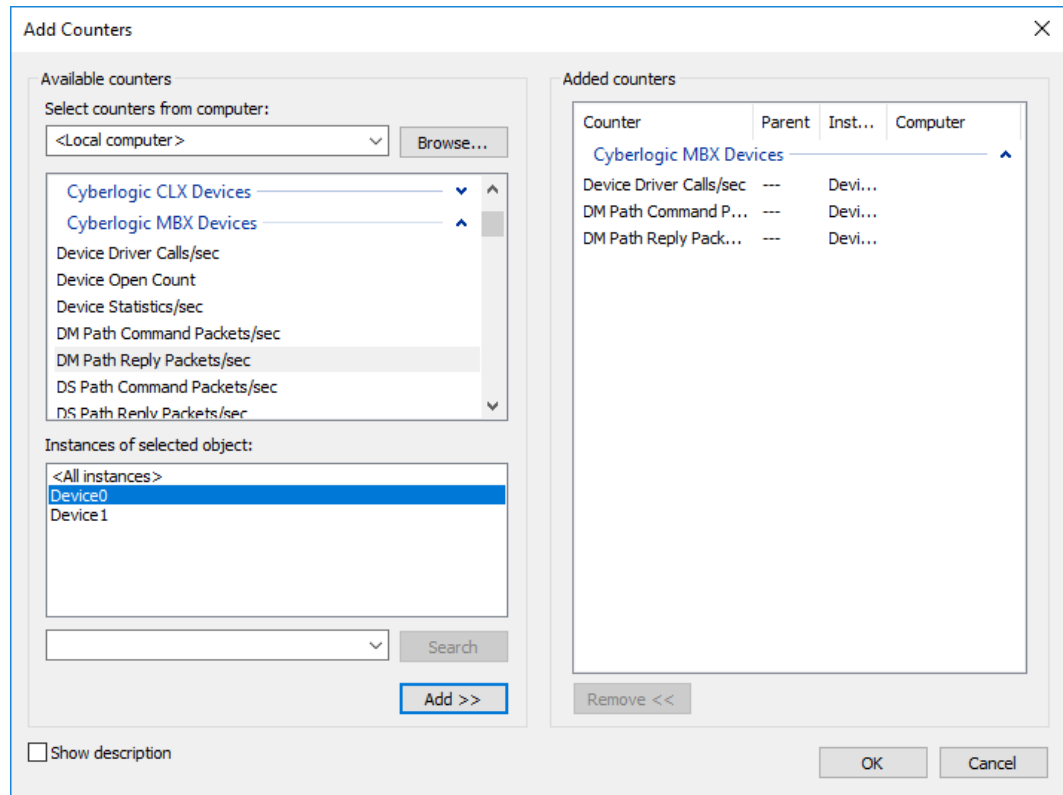
3. Choose a counter and the MBX device, and then click **Add**. Repeat this for all the counters you want to view.

4. Click **OK**. The counters you chose will then be displayed in graphical format.
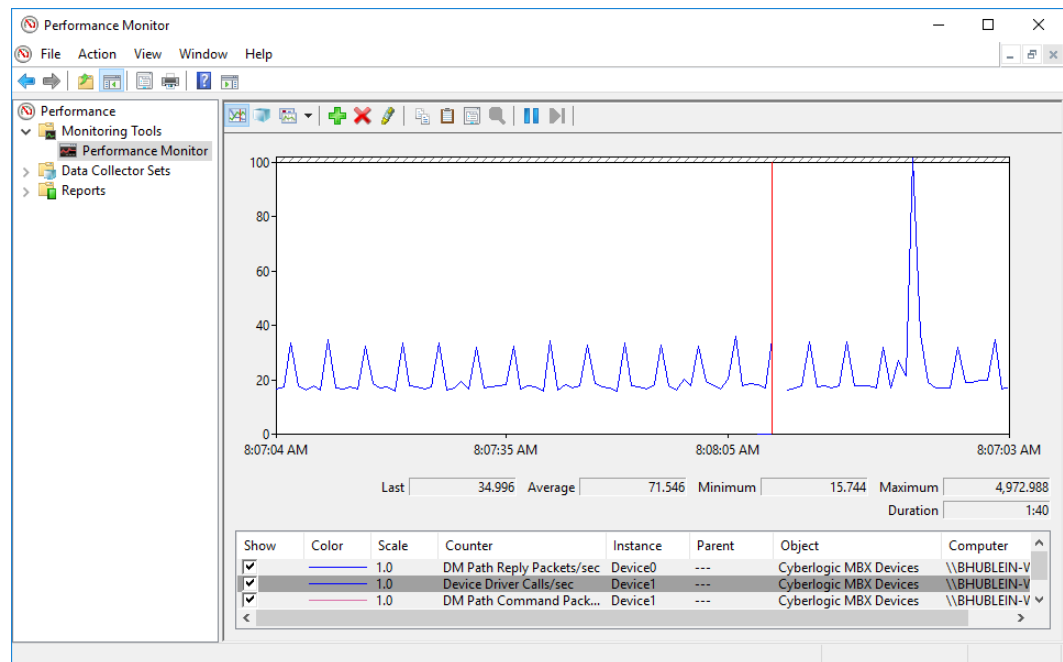
Event Viewer

During startup and operation, the MBX drivers may detect problems or other significant events. When a noteworthy event is detected, the driver sends an appropriate message to the Windows Event Logger. You can view these messages using the following procedure.

1. Open the Windows **Start** menu and locate the **Cyberlogic Suites** sub-menu. From there, go to **Diagnostics** and select **Event Viewer**.
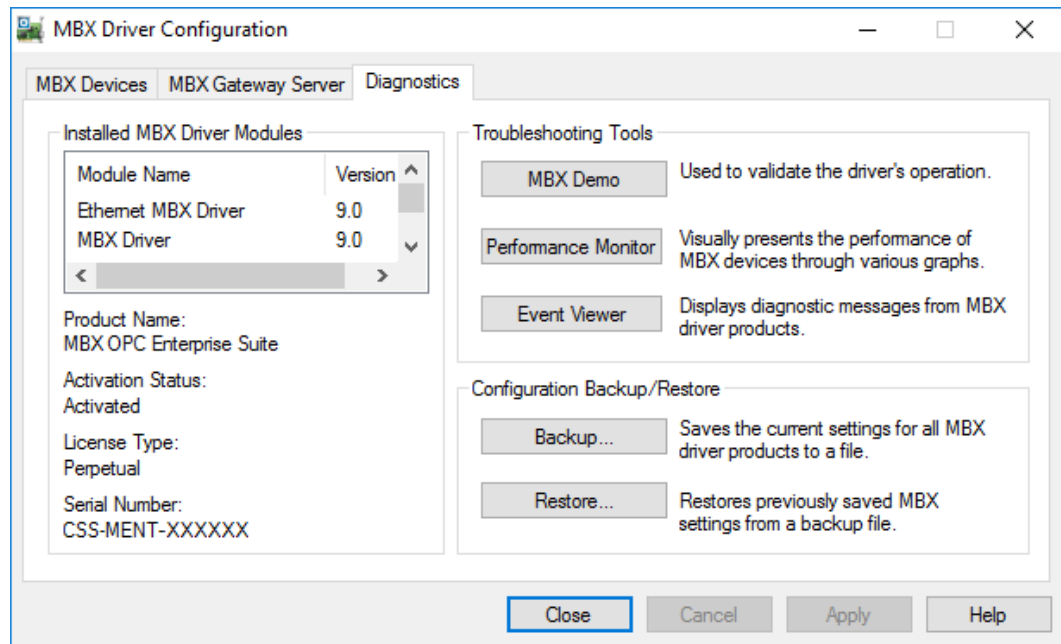
Alternatively, click the **Event Viewer** button on the Diagnostics tab of the MBX Driver Configuration Editor.
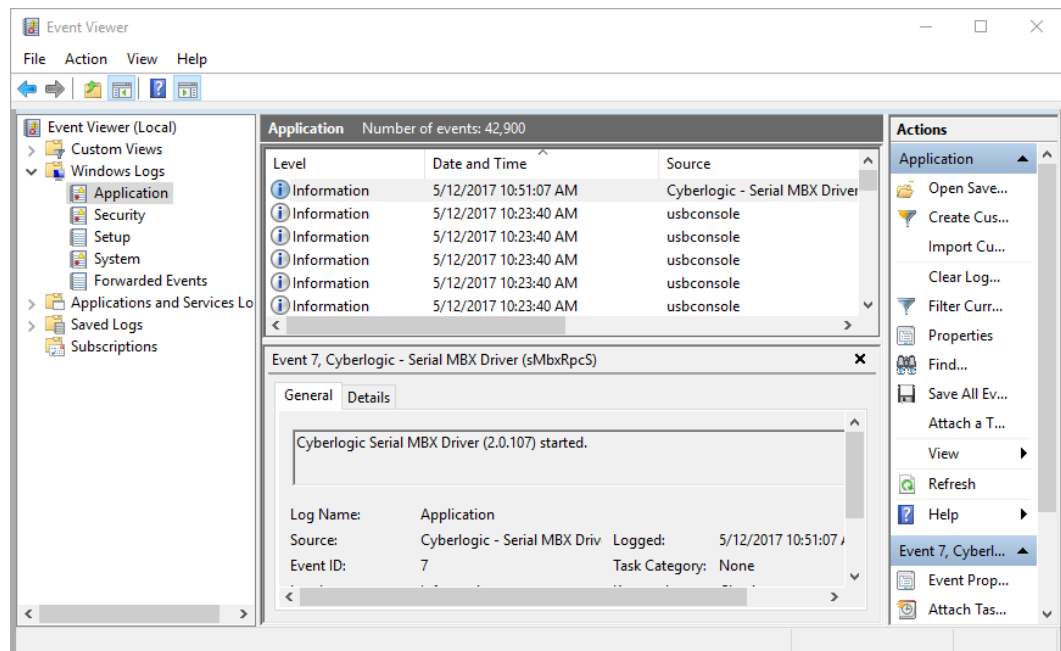
2. If you are looking for events relating to the Serial MBX Driver, select the **Windows Logs|Application** branch from the Event Viewer tree, and look for entries in the **Source** column named **Cyberlogic – Serial MBX Driver(sMbxRpcS)**.

    For other types of events, select the Application branch from the Event Viewer tree, and look for entries in the **Source** column that begin with **Cyberlogic**.

Caution!	The Event Viewer does not clear itself after rebooting. Check the time-stamps of the messages to be sure that you are not looking at an old error message.

3. Double-click on the desired entry to display a complete event message.
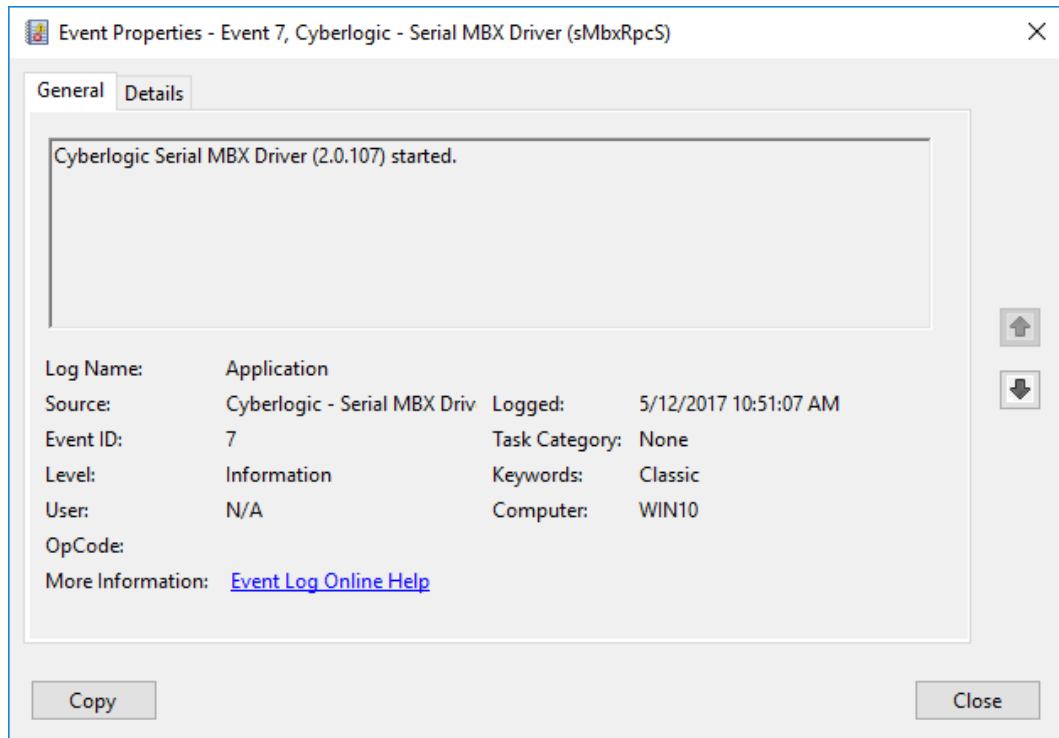
4. For further descriptions of the error log messages, refer to the Serial MBX Driver Messages section.

Serial MBX Driver Messages

SMBXAPIM.DLL failed to load. Reinstall the product.

A necessary DLL could not be loaded. This may indicate a corrupted installation. Repair the existing installation, or remove and reinstall the software.

Cyberlogic Serial MBX Driver is already running! Server start operation has been aborted.

The driver could not start because another copy of it is already running.

Cyberlogic Serial MBX Driver (<Version Number>) started.

The driver successfully started. The driver's version number may be requested if you call Cyberlogic Tech Support.

Unable to initialize global system resources.

The driver was unable to allocate enough memory to start. Close other open applications or add more memory to the system, and then try to restart the driver.

Invalid or missing configuration parameters for Serial MBX device <MBX Device Number>. Re-configure the device.

The driver was unable to create the Serial MBX device because of problems reading all the necessary configuration parameters. Use the MBX Driver Configuration editor to re-save the device settings, or, if that doesn't work, to delete and re-create the device. Restart the driver to use the new settings.

Unable to open COM<COM Port Number> for Serial MBX device <MBX Device Number>. Port may be in use.

The driver will be unable to communicate on the specified COM port because it is in use by another application. Stop the application using the COM port or use the MBX Driver Configuration editor to select a different COM port. Restart the driver to use the new settings.

Serial MBX device <MBX Device Number>: Modbus Plus address is already mapped.

The driver detected that the serial device's active node mapping contained a duplicate assignment. The first assignment will be used.

Cyberlogic Serial MBX Driver service stopped because there are no Serial MBX devices configured.

The driver shut down because there were no devices configured. To run the driver, create at least one Serial MBX device in the MBX Driver Configuration editor and restart the driver.

Registration DLL failed to load. The I/O operations of the Serial MBX Driver have been disabled. Reinstall the product.

A necessary registration DLL could not be loaded. This may indicate a corrupted installation. Repair the existing installation, or remove and reinstall the software.

Product license verification failed. The I/O operations of the Serial MBX Driver have been disabled. Reinstall the product.

A registration check indicated that the software's evaluation time has expired. Run the Activation Wizard to authorize further use of the software.

This is a <Number of Hours>-hour promotional copy of the Serial MBX Driver. The application started at <Start Time> and the driver will stop at <Stop Time>.

This is a time-limited installation of the software. After the stop time, the driver will not allow any further I/O operations.

This is a promotional copy of the Serial MBX Driver. The allowed operation time has expired. The I/O operations of the Serial MBX Driver have been disabled.

This is a time-limited installation of the software. The stop time has been reached or exceeded, so the driver will not allow any further I/O operations.

The Cyberlogic License Server failed to respond with valid license information. The I/O operations of the Serial MBX Driver have been disabled. Contact the manufacturer's technical support.

The driver experienced a problem when it tried to contact the Cyberlogic License Server. If the license server is not running, start it and then try restarting the driver. If the license server is already running, contact Cyberlogic Tech Support.

Frequently Asked Questions

I've installed the software. What's next?

The next step is to configure a logical device (Serial MBX Master or Serial MBX Slave). Refer to the Quick-Start Guide for detailed instructions. After this is done, run the MBX Demo to test the driver.

I've configured a Serial MBX device. How do I know that it's working?

To test the Serial MBX Driver, there are two options in the Validation & Troubleshooting section. First use the MBX Demo to confirm that the device is operating properly and then use the Performance Monitor as a benchmark reference.

I have configured my Serial MBX device. However, when I try to do any Peer Cop related I/O requests, I get an error. What's the problem?

Serial Modbus communications do not support Peer Cop.

In MBX Demo, when I select Read Device Status or Device Information, I get an error that says "The system cannot find the file specified (Error code 1806)."

Cause 1

Verify that at least one Serial MBX device has been configured. If not, refer to the Quick-Start Guide for details on setting up a Serial MBX device.

Cause 2

The Serial MBX Driver could not find the MBX device specified under Device Number. Refer to the Quick-Start Guide for details on finding and entering this information.

I have two Serial MBX devices configured in the system. How do I communicate through the second one?

MBX Demo uses the device number to determine which MBX device to use. _[1] Set Device Number_ lets you choose which configured Serial MBX device the demo will use. If you are using some other software product, contact the manufacturer for more information on using multiple devices.

APPENDIX: MBX ARCHITECTURE AND COMPANION PRODUCTS

The Serial MBX Driver is part of the Cyberlogic MBX family. This family consists of several well-integrated components that provide connectivity for Modicon's Modbus, Modbus Plus and Modbus TCP (Ethernet) networks in distributed environments.

This section illustrates the layout of the MBX architecture. It includes a description of each MBX component along with suggested methods for employing them to support Modicon networks.
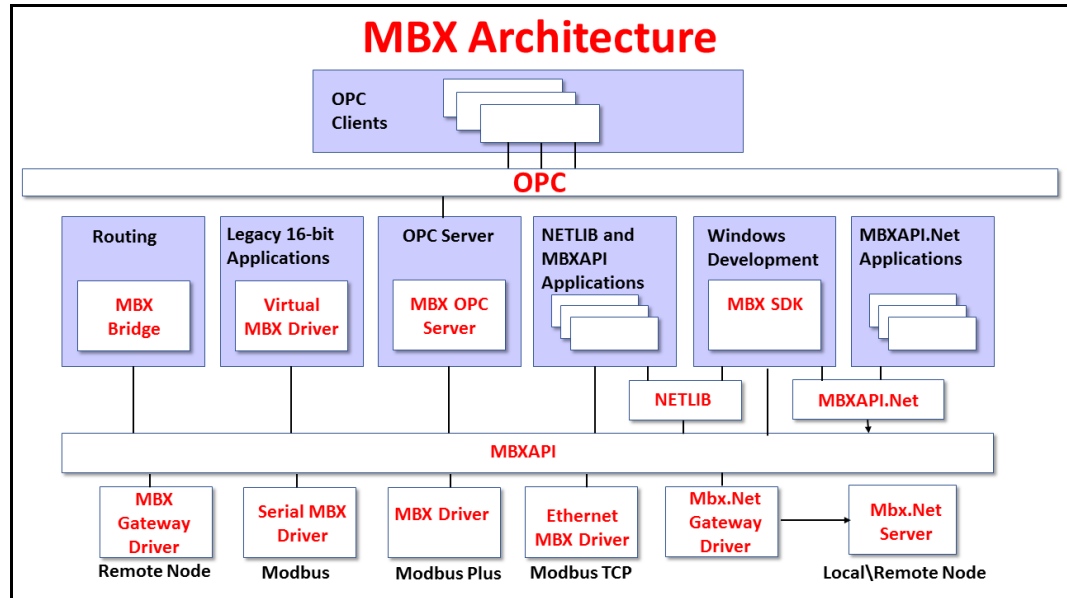
The MBX architecture presents a consistent framework
to address different connectivity needs.

MBX Driver

The MBX Driver provides connectivity between Modbus Plus interface adapters and Windows-based applications. It supports all Modbus Plus interface adapters for PCI Express (PCIe), PCI, USB and PCMCIA buses that are compatible with the supported operating systems. For a complete list of supported adapters, refer to the MBX Driver help file. Multiple interface cards can be installed at the same time, limited only by the number of available slots.

The kernel mode device driver of the MBX Driver is the highest-performance Modbus Plus driver in the industry. The driver operates in either interrupt or polled mode and fully implements all Modbus Plus features, providing support for Data Master/Slave, Program Master/Slave, Global Data and Peer Cop. The high-performance native API (MBXAPI) of the MBX Driver takes advantage of the event-driven, multitasking, multithreaded features of Windows operating systems.

The driver includes the MBX Gateway Server for remote access by the MBX Gateway Driver and is fully compatible with all other components of the MBX family.

The MBX Driver is included in the following products:

- MBX OPC Enterprise Suite
- MBX OPC Premier Suite
- MBX OPC Server Suite
- MBX Bridge Suite
- MBX Driver Suite

Ethernet MBX Driver

The Cyberlogic Ethernet MBX Driver emulates Modbus Plus over the Modbus TCP protocol. This allows most Modbus Plus-compatible software to gain instant access to Modbus TCP-enabled devices without code modifications. It is compatible with all Ethernet cards supported by Windows.

The driver includes the MBX Gateway Server for remote access by the MBX Gateway Driver and is fully compatible with all other components of the MBX family.

The Ethernet MBX Driver is included in the following products:

- MBX OPC Enterprise Suite
- MBX OPC Premier Suite
- MBX OPC Server Suite
- MBX Bridge Suite
- MBX Driver Suite

Serial MBX Driver

The Serial MBX Driver provides connectivity to Modbus-compatible devices through standard serial COM ports. It supports both master and slave node communications for Modbus ASCII and Modbus RTU protocols.

The driver includes the MBX Gateway Server for remote access by the MBX Gateway Driver and is fully compatible with all other components of the MBX family.

The Serial MBX Driver is included in the following products:

- MBX OPC Enterprise Suite
- MBX OPC Premier Suite
- MBX OPC Server Suite
- MBX Bridge Suite
- MBX Driver Suite (Some OEM versions do not include the Serial MBX Driver.)

MBX Gateway Driver

The MBX Gateway Driver lets applications use MBX devices on remote MBX Gateway Server nodes as though they were on the local system. The client system running the MBX Gateway Driver must be a Windows node connected over a standard LAN to another system running the MBX Gateway Server. It can then access the Modbus, Modbus Plus and Modbus TCP networks that are connected to the server node.

For example, the MBX Gateway Driver provides complete MBX Driver functionality to the client node applications, including support for Data Master/Slave, Program Master/Slave, Global Data and Peer Cop. An interface adapter, such as a PCIe-85 card, is not required on the client node. MBX Gateway Driver nodes can communicate with multiple remote servers and all Windows-compatible TCP/IP networks are supported.

The MBX Gateway Driver is compatible with all other components of the MBX family.

The MBX Gateway Driver is included in the following products:

- MBX OPC Enterprise Suite
- MBX OPC Premier Suite
- MBX OPC Server Suite
- MBX Bridge Suite
- MBX Driver Suite

Mbx.Net Gateway Driver

The Cyberlogic Mbx.Net Gateway Driver provides similar functionality to the Cyberlogic MBX Gateway Driver. However, the Mbx.Net Gateway Driver uses newer and more secure infrastructure, and allows for more communication protocols.

The driver connects existing applications that use the MBXAPI, MBXAPI.Net or NETLIB to the Mbx.Net Server. The server can be on the local network or anywhere on the internet. The Mbx.Net Gateway node provides access to all MBX devices as though they were on the local system. The connection can be through Http, Net.Tcp or Net.Pipe protocols. A secure connection can be established and protected by a user name/password.

The Mbx.Net Gateway Driver is compatible with all other components of the MBX family.

The Mbx.Net Gateway Driver is included in the following products:

- Mbx.Net Suite
- Mbx.Net Premier Suite

Virtual MBX Driver

The Virtual MBX Driver enables 16-bit NETLIB/NetBIOS-compatible applications, such as Modsoft and Concept, to run concurrently with 32-bit applications on the same computer.

It allows multiple 16-bit applications and multiple instances of a single 16-bit application to run under the latest 32-bit Windows operating systems.

If your computer uses a 64-bit edition of Windows, refer to Cyberlogic Knowledge Base article *KB2010-02 Running 16-Bit Applications* for important information on using the Virtual MBX Driver on your system.

The Virtual MBX Driver is fully compatible with all MBX components and requires at least one of these drivers to operate:

- MBX Driver
- Ethernet MBX Driver
- Serial MBX Driver
- MBX Gateway Driver
- Mbx.Net Gateway Driver

The Virtual MBX Driver is included in the following products:

- MBX OPC Enterprise Suite
- MBX OPC Premier Suite
- MBX OPC Server Suite
- MBX Bridge Suite
- MBX Driver Suite

MBX Bridge

The MBX Bridge seamlessly routes messages between MBX-compatible devices. For example, the MBX Bridge can route messages between Ethernet and Modbus Plus networks, between Modbus and Modbus Plus networks or any other combination of the supported networks.

Depending on the user's needs, it requires one or more of the following drivers to operate:

- MBX Driver
- Ethernet MBX Driver
- Serial MBX Driver
- MBX Gateway Driver
- Mbx.Net Gateway Driver

The MBX Bridge is included in the MBX Bridge Suite.

MBX OPC Server

The Cyberlogic MBX OPC Server connects OPC-compliant client applications to Modbus, Modbus Plus and Modbus TCP networks. It supports the latest OPC Data Access and OPC

Alarms and Events specifications and uses the MBX drivers for connectivity to Modicon networks.

The MBX OPC Server supports multiple, priority-based access paths for reliable, redundant communications. It also supports both solicited and unsolicited communications and uses an advanced transaction optimizer to guarantee minimum load on your networks. With only a couple of mouse clicks, the MBX OPC Server will automatically detect and configure the attached networks and node devices. Other noteworthy features include DirectAccess, Data Write Protection and Health Watchdog.

The MBX OPC Server is included in the following products:

- MBX OPC Enterprise Suite
- MBX OPC Premier Suite
- MBX OPC Server Suite

MbpStat.Net

The MbpStat.Net provides the same functionality as the original DOS-based MBPSTAT application. It can monitor local or remote Modbus Plus by connecting to systems running the Mbx.Net Server. It can run on 32-bit and 64-bit Windows, and it does not require the Virtual MBX Driver.

The MbpStat.Net is included in the following products:

- Mbx.Net Suite
- Mbx.Net Premier Suite

Mbx.Net Server

The Cyberlogic Mbx.Net Server connects applications that use the MBXAPI, MBXAPI.Net or NETLIB to Modbus, Modbus Plus and Modbus TCP networks. The client applications and server can be on the local network or connected through the Internet.

The server is a Windows Communication Foundation (WCF) server that supports Http, Net.Tcp and Net.pipe protocols. It allows connections to be secured with a user name/password. When used with an application like MbpStat.Net it allows remote monitoring of networks and devices. Existing applications can use the server along with the Mbx.Net Gateway Driver to securely connect to remote devices.

The Mbx.Net Server is included in the following products:

- Mbx.Net Suite
- Mbx.Net Premier Suite

MBX SDK

Software developers can use the MBX Software Development Kit to provide connectivity to Modbus, Modbus Plus and Modbus TCP networks from their 32-bit and 64-bit C/C++/C# applications.

The SDK supports NETLIB, and Cyberlogic's high-performance MBXAPI and MBXAPI.Net interfaces. NETLIB is an excellent bridge for developers who would like to port their 16-bit applications to the latest Windows environments. Developers of new applications can use any of the three interfaces. For a complete reference of all NETLIB library functions, refer to *Modicon IBM Host Based Devices User's Guide*, available from Schneider Electric (Order #890 USE 102 00).

Since all MBX family drivers are built on the same MBX architecture, applications developed with the MBX SDK can be used with all MBX family drivers and can execute under all current Windows operating systems.