ETHERNET MBX DRIVER

Ethernet MBX® Driver for Modicon® TCP/IP Networks

Version 6.0 for Windows® XP/2000/NT/Server 2003

Document last revision date March 6, 2006

TABLE OF CONTENTS

INTRODUCTION

The Ethernet MBX Driver (eMBX) emulates Modbus Plus over TCP/IP. It is implemented as part of the Cyberlogic MBX architecture. The MBX architecture is a foundation used in other Cyberlogic products such as the MBX Driver, the Serial MBX Driver and the MBX Gateway Driver. Consequently, these products consistently support identical programming interfaces: the MBXAPI and Modicon's industry-standard NETLIB. As a result, virtually all Modbus Plus compatible software products can gain instant access to TCP/IP-based communications with no code modifications. This includes both 32-bit Windows XP/2000/NT and 16-bit legacy DOS/Windows applications.

Supporting these existing standards protects the software and R&D investments of end-users and OEMs. However, the Ethernet MBX Driver does not sacrifice or eliminate any functionality available in direct communications through the Winsock API. On the contrary, Ethernet MBX solves certain critical problems that the Winsock interface creates. For example, since only one process can listen on the TCP port 502 for incoming messages, a conflict will arise if more than one application tries to receive unsolicited messages. Ethernet MBX eliminates this problem by serving as a global dispatcher for these types of messages. By implementing the Slave Path concept found in Modbus Plus, up to 256 applications can subscribe to unsolicited messages and process them simultaneously.

Benefits

The main advantages of using the Ethernet MBX Driver over the Winsock API include:

- No required changes to existing NETLIB/NetBIOS/MBXAPI compatible applications. The software investments of end users and developers are fully protected.
- Consistent handling and dispatching of unsolicited messages that eliminate resource contention among different products running on the same system.
- Full TCP/IP communications functionality while protecting existing NETLIB/NetBIOS/MBXAPI standards. For example, Ethernet TCP/IP communications require a destination address in the form of an IP address and a message including a destination index byte. Ethernet MBX preserves this functionality.
- A benefit for software developers of TCP/IP communications, who need not be familiar with the complicated Winsock API.
- A single programming model for software developers implementing communications over all Modicon supported networks: Modbus, Modbus Plus and Ethernet TCP/IP.
- Less effort implementing TCP/IP message handling and unusual exceptions.
- More compatibility with different products. Winsock API-focused developers implementing TCP/IP-related strategies in a slightly different manner may create compatibility problems in different products.
- Compatibility with all MBX family companion products, such as the Virtual MBX Driver for 16-bit legacy DOS/Windows applications and the MBX Bridge, which routes messages between Ethernet, Modbus, Modbus Plus and MBX Gateway nodes.

Compatibility

The Ethernet MBX Driver is compatible with applications supporting the high-performance MBXAPI application programming interface and the industry-standard NETLIB interface specification from

Modicon. Supporting these existing standards protects the software and R&D investments of end-users and OEMs.

The 32-bit NETLIB compatibility provides an excellent bridge for developers who would like to port their 16-bit, NETLIB-compatible applications to 32-bit Windows operating systems (Windows XP/2000/NT). Application developers can use either NETLIB or the high-performance MBXAPI programming interface. To obtain the MBX Software Development Kit (including the MBXAPI specification, MBXAPI sample source code and NETLIB sample source code), contact your Schneider Automation, Inc. Modicon-brand distributor. For a complete reference of all NETLIB library functions, refer to "Modicon IBM Host Based Devices User's Guide" from Schneider Automation (Order #890 USE 102 00).

Remote Connectivity

The Ethernet MBX Driver includes the MBX Gateway server. When enabled, the server allows access to all local MBX devices from remote client nodes over any Windows XP/2000/NT-compatible network. The remote client can be a Windows XP/2000/NT node, running the MBX Gateway Driver product. The MBX Gateway Driver provides complete Ethernet MBX Driver functionality to the client node, including support for Data Master/Slave and Program Master/Slave. Any node on the network can be configured as a client to a number of Gateway servers while communicating over its local MBX devices.

Running 16-Bit Software

A companion product, the Virtual MBX Driver, 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 on this product, refer to the MBX Architecture and Companion Products section.

What Should I Do Next?

The Cyberlogic MBX family for Windows XP/2000/NT consists of several well-integrated products that provide connectivity for Modicon Modbus, Modbus Plus and Ethernet networks in distributed environments. For more information about these products, refer to the MBX Architecture and Companion Products section.

For architectural and implementation details of the Ethernet MBX Driver product, read the Theory of Operation section. This section compares Cyberlogic's Ethernet MBX Modbus Plus emulation architecture to direct Ethernet Modbus communications. It shows the architectural differences between these types of communications and explains how the Ethernet MBX Driver solves these architectural differences.

After installation, the Ethernet MBX Driver must be configured. You will find information on this topic in the Configuration section.

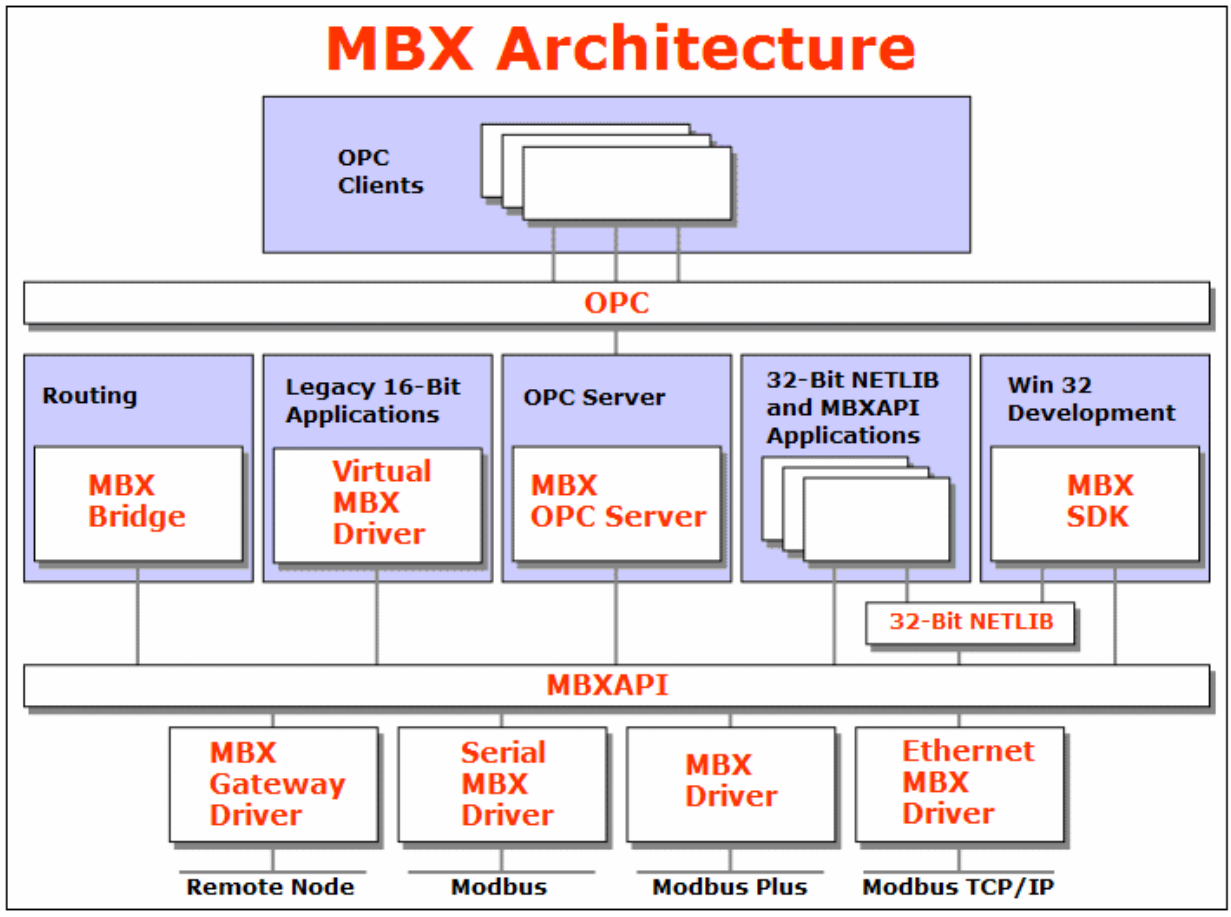
If you have already configured the driver, verify that it operates as expected. Refer to the Validation & Troubleshooting section for assistance. In case of communication problems, this section also provides problem-solving hints.

The content of this document is also provided in the PDF file format. PDF files can be viewed using the Adobe® Reader program. The printer-friendly PDF files can be used to print the complete document with good quality output.

MBX ARCHITECTURE AND COMPANION PRODUCTS

This section illustrates the layout of the MBX architecture. It includes a description of each MBX product along with suggested methods for employing these products to support Modicon networks.

The Cyberlogic MBX family for Windows XP/2000/NT consists of several well-integrated products that provide connectivity for Modicon's Modbus, Modbus Plus and Modbus TCP/IP (Ethernet) networks in distributed environments.



The MBX architecture presents a consistent framework to address different connectivity needs.

Software products available in the MBX family are:

MBX Driver: This is Cyberlogic's device driver for Modbus Plus host interface adapters. The MBX Gateway Server is included for remote connectivity.

Ethernet MBX Driver: This driver provides Modbus Plus emulation over TCP/IP. The MBX Gateway Server is included for remote connectivity.

Serial MBX Driver: This driver provides Modbus Plus emulation over serial Modbus. The MBX Gateway Server is included for remote connectivity.

MBX Gateway Driver: This product provides access to Modicon's Modbus, Modbus Plus and Modbus TCP/IP networks from remote locations.

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

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

MBX OPC Server: Cyberlogic's premium OPC Server connects OPC compliant client software applications to data sources over all Modicon networks.

MBX SDK: This is a software development kit for MBXAPI and NETLIB compliant development.

MBX Driver

The 32-bit MBX Driver provides connectivity between Modicon ModConnect host interface adapters and 32-bit applications running under Windows XP/2000/NT.

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 supports all current Modicon ModConnect host interface adapters for ISA, EISA, MCA, PCI and PC Card (PCMCIA) buses. Multiple interface cards can be installed at the same time, limited only by the number of available slots. Full implementation of all Modbus Plus features provides 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 32-bit operating systems.

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

Ethernet MBX Driver

The 32-bit Ethernet MBX Driver provides connectivity between Modbus TCP/IP compatible processors and Windows XP/2000/NT based 32-bit applications using either Modicon NETLIB or Cyberlogic's high-performance MBXAPI interface specification. It provides Data Master/Slave and Program Master/Slave features of Modbus Plus on Ethernet networks.

The driver includes the MBX Gateway Server for remote access by the MBX Gateway Driver and is fully compatible with all other MBX family products. The Ethernet MBX Driver does not require a special Ethernet adapter. It is compatible with all Ethernet cards supported by Windows.

Serial MBX Driver

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

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

MBX Gateway Driver

The MBX Gateway Driver lets you access Modbus, Modbus Plus and Modbus TCP/IP networks from a remote location. Through a standard LAN, your local applications can use MBX devices on remote server nodes as though they were on your local system.

The remote client running the MBX Gateway Driver must be a Windows XP/2000/NT node. By accessing the Modbus, Modbus Plus and Ethernet networks connected to server nodes on a network, the MBX Gateway Driver provides complete MBX Driver functionality to the client node, including support for Data Master/Slave, Program Master/Slave, Global Data and Peer Cop. A host interface adapter, such as a Modicon SA85 card, is not required on the client node. MBX Gateway Driver nodes can communicate with multiple remote servers and all Windows XP/2000/NT-compatible computer networks are supported.

The MBX Gateway Driver is compatible with all other MBX family products.

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 32-bit Windows operating systems.

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 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 products to operate:

- MBX Driver
- Ethernet MBX Driver
- Serial MBX Driver
- MBX Gateway Driver

MBX OPC Server

The Cyberlogic MBX OPC Server connects OPC-compliant clients to Modicon Modbus, Modbus Plus and Ethernet 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 in seconds. Other noteworthy features include DirectAccess, Data Write Protection and Health Watchdog.

MBX SDK

Software developers can use the MBX SDK to provide connectivity to Modbus, Modbus Plus and Ethernet networks from their 32-bit C/C++ applications.

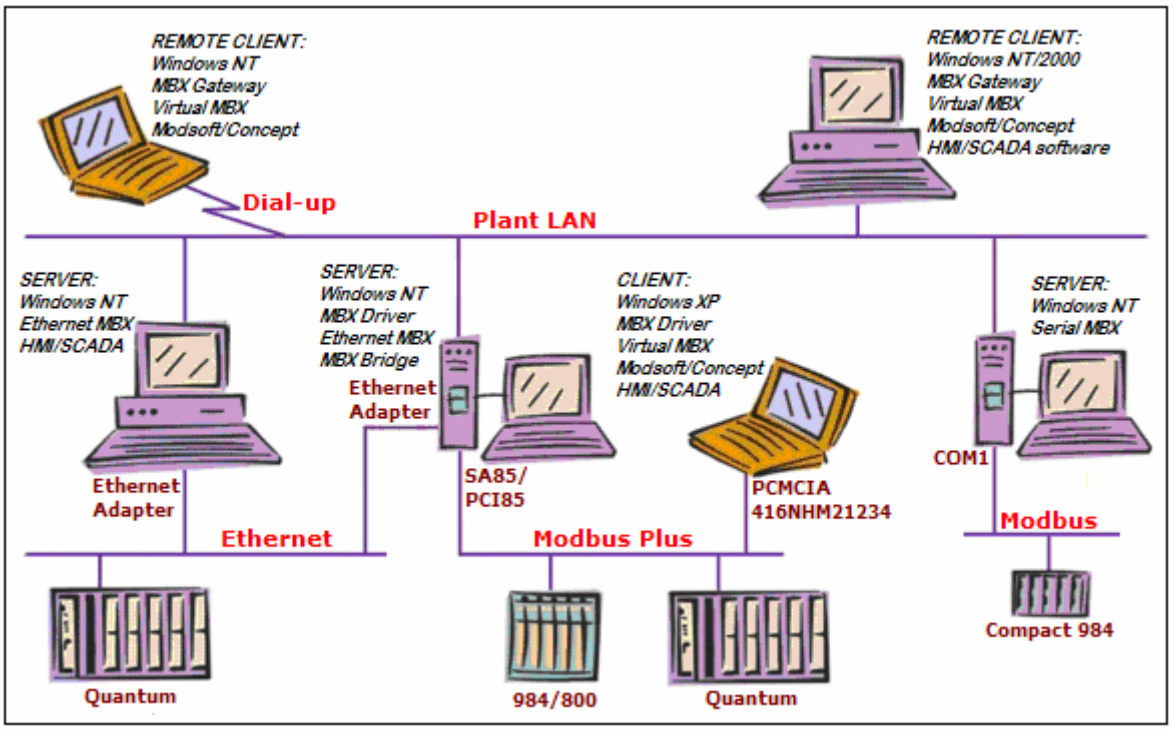
The SDK supports two styles of interfaces, the industry-standard NETLIB and Cyberlogic's high-performance MBXAPI. The NETLIB interface is an excellent bridge for developers who would like to port their 16-bit applications to the 32-bit Windows environments. Developers of new applications can use either the NETLIB or the MBXAPI interface.

Since all MBX driver products are built on the same MBX architecture, applications developed with the MBX SDK can be used with all MBX drivers and can execute under all 32-bit Windows operating systems.

Blending MBX Supported Networks

The MBX driver products provide support for all Modicon networks through a common architecture, with identical programming interfaces: the MBXAPI and the industry-standard NETLIB. This ensures that virtually all of the existing Modbus Plus compatible software programs can operate over all Modicon supported networks with no code modifications. A product operating with one of the MBX driver products, such as the MBX Driver, will operate with the rest of the MBX driver products as well.

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



MBX enabled system deployment:
New hardware solutions will blend into existing installations without software or network modifications

THEORY OF OPERATION

This section is intended to familiarize you with the main features of the Ethernet MBX Driver. Although the Ethernet MBX Driver only supports Ethernet, we have also included an overview of all Modicon networks.

Schneider Automation, Inc. provides multiple network solutions that allow communications to a variety of its own – as well as third-party – hardware products. The main communication networks include:

- Modbus
- Modbus Plus
- Ethernet

Modbus

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. Its serial communications are relatively slow. The message structure supports only a single byte destination node addressing. Because of its limitations, serial Modbus communications are primarily used in old legacy installations.

Modbus Plus

Modbus Plus is a 1 Mbit/sec peer-to-peer communication network. Its architecture supports both solicited (Master Path) and unsolicited (Slave Path) communications. It also supports Global Data and Peer Cop communications. The message structure used by Modbus Plus is identical to the Modbus message structure with the exception of the destination node address. Modbus Plus uses a 5-byte routing path to identify the destination node versus the 1-byte destination node addressing of Modbus. Also, a local network is limited to 64 nodes. Modbus Plus is the most prevalent communication network of the three Modicon networks and, as a result, has the best support in third-party automation software products. Most of these products communicate through the NETLIB library, which is well-supported on both 16-bit (DOS/Windows) and 32-bit platforms (Windows XP/2000/NT).

Ethernet

Ethernet is the most recent Modicon network supported by Schneider Automation products such as Quantum PLCs. The communication protocol is based on the standard TCP/IP protocol with Modbus messages embedded in the standard PDU messages. Ethernet operates at 10 Mbits/sec and supports both solicited and unsolicited communications. The message structure used by Ethernet communications is almost identical to the Modbus message structure with the exception of the destination node address, which is a standard IP address.

The MBX Driver products provide consistent support for all of the above networks through an identical software architecture.

Winsock API

Schneider Automation offers an SDK product that allows software developers to provide TCP/IP communications support through the use of Microsoft's Winsock. However, the direct Winsock API communication approach may raise a number of potential problems:

Since all Ethernet applications are forced to listen to communication messages over the same port (502), this approach is prone to resource contention among different software products running in a multitasking operating system. Therefore, only one application in the system can receive unsolicited messages.

The use of Winsock API requires modification and/or redesign of existing applications. For a number of legacy products, this single requirement is prohibitively expensive.

Software developers must be familiar with the Modbus message structure as well as the Winsock architecture and programming API.

The Ethernet MBX Driver eliminates these constraints by abstracting TCP/IP communications and exposing the communication architecture common to all MBX products. These are the major benefits of Ethernet MBX:

Modbus Plus has the most feature-rich architecture among all Modicon networks and is supported by virtually all industrial automation software. The MBX architecture has therefore been modeled on the Modbus Plus architecture.

For software developers, it is easier to support a single communication architecture rather than multiple architectures. The Ethernet MBX Driver exposes an interface for applications identical to other MBX products. As a result, a product compatible with one MBX driver product will operate with the Ethernet MBX Driver as well.

Since the MBX architecture emulates Modbus Plus over Ethernet TCP/IP communications, all existing Modbus Plus compatible automation software can gain instant access to Ethernet. This includes both 16-bit (DOS/Windows) and 32-bit (Windows XP/2000/NT) applications. No software changes are required since Ethernet communications are provided through existing standards, NETLIB and MBXAPI software libraries.

Issues Regarding Architectural Differences

The Quantum Ethernet TCP/IP protocol embeds Modbus messages inside the data fields of standard TCP or UDP packets. Although this protocol is similar to the Modbus Plus protocol, a number of architectural differences exist. The Ethernet MBX Driver resolves these differences.

Destination Index Byte

In addition to a regular Modbus message, an Ethernet message includes an additional byte called a destination index. This byte has no predefined purpose and its value can be programmed through the MSTR function in the PLC logic. The Ethernet MBX Driver uses this byte's value to map into either a Data Slave (DS) or a Program Slave (PS) path number. The type of path is determined by the Modbus command embedded in the message.

Solicited Communications

Problem

Solicited messages used by Ethernet communications require an IP address to specify its destination node. Modbus Plus, on the other hand, uses an array of five destination routing addresses.

Solution

To accommodate Modbus Plus architecture, the Ethernet MBX Driver allows two alternative methods that applications can use to map into an IP address:

Explicitly specifying an IP address

Specifying an address by indirect mapping

Explicit IP Address

In this method, an application can use the array of five routing addresses to explicitly specify an IP address as follows:

Byte 0	Send as destination index byte (0-255 max) – 0 treated as 256
Byte 1	IP Address High byte (bits 24 - 31)
Byte 2	IP Address (bits 16 - 23)
Byte 3	IP Address (bits 8 - 15)
Byte 4	IP Address (bits 0 - 7)

The value in the first routing byte is transmitted as a destination index code. This value is useful only when communicating to another Ethernet MBX server. Currently, Quantum PLCs ignore this value.

Explicit IP addressing requires no configuration and is the most flexible addressing method. Some applications, however, do not allow routing addresses to fall outside the 1-64 range. For these types of applications, indirect address mapping should be used.

Indirect Address Mapping

In this method, an application uses only the first byte in the array of five routing addresses:

Byte 0	Mapping table entry (1 - 256)
Byte 1	0
Byte 2	0
Byte 3	0
Byte 4	0

The Ethernet MBX Configuration Editor allows the creation of logical devices (virtual host interface adapters) and the configuration of a mapping table for each device. The Ethernet MBX server takes the value from byte 0 and maps it to an entry in the mapping table. Mapping table entries can specify either an IP address or a host name that the server converts to an IP address.

Since each logical device allows a maximum of 256 mapping entries, more devices can be created as more entries are needed. Note that some applications may limit the value in the first routing byte to a range of 1-64.

Problem

The local Modbus Plus network is limited to a maximum of 64 nodes. An Ethernet network does not have this limitation since it uses a global IP address.

Solution

The Ethernet MBX Configuration Editor allows the creation of logical devices (virtual host interface adapters) and the configuration of a mapping table for each device. Each table entry corresponds to a local Modbus Plus node address. Mapping table entries can specify either an IP address or a host name that the server converts to an IP address. If all mapping entries are used, more devices can be created. In addition, Ethernet MBX supports the direct use of an IP address and the destination index value inside the Modbus Plus routing array. This is the most flexible addressing method and requires no configuration.

Problem

Modbus Plus applications select a host interface adapter prior to communicating to a destination device. Winsock, on the other hand, automatically selects the network card based on the IP address. Applications do not have control over which network card is used for communications.

Solution

The Ethernet MBX Configuration Editor allows the creation of logical devices (virtual host interface adapters). From the application's perspective, this emulates a physical host interface adapter. However, the physical network card being used depends strictly on the destination IP address and not on the logical device number.

Problem

Modbus Plus has a notion of Data/Program Master paths used as logical communication channels for solicited communications. Modbus protocol used by Ethernet communications has no concept of these communication paths.

Solution

For applications using the MBXAPI interface, Ethernet MBX can emulate up to 65,535 Data/Program Master paths per device handle (device connection). The number of open device handles is limited only by the available amount of memory and the maximum number of Winsock sockets in the system.

Since the NETLIB library interface uses both a separate device handle and a single Data/Program Master path per ncb_open function call, even for NETLIB-compliant applications, the number of concurrent communications is limited only by the available amount of memory and the maximum number of Winsock sockets in the system. By comparison, Modbus Plus supports a maximum of eight Data/Program Master paths.

Problem

Modbus Plus communications allow only one programming software (e.g., Modsoft) at a time to program a PLC. Since Ethernet communications do not have a Program Master path concept, no inherent arbitration is provided to guarantee that only one user at a time programs a PLC.

Solution

For Program Master paths, Ethernet MBX does not allow the user to concurrently reuse Winsock sockets. Since the same socket is tied to a PM path in use, a target Quantum PLC provides the rest of the necessary arbitration.

Unsolicited Communications

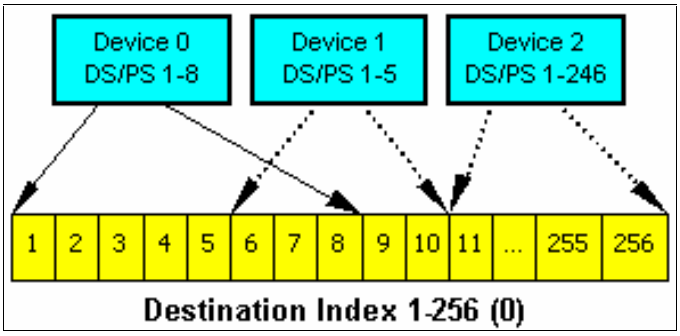
Problem

Modbus Plus applications select a host interface adapter through which they intend to receive unsolicited messages. Winsock, on the other hand, automatically selects the network card based on the IP address and all unsolicited messages are received through a single port (502).

Solution

The Ethernet MBX Configuration Editor allows the creation of logical devices (virtual host interface adapters). From the application's perspective this emulates a physical host interface adapter. This emulation is very accurate for solicited messages. However, unsolicited messages received through TCP/IP port 502 have no device designation. The only additional information provided in the message is the destination index byte. Ethernet MBX uses this value for mapping to a DS/PS path number. The type of path is determined by the Modbus command code embedded in the message. Data access messages are routed to DS paths while programming messages are routed to PS paths.

For each logical device, the configuration editor allows the user to select a range of destination index values that are mapped to DS/PS path numbers starting from path number 1. By default, a direct one-to-one mapping is used and a destination index value represents a DS/PS path number. This provides a maximum of 256 DS/PS paths. By comparison, Modbus Plus supports a maximum of eight DS/PS paths.



In the default configuration, the unsolicited messages can be received through any logical device. If broken up into ranges, each device can receive messages within a certain range of destination index values. Since the NETLIB library supports a maximum of 63 Slave Paths, multiple logical devices may be created to cover the entire 1-256 destination index range. When ranges overlap, an overlapped Slave Path can be opened through only one device at a time.

Problem

Modbus Plus has a notion of Data/Program Slave paths used as logical communication channels for unsolicited communications. Modbus protocol used by Ethernet communications has no concept of these

communication paths. Instead, any application that intends to receive unsolicited messages must listen to TCP/IP port 502.

Solution

Ethernet MBX listens to TCP/IP port 502 and internally emulates the Data/Program Slave paths based on the destination index value in the message. A total of 256 DS and 256 PS paths are supported. They can be accessed through any logical device. Refer to the previous section for more information.

Ethernet MBX provides buffering of incoming Slave Path messages. Only one command message at a time can be processed on a given Slave Path. A DS/PS path subscriber must respond to a message or cancel it before the next command message is forwarded. If the Slave Path subscription is cancelled (closed), any pending messages are NAKed.

Message handling differs slightly depending on the Slave Path type. DS paths allow messages from multiple nodes (multiple sockets). PS paths, on the other hand, accept messages from only one node at a time (one socket). A message-sending node must close its connection before another node is allowed to send messages over the same PS path. While one node has an active connection to a given PS path, messages received from other nodes designated to the same PS path are NAKed. This behavior guarantees that only one client at a time has programming privileges.

Problem

Modbus Plus command messages received through Data/Program Slave paths include an array of five routing addresses and an address of a source device on the local network. Modbus protocol used by Ethernet communications does not provide this type of information. Instead, it uses a 1-byte node address and a destination index byte.

Solution

Command messages received over *Slave Paths* provide the routing information of the source node as follows:

Byte 0	Destination index byte
Byte 1	Source IP Address (bits 24 - 31)
Byte 2	Source IP Address (bits 16 - 23)
Byte 3	Source IP Address (bits 8 - 15)
Byte 4	Source IP Address (bits 0 - 7)
Byte 5	Source node address

Ethernet MBX uses the mapping table associated with the logical device to map the source IP address into a source node address (Byte 5) on the local network. For this mapping to work correctly, a user should select a logical device that provides the required IP address mapping. Otherwise, the source address field will be set to 0. Most applications do not use the source address field.

Global Data Communications

Problem

Modbus Plus supports Global Data type communications. Ethernet does not support this type of communication.

Solution

Ethernet communications do not support Global Data. Therefore, all Global Data reads receive a *no data* reply. Similarly, all Global Data writes report *success*.

Ethernet to Ethernet/Modbus Plus Routing

Problem

Modbus Plus allows Modbus Plus to Modbus Plus routing through the use of a five routing address array. Modbus protocol used by Ethernet communications does not support this type of routing. Only global TCP/IP routing to other Ethernet devices is supported.

Solution

Cyberlogic provides a separate product called MBX Bridge that can be used for routing messages between all MBX compatible devices. Specifically, the MBX Bridge can be used to route messages between Ethernet-to-Modbus Plus or Modbus Plus-to-Ethernet networks.

CONFIGURATION

Before the Ethernet MBX Driver can be used, it must be properly configured. To accomplish this, you must run the MBX Driver Configuration Editor at least once after the installation.

The MBX Driver Configuration Editor is a common component of MBX products. When configuring an Ethernet MBX type device, the MBX Driver Configuration Editor automatically dispatches the Ethernet MBX Configuration Editor. Both editors are well-integrated, allowing for seamless editing.

The procedures you will use to configure the Ethernet MBX driver are broken down into several sections:

- Typical Driver Configuration is a good place to start if you are a first-time user. It is a tutorial that walks you through a complete driver configuration session. It also introduces some diagnostic tools used for validating and troubleshooting the driver.
- Creating an Ethernet MBX Device is a guide to creating a new MBX device if you are just getting started or need to add a new adapter card.
- Editing Device Configuration describes how to open an existing device for editing.
- MBX Driver Configuration Editor describes the configuration of the MBX Driver.
- Ethernet MBX Configuration Editor describes the configuration of an Ethernet MBX device.
- Configuration Backup/Restore shows how to backup and restore your configuration of MBX driver products.

Typical Driver Configuration

The following steps show a typical configuration session. Use it only as a guideline. Only the most common features will be shown here. For detailed descriptions, refer to the MBX Driver Configuration Editor and the Ethernet MBX Configuration Editor sections.

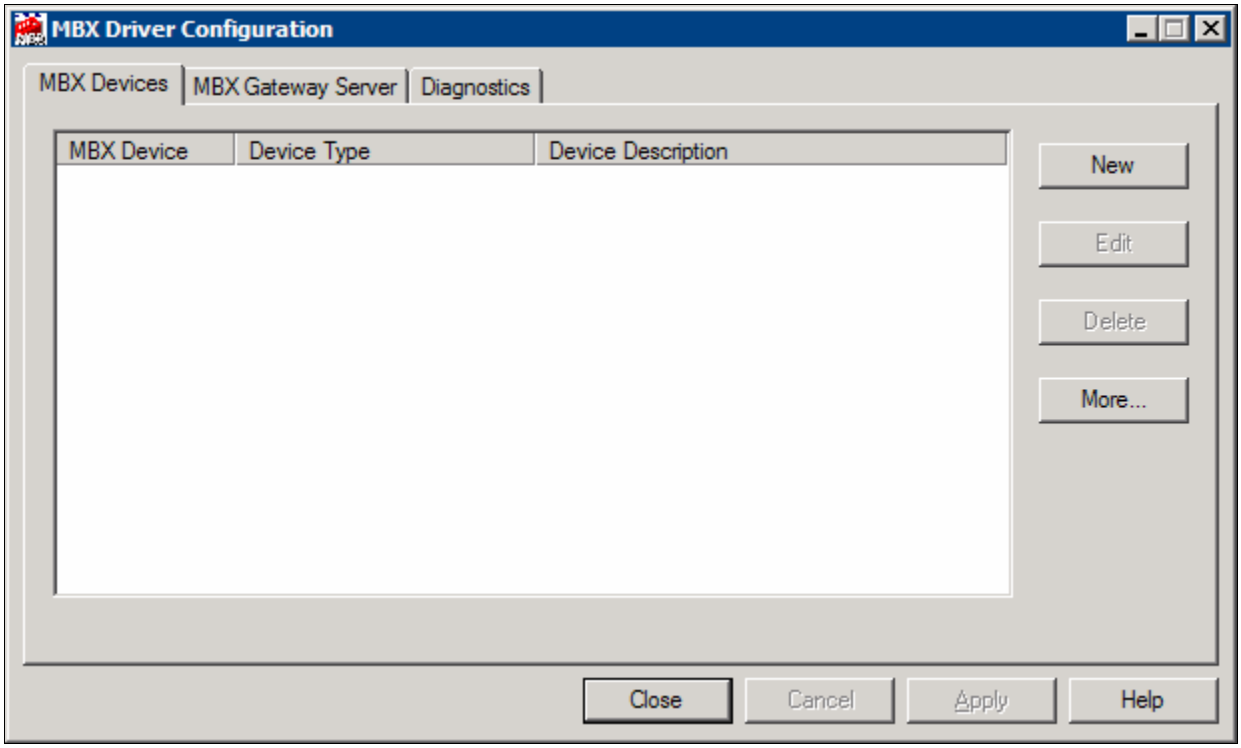
The procedure is broken into three segments. The first shows how to create and configure an Ethernet MBX device. The second covers configuration of the MBX Gateway server. The last segment introduces the diagnostic capabilities of the software.

To begin, go to Creating and Configuring an Ethernet MBX Device.

Creating and Configuring an Ethernet MBX Device

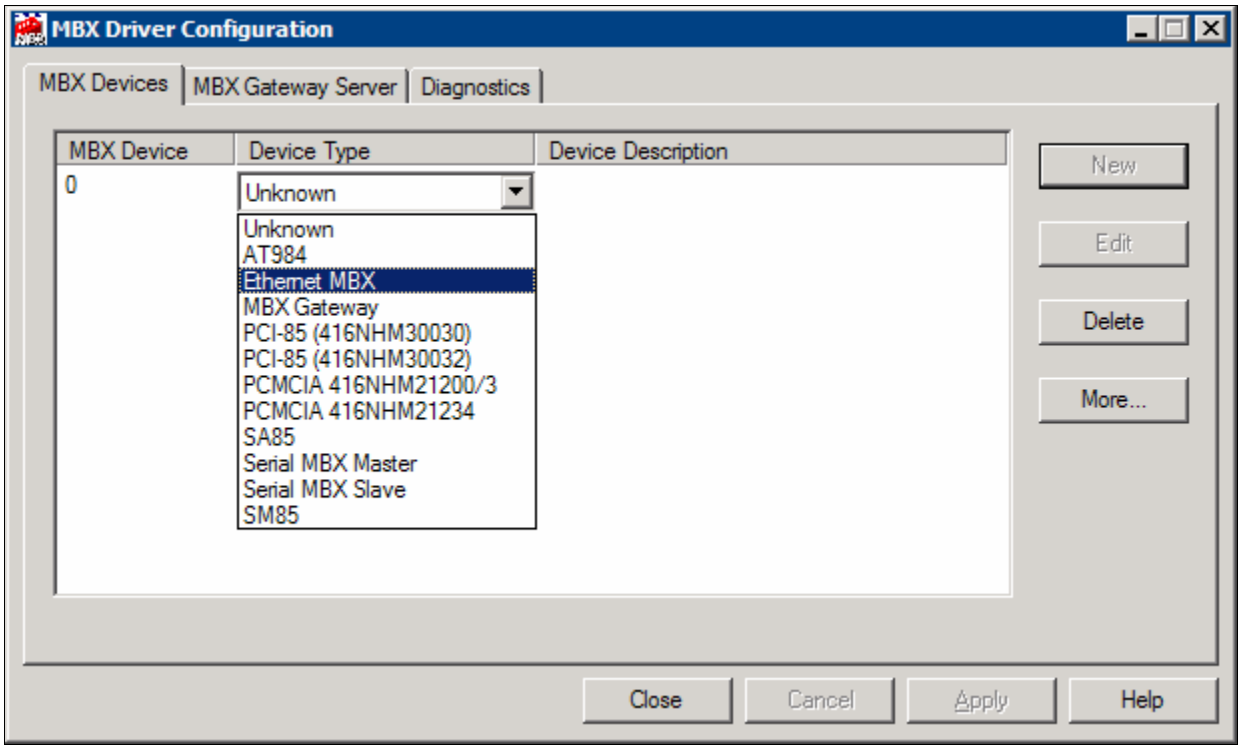
1. From the Windows Start menu, locate the MBX Ethernet Driver submenu and select the *MBX Driver Configuration* menu item.

   Running the editor for the first time displays the following screen:

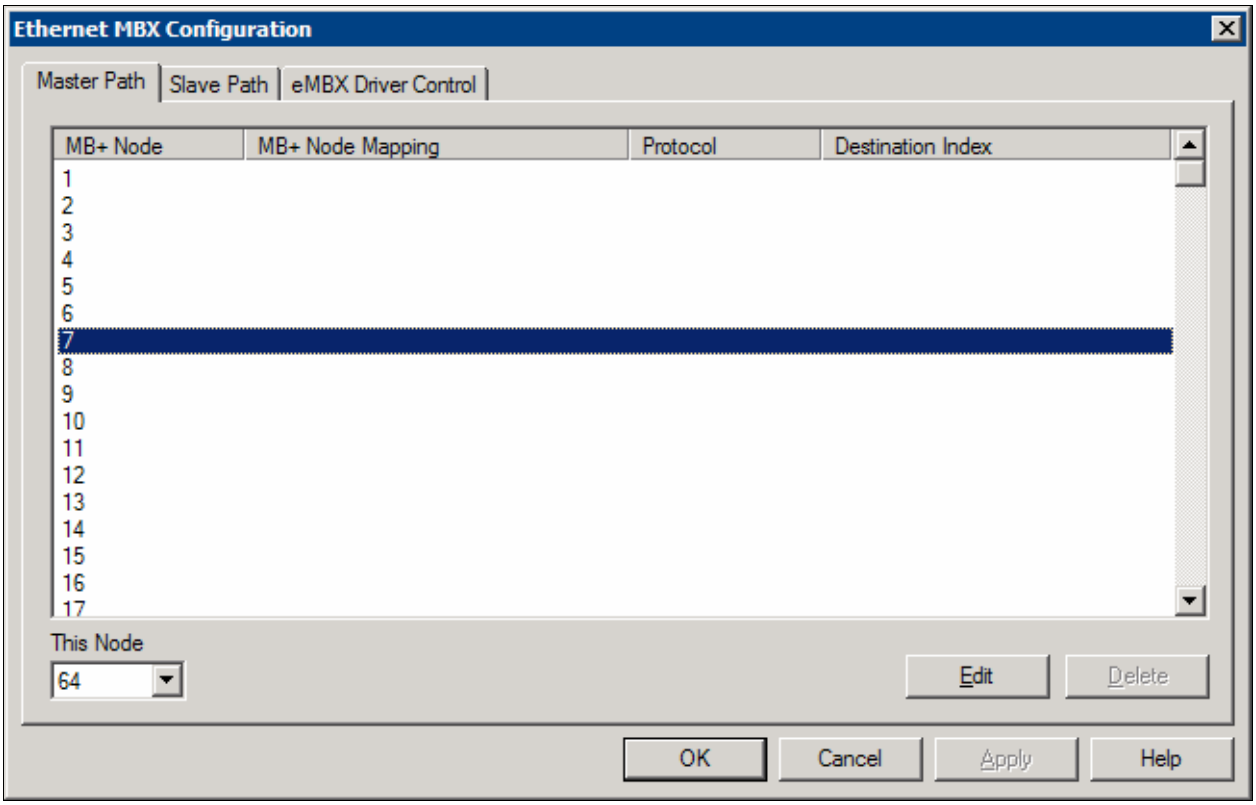


   The first step in configuring the Ethernet MBX Driver is to create at least one Ethernet MBX device. This is just a logical device that emulates a host interface adapter, such as an SA85 card, and has nothing to do with an actual, physical network card. Refer to the Theory of Operation section for more information. The physical network card used depends strictly on the destination IP address and not on the logical device number.

2. Click *New* and select *Ethernet MBX* from the drop-down list.



Upon selecting the *Ethernet MBX* device type, the MBX Driver Configuration Editor automatically dispatches the Ethernet MBX Configuration Editor. You will see the following screen:



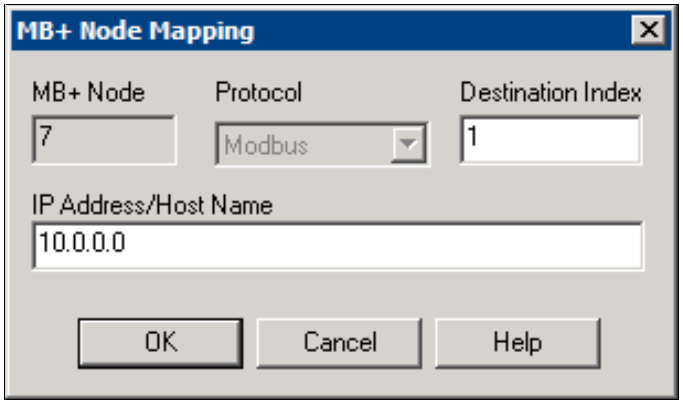
The Ethernet MBX Configuration Editor has three configuration tabs. By default, the *Master Path* tab is selected. This tab allows the user to configure all parameters related to the Data/Program Master Path operation.

3. Because the Ethernet MBX device emulates the behavior of a Modbus Plus adapter card, you must assign a Modbus Plus node address to this device by selecting the address from the *This Node* drop-down list.

4. If your application supports explicit IP addressing, as explained in the Theory of Operation section, you need not configure additional Master Path parameters. If this is the case, you can skip this step.

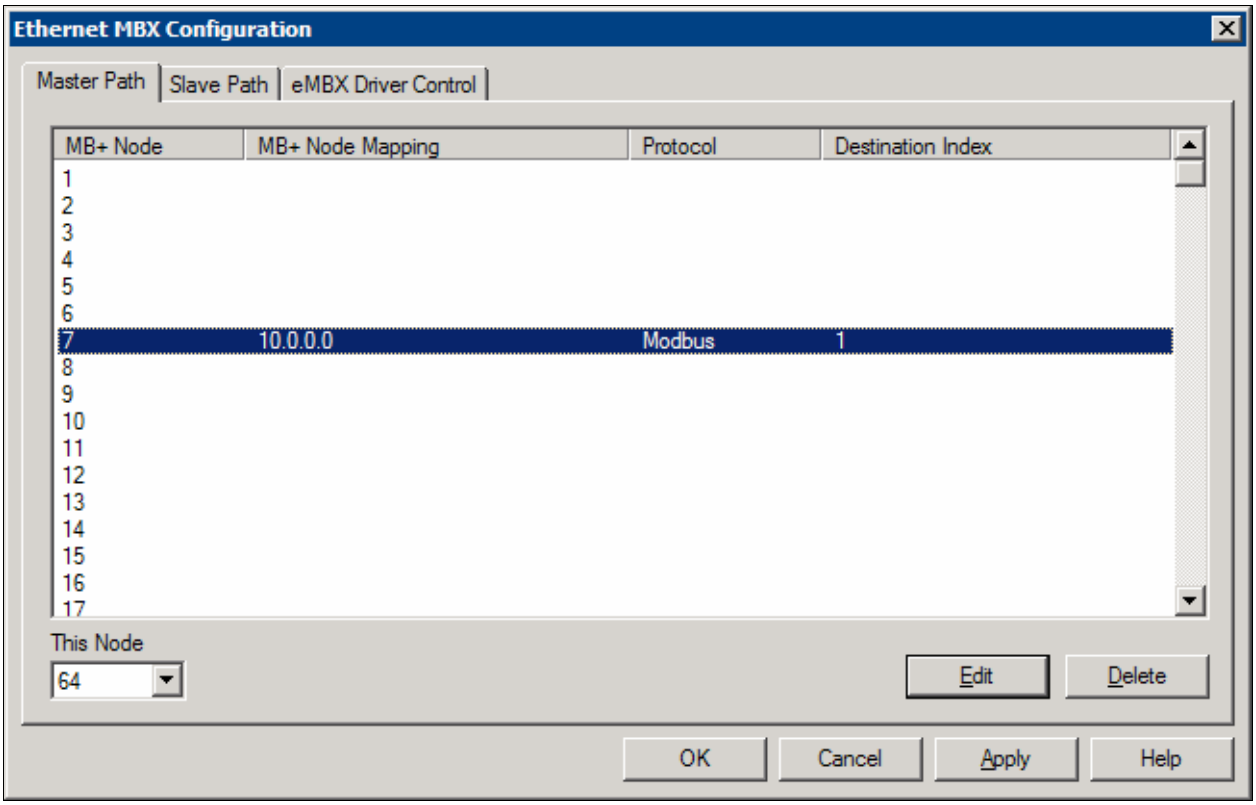
   For legacy applications not supporting the explicit IP addressing mode, you can use indirect address mapping. In this mode, Ethernet networks are viewed as a single local Modbus Plus network. For every node address on this network, you can assign either an IP address or a Host Name to the node you will communicate with. Then, in your application, select only the Modbus Plus node address as your destination node address. The Ethernet MBX Driver automatically converts this address to a proper IP address.

   Select a Modbus Plus node address to map to an IP Address/Host Name. Click *Edit*. The following screen appears.



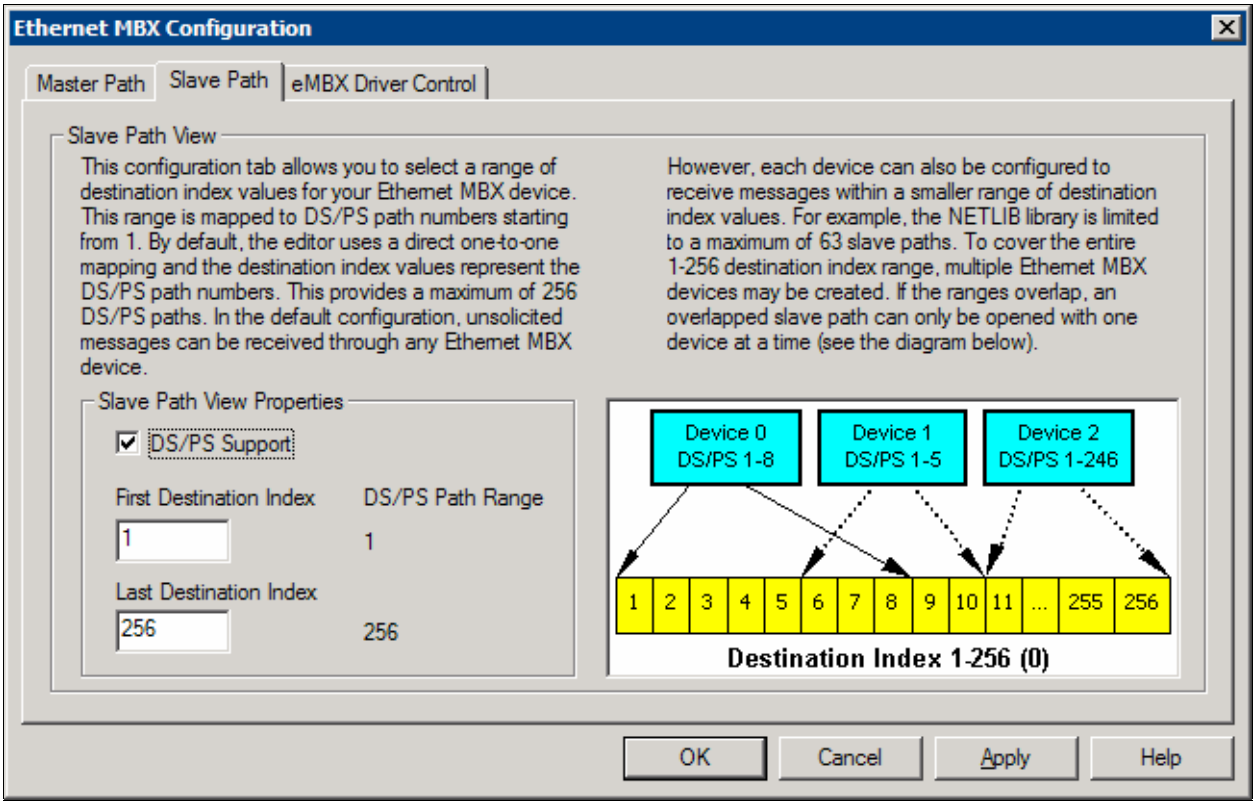
   In most cases, you will only enter the IP/Host Name address here. The *Destination Index* value is ignored when communicating to Modicon PLCs. If your destination is another Ethernet MBX node, you may have to enter the *Destination Index* value. This value is used for mapping into the *Data/Program Slave* path number of your destination Ethernet MBX device. Refer to the Theory of Operation section for more information.

Enter the *Destination Index* value. Then, enter the IP address or Host Name in the *IP Address/Host Name* box. Click *OK* to return to the Ethernet MBX Configuration Editor.



Repeat this step until all destination nodes are properly mapped.

5. Select the *Slave Path* tab.



Slave Path Support should be used only if your application supports and requires unsolicited (Data/Program Slave Path) communications. By default, support for this type of communication is disabled.

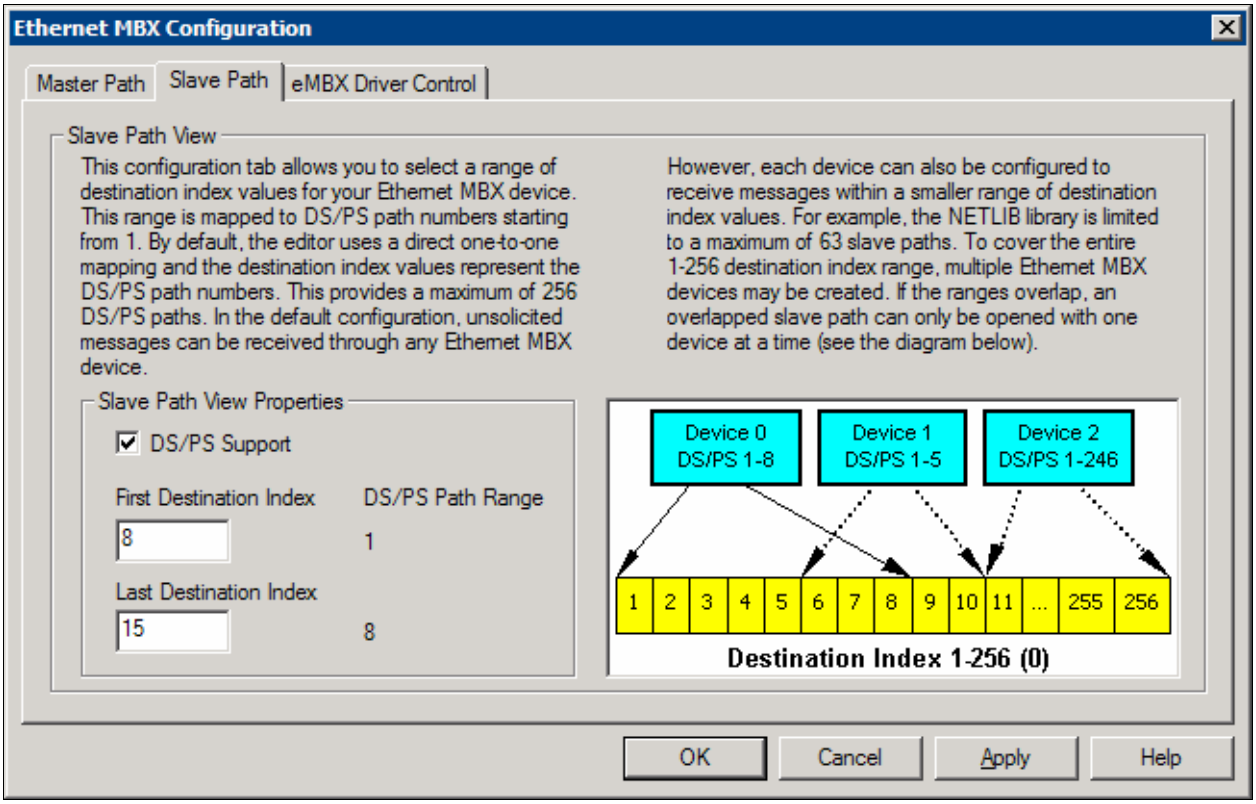
Note:	Checking/clearing the *DS/PS Support* box enables/disables support for unsolicited communications for all Ethernet MBX devices on the system.

If your application does not require Slave Path support, clear the *DS/PS Support* check box and skip the rest of this step.

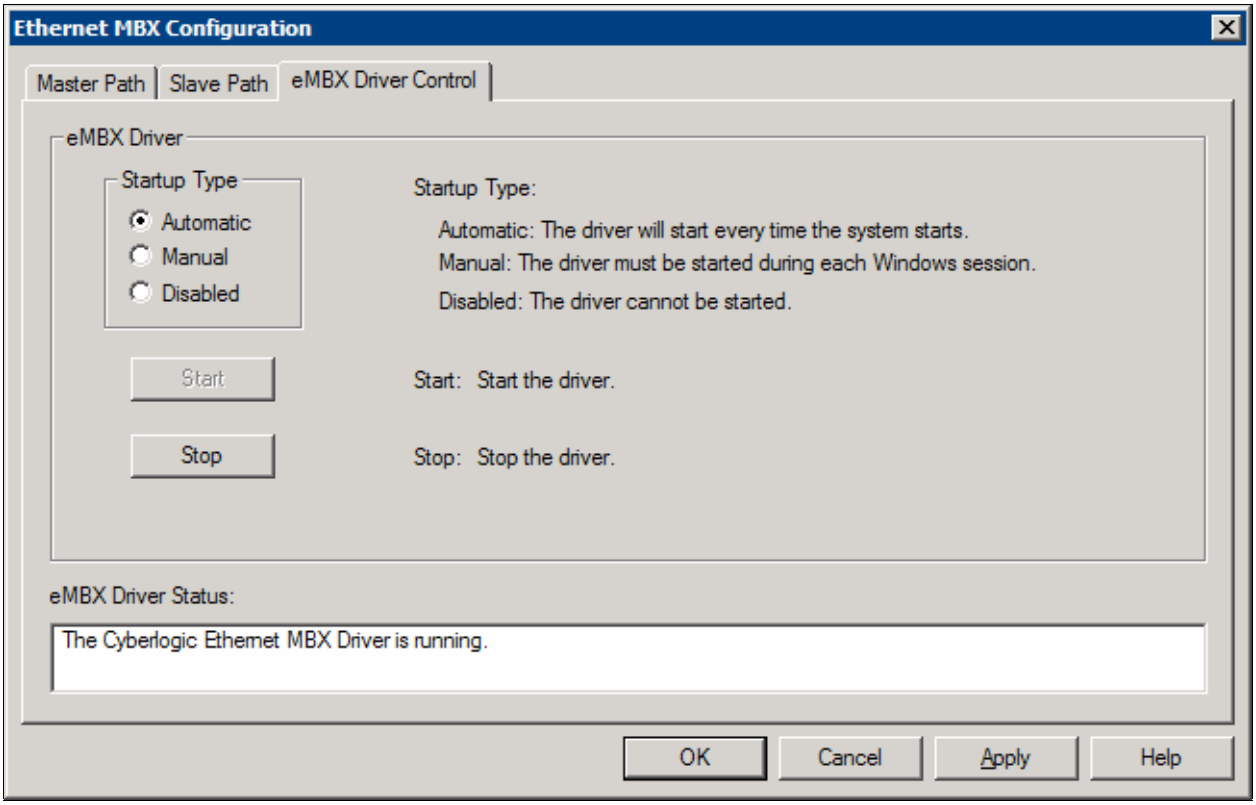
If your application does require Slave Path support, check the *DS/PS Support* check box.

The *First Destination Index* and the *Last Destination Index* fields allow proper mapping of messages into the Data/Program Slave Path numbers.

Enter proper values for the *First Destination Index* and the *Last Destination Index*.



6. Choose the *eMBX Driver Control* tab.



By default, the Ethernet MBX Driver is configured for the Automatic startup type. In this mode, the driver starts automatically when the operating system boots. Most users should select the Automatic startup type.

If you want to control the driver manually, select the *Manual* option in the *Startup Type* choices and click *Apply*.

7. Click *OK* to return to the MBX Driver Configuration editor.

Your Ethernet MBX device is fully configured.

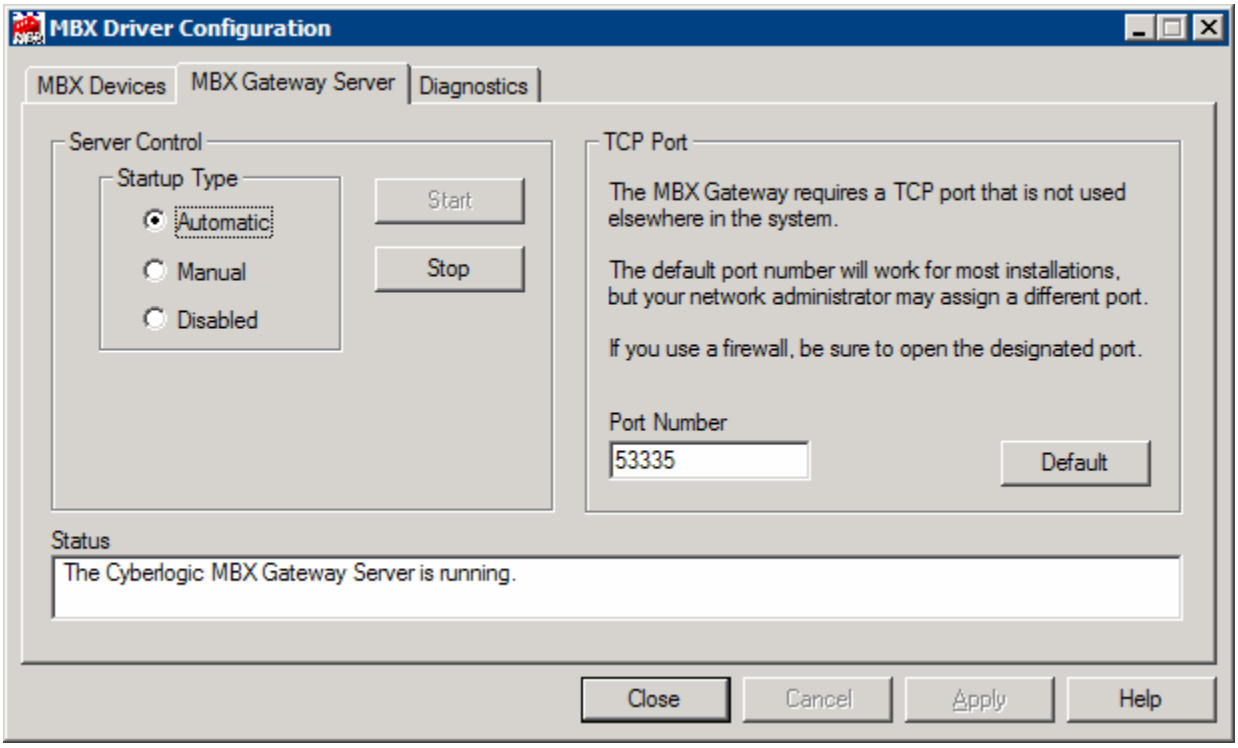
If you plan to use the MBX Gateway Driver on remote Windows XP/2000/NT nodes, go to the MBX Gateway Server Configuration segment. Refer to the Remote Connectivity section for more information on this capability.

Otherwise, go to the Diagnostics segment.

MBX Gateway Server Configuration

The MBX Driver comes with the MBX Gateway Server, a remote connectivity component of the MBX family. The Gateway Server, running in a system, allows remote nodes to access all configured MBX devices present on the same system.

8. Select the *MBX Gateway Server* tab. You will see the following screen:



9. By default, the Gateway Server is created in the Automatic startup type. In this mode of operation, the server will start whenever the system is booted, and this is the mode that most users should select. If you want to control the Gateway Server manually, choose *Manual* in the Startup Type selection.

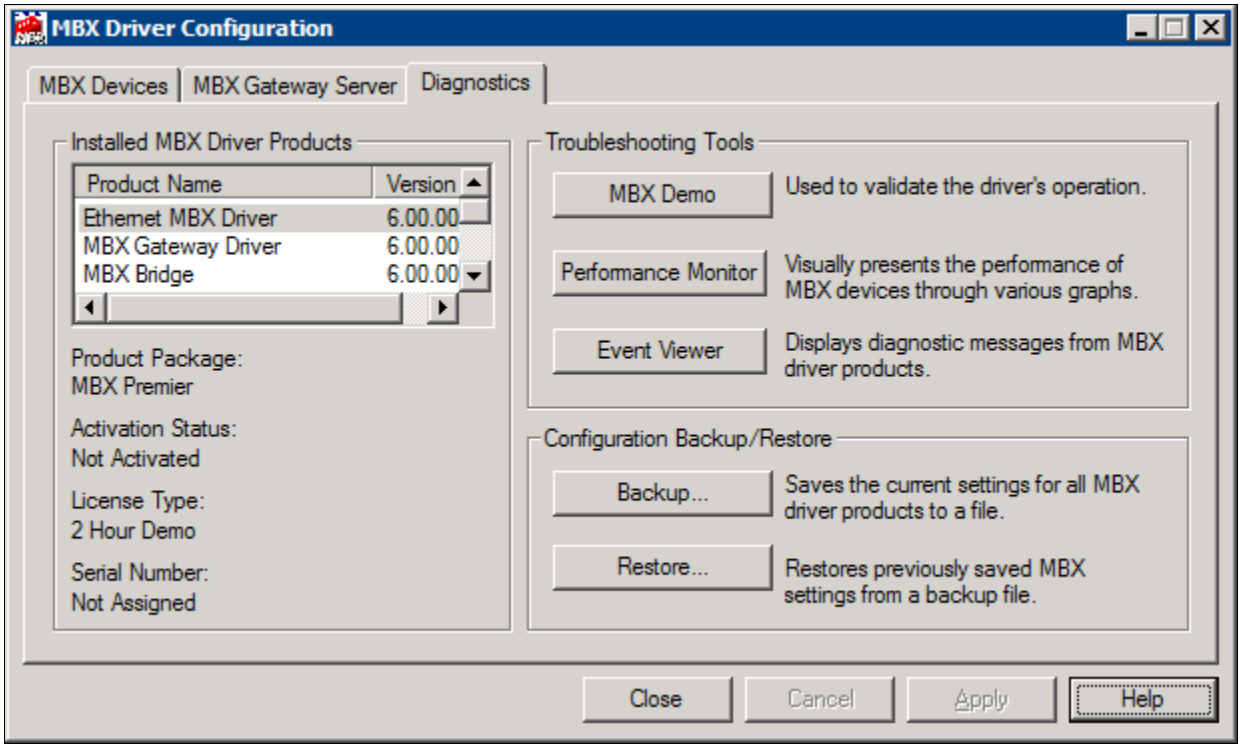
10. You must enter a TCP port that is not used elsewhere in the system. The default, 53335, will work for most installations, but this port may be taken in some unusual cases. If that applies to your system, the system administrator will assign a different port and you can set that value here.

    If your system uses a firewall, you must open the port that you configure here. 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.

    The Ethernet MBX Driver configuration is now complete. Go on to the Diagnostics segment, which will introduce you to the diagnostic features of the product.

Diagnostics

11. Select the *Diagnostics* tab. You will see the following screen:



The left pane of this screen shows all MBX driver products 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 demo mode.

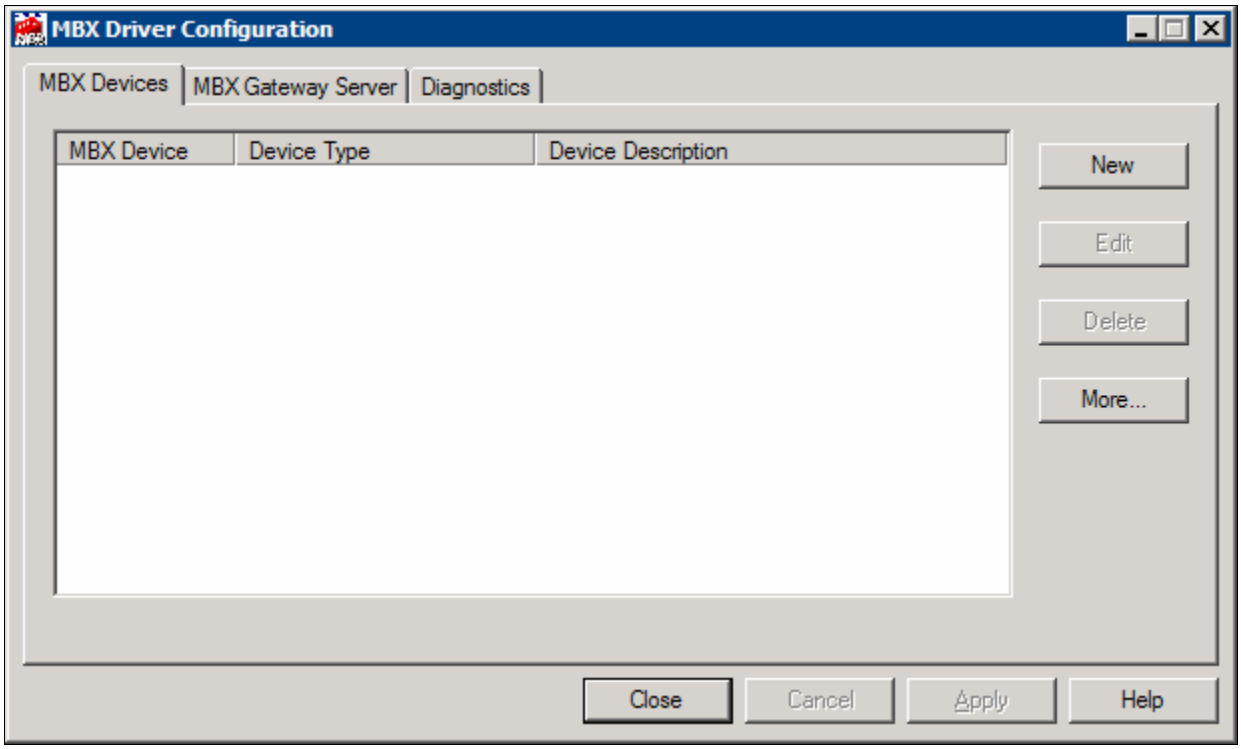
The right pane of the screen provides shortcuts to troubleshooting and backup/restore tools. You should run the MBX Demo program after configuring the MBX Driver to ensure the driver is configured and running properly. You should also back up your configuration. Refer to the Diagnostics Tab discussion in the MBX Driver Configuration Editor section for more information on how to use these tools.

Creating an Ethernet MBX Device

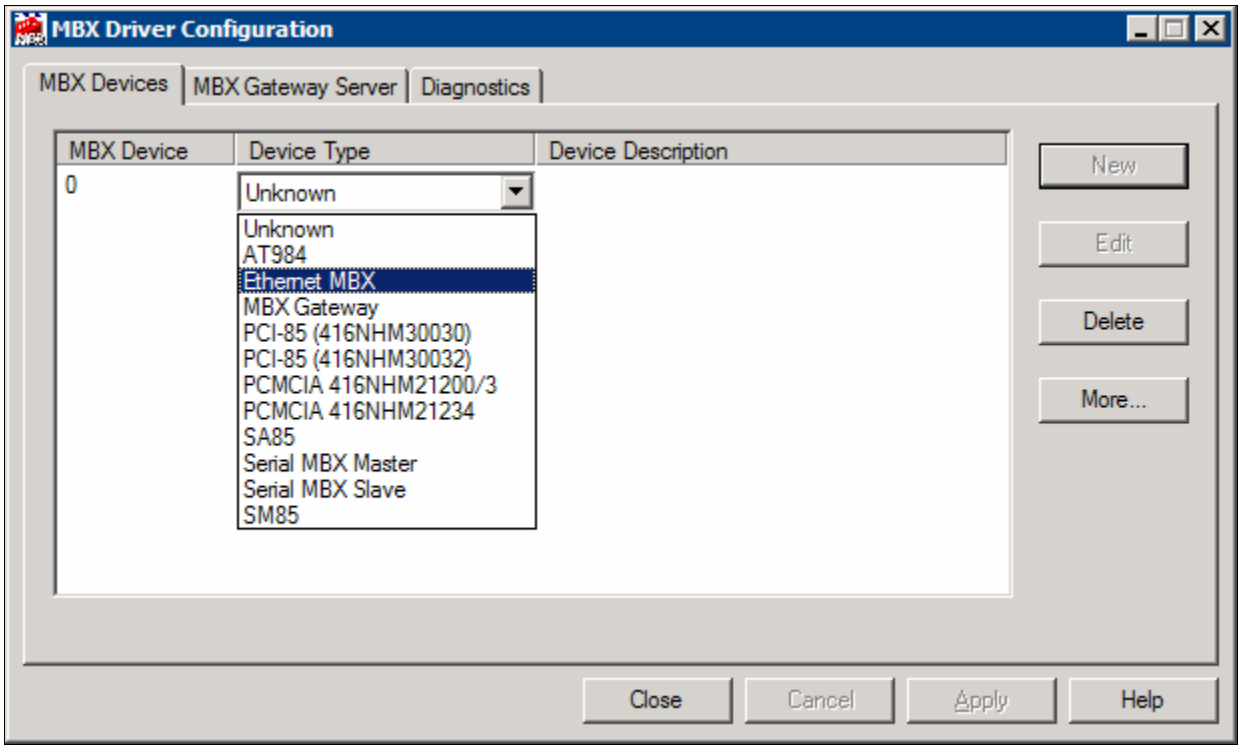
This section describes how to create a new Ethernet MBX device. Once an Ethernet MBX device is created, refer to the Ethernet MBX Configuration Editor section for information on editing the existing configuration. If you need a quick-start guide or a step-by-step configuration session tutorial, go back to the Typical Driver Configuration section.

1. From the Windows Start menu, locate the MBX Ethernet Driver submenu and select the *MBX Driver Configuration* menu item.

   Running the editor for the first time displays the following screen:



2. Click *New* and select *Ethernet MBX* from the drop-down list.



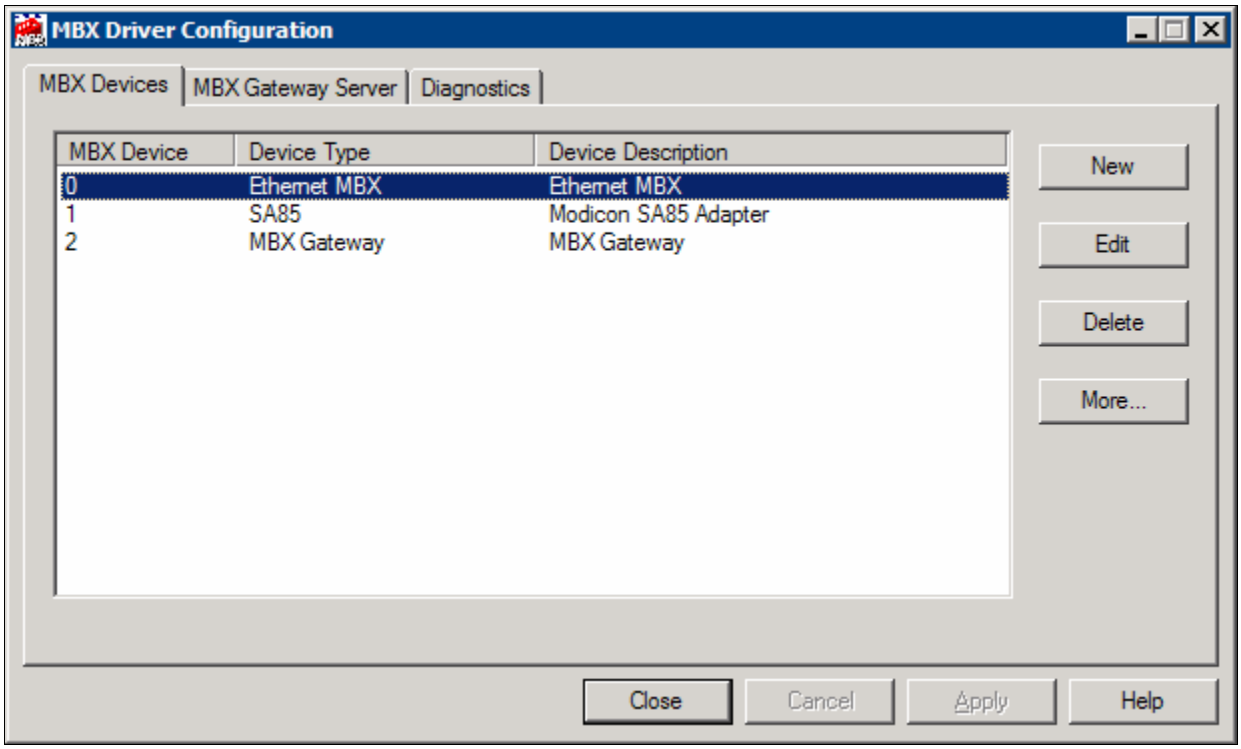
The MBX Driver Configuration Editor will automatically dispatch the Ethernet MBX Configuration Editor. For detailed information about this editor, proceed to the Ethernet MBX Configuration Editor section.

Editing Device Configuration

This section shows you how to reconfigure an existing Ethernet MBX device. For information on creating an Ethernet MBX device, refer to the Creating an Ethernet MBX Device section. If you need a quick-start guide or a step-by-step configuration session tutorial, go to the Typical Driver Configuration section.

1. From the Windows Start menu, locate the MBX Ethernet Driver submenu and select the *MBX Driver Configuration* menu item.

2. Select *Ethernet MBX* and click *Edit*.



The MBX Driver Configuration Editor will automatically dispatch the Ethernet MBX Driver Configuration Editor. For detailed information about this editor, proceed to the Ethernet MBX Configuration Editor section.

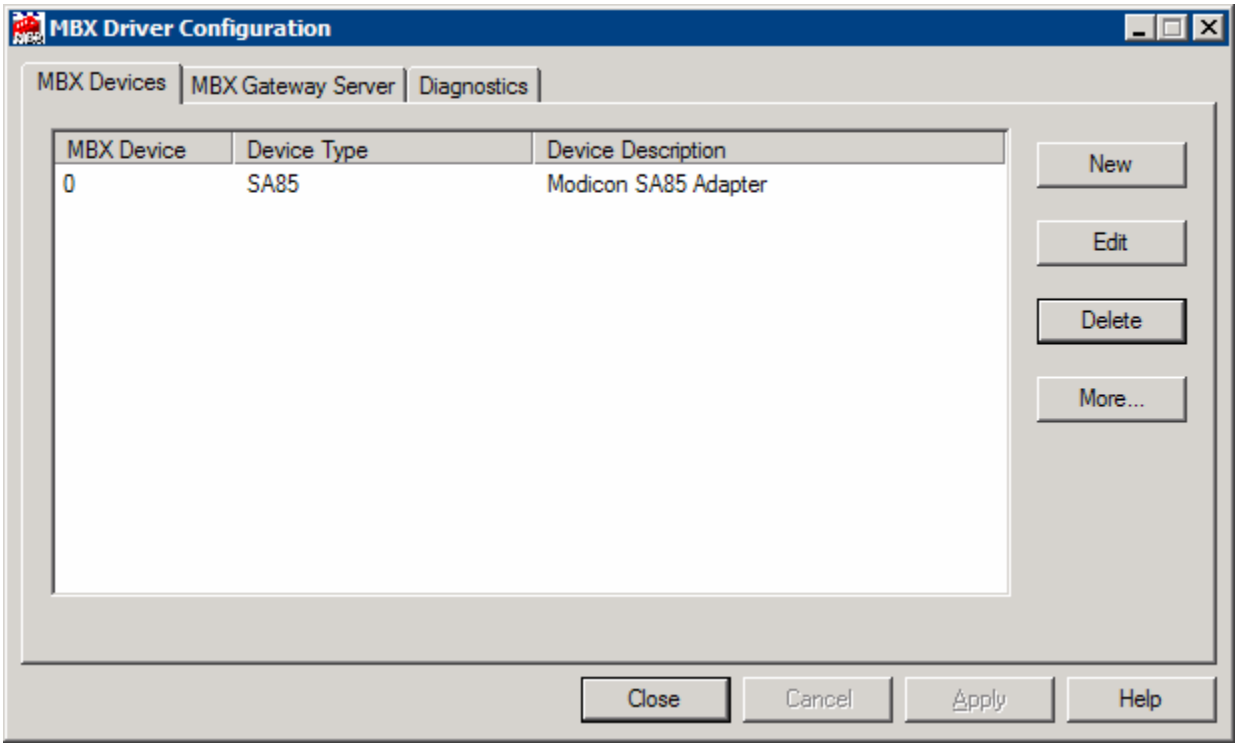
MBX Driver Configuration Editor

The MBX Driver Configuration Editor is a common component of MBX driver products. When configuring an Ethernet MBX type device, the MBX Driver Configuration Editor automatically dispatches the Ethernet MBX Configuration Editor. Both editors are well-integrated, allowing for seamless editing.

The MBX Driver Configuration Editor consists of three tabs: MBX Devices Tab, MBX Gateway Server Tab and Diagnostics Tab. The following sections provide complete descriptions of these pages.

MBX Devices Tab

Every MBX device must be configured in the MBX Device tab before it can be used by client applications. The MBX Device 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 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

Identifies the type of the MBX device, such as SA85, Ethernet MBX or MBX Gateway.

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 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.

Creating a New MBX Device

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

Upon selecting the device type, the MBX Driver Configuration Editor will automatically dispatch the Host Interface Adapter Configuration Editor.

Deleting an Existing MBX Device

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

Editing an Existing MBX Device configuration

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

Changing Device Description

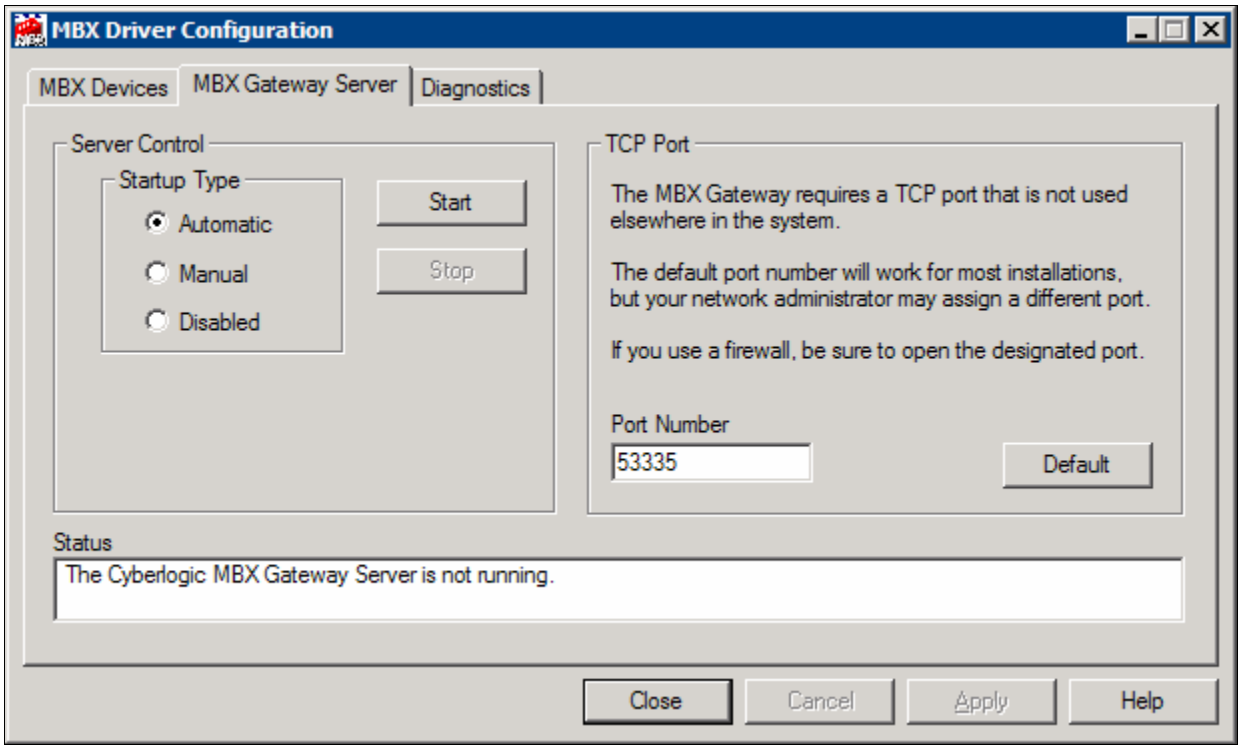
Select the device and click the *More…* button or right-click and select *Edit Description* from the pop-up menu. Modify your device description and press *Enter* when done.

Changing Device Type

Select the device and click the *More…* button or right-click and select *Change Type* from the pop-up 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 following screen will depend on the device type selected.

MBX Gateway Server Tab

The MBX Driver comes with the MBX Gateway Server, a remote connectivity component of the MBX family. The Gateway Server allows remote nodes to access all configured MBX devices present on the server system, including MBX Driver devices. You set up the Gateway Server on the *MBX Gateway Server* tab.



Selecting the Startup Type

By default, the Gateway Server is created in the Automatic startup type. In this mode of operation, the server will start whenever the system is booted, and this is the mode that most users should select. If you want to control the Gateway Server manually, choose *Manual* in the Startup Type selection.

If you select *Disabled* while the Gateway Server is running, it will continue to run until you stop it or reboot the system. After that, it will not run until you change the startup type to Automatic or Manual.

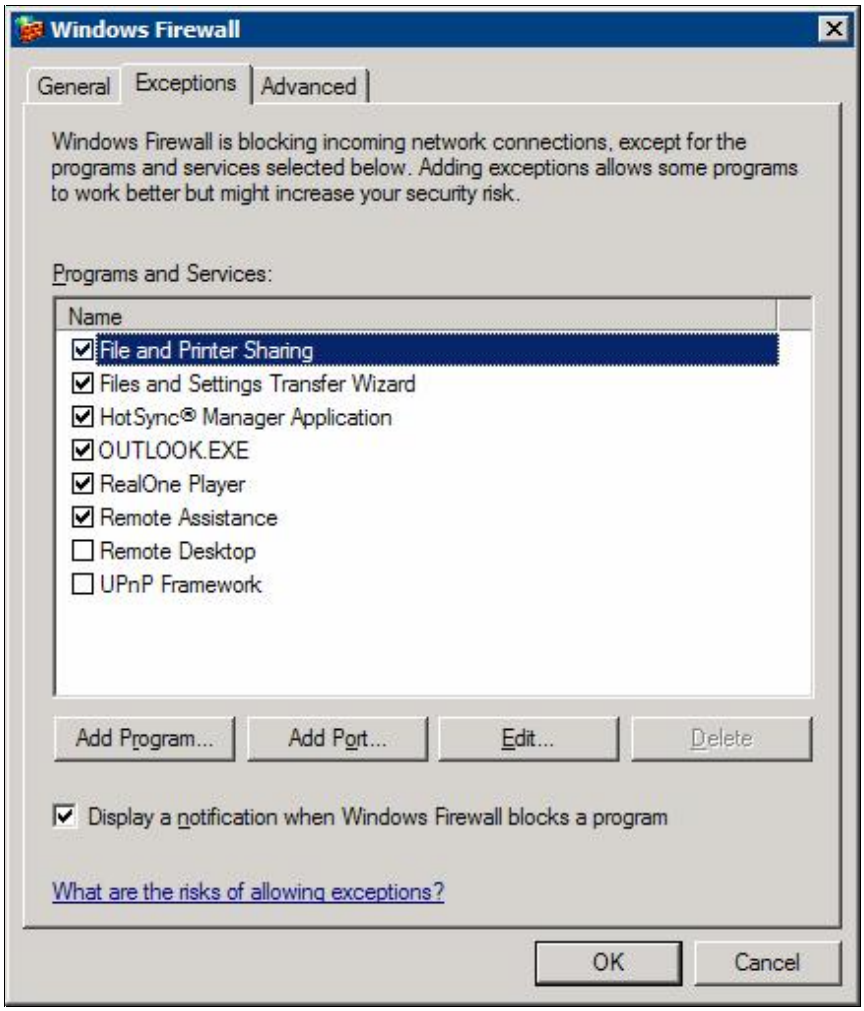
Start/Stop the Gateway Server

Click the *Start* or *Stop* button.

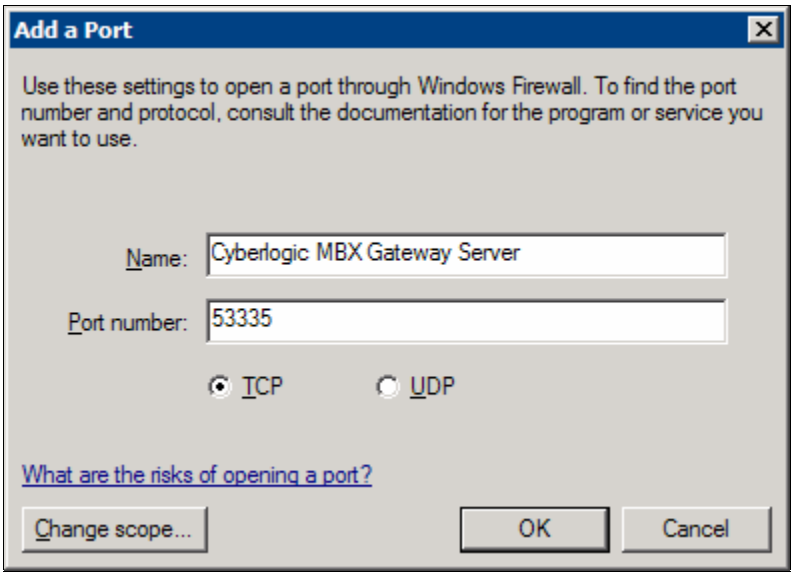
Selecting the TCP Port

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 and you can set that value here.

If your system uses a firewall, you must open the port that you configure here. The procedure will depend upon the firewall you are using. To open a port using Windows XP's firewall, go to *Control Panel*, open *Windows Firewall* and select the *Exceptions* tab. Now click *Add Port...* .

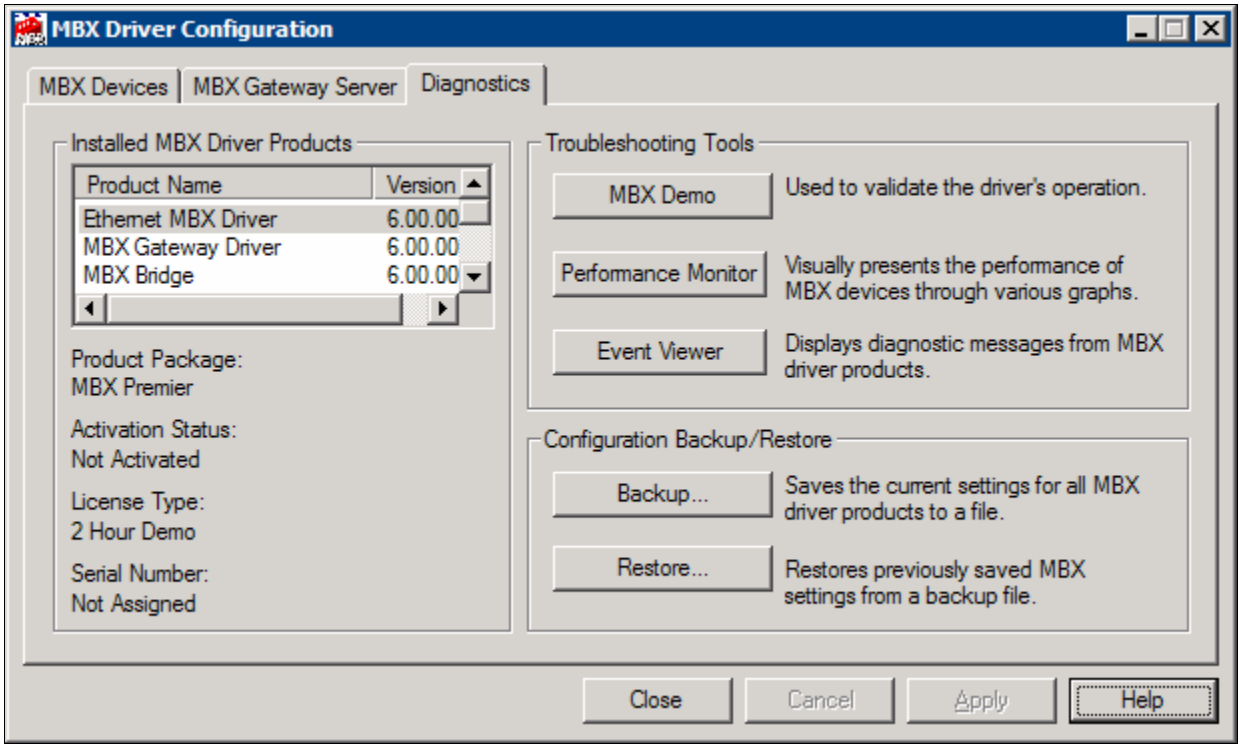


Enter a descriptive name in the *Name* field and the port you wish to open in the *Port number* field. Select *TCP* and click *OK* twice to save your changes and exit.



Diagnostics Tab

The left pane of this screen shows all MBX driver products 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 demo mode.



The right pane of the screen is divided into two groups: Troubleshooting Tools and Configuration Backup/Restore.

Troubleshooting Tools

The Troubleshooting Tools group provides shortcuts to diagnostic tools. Run the MBX Demo program after configuring the MBX Driver to ensure the driver is configured correctly and running properly.

To observe the performance of your communications, run the Performance Monitor.

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

Refer to the Validation & Troubleshooting section for more information on these features.

Configuration Backup/Restore

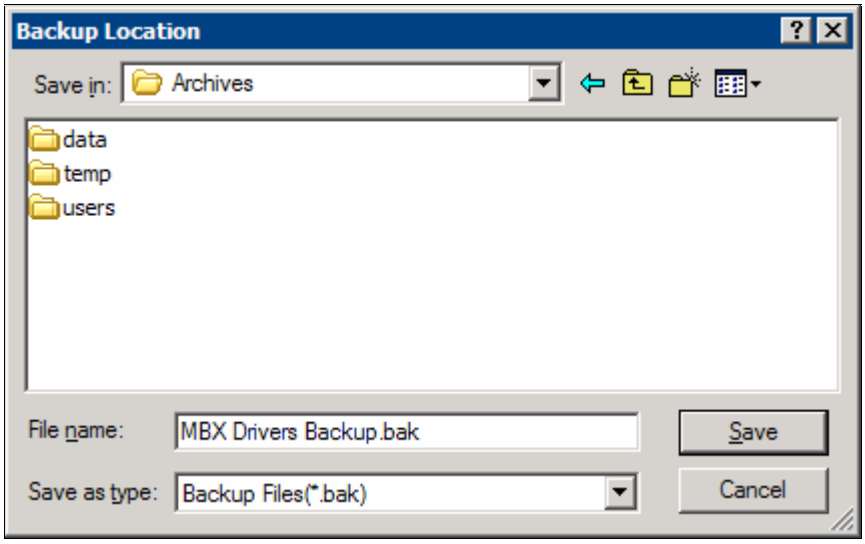
Creating and configuring MBX devices may be a time-consuming process. Therefore, we strongly recommend that you backup the configuration data.

The *Backup…* and *Restore…* buttons in the Configuration Backup/Restore group can be used to backup and restore configurations of all MBX driver products on your system.

Backup Configuration

Click the *Backup…* button. Browse for your backup directory and enter the file name of your configuration backup file. By default, the last-used directory will be selected.

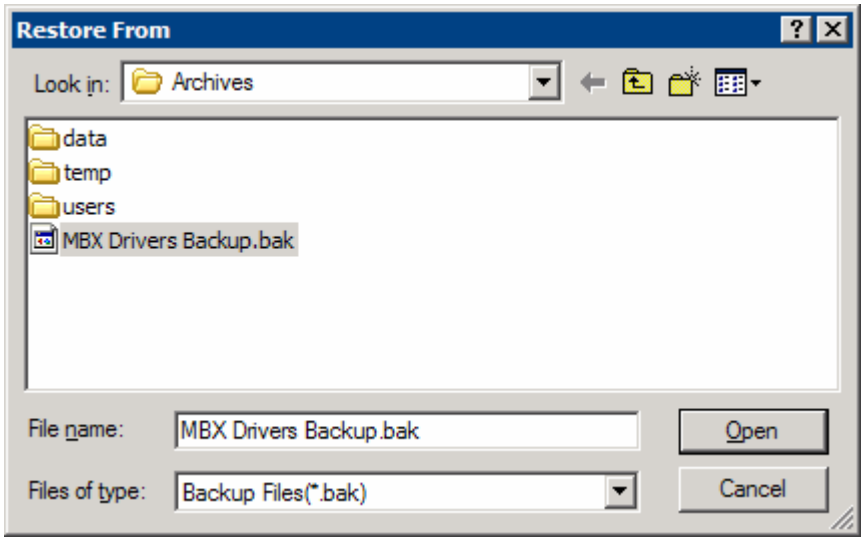
Click the *Save* button to complete the backup operation.



Restore Configuration

Click the *Restore…* button. Browse for your configuration backup file. By default, the last used directory will be selected.

Select the backup file and click the *Open* button to complete the restore operation. Restart the system to ensure proper operation of the restored devices.



Configuration Backup/Restore Utility

The MBX driver products also provide a utility program (ClMbxCfg.exe) you can use to backup and restore MBX device configurations. The program is located in the \Program Files\Common Files\Cyberlogic Shared\ directory. It accepts the following command line switches:

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

For example, to save the configuration of all MBX devices in the MbxCfg.bak file (located in 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 saved in MbxCfg.bak, use the following command:

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

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

Ethernet MBX Configuration Editor

The MBX Driver Configuration Editor dispatches the Ethernet MBX Configuration Editor when editing Ethernet MBX-type devices. Both editors are well-integrated, allowing for seamless editing.

The Ethernet MBX Configuration Editor consists of three tabs: Master Path Tab, Slave Path Tab and eMBX Driver Control Tab. The following sections provide complete descriptions of these tabs.

Master Path Tab

To accommodate the Modbus Plus architecture, the Ethernet MBX Driver allows two alternative methods that applications can use to map into an IP address:

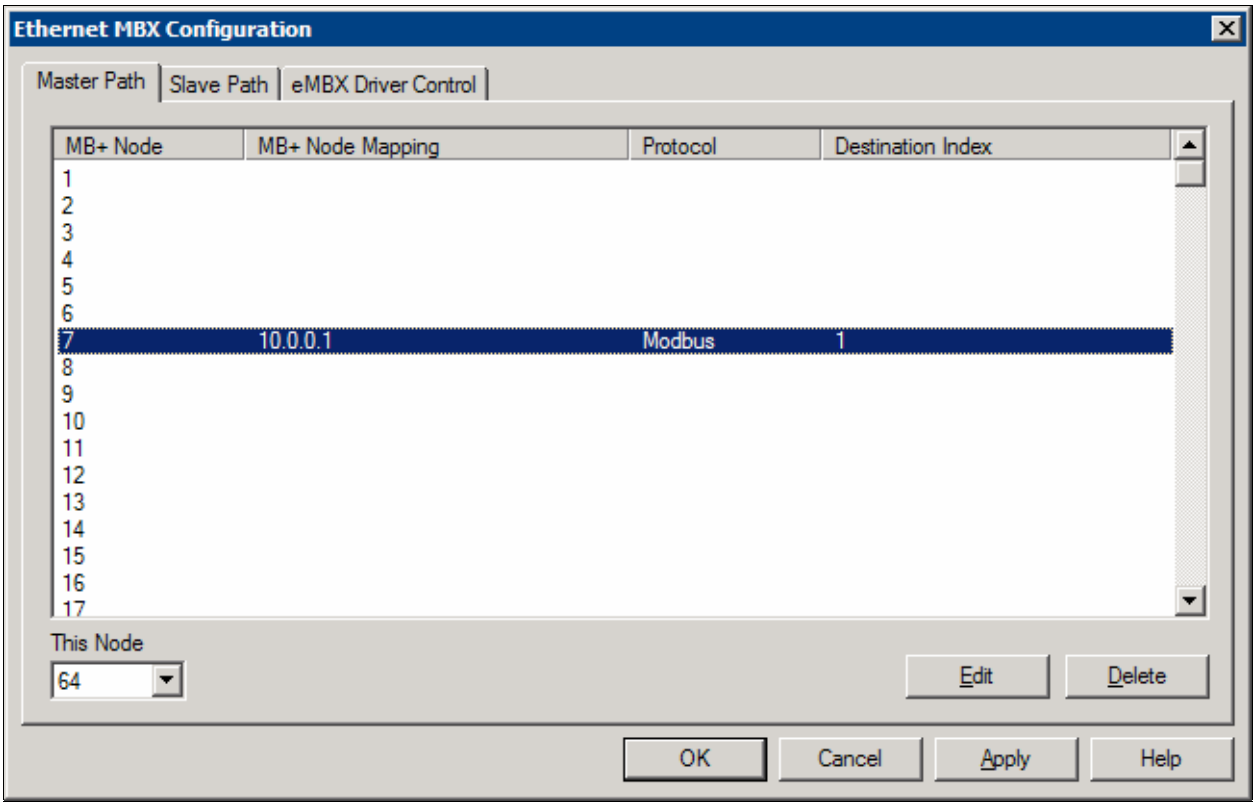
- Explicitly specifying an IP address (Explicit IP Address)
- Specifying an address by indirect mapping (Indirect Address Mapping)

Explicit IP addressing is the most flexible addressing method and requires no configuration. Some applications, however, do not allow routing addresses to fall outside the 1-64 range. For these types of applications an indirect address mapping should be used. Refer to the Theory of Operation section for a more detailed discussion.

In indirect addressing mode, Ethernet networks are viewed as a single local Modbus Plus network. For every node address on this network, you can assign either an IP address or a Host Name to the node you will communicate with. Then, in your application, just select the Modbus Plus node address as the destination node address. The Ethernet MBX Driver automatically converts this address to a proper IP address.

The Master Path tab of the Ethernet MBX Configuration Editor is primarily used to set up the mapping table used for indirect addressing. Also, since the Ethernet MBX device emulates the behavior of a Modbus Plus host adapter card, you can assign a Modbus Plus node address to your MBX device here as well.

The main part of the Master Path tab displays the content of the indirect addressing mapping table. The information is presented in four columns: *MB+ Node*, *MB+ Node Mapping*, *Protocol* and *Destination Index*.



MB+ Node

Modbus Plus allows a maximum of 64 nodes with addresses in the range of 1-64. The Ethernet MBX Driver extends this range to 1-256. However, some applications may limit the value in the first routing byte to a range of 1-64. Node addresses in the range of 65-256 should be used only if your application allows that range. More Ethernet MBX devices can be created as more mapping entries are needed.

MB+ Node Mapping

For every Modbus Plus node address, you can assign either an IP address or a Host Name to the node you will communicate with.

Protocol

Currently, the Ethernet MBX Driver supports only the Modbus message protocol, which is used by Quantum PLCs when communicating over the Ethernet network. Future releases of this product may support additional message protocols.

Destination Index

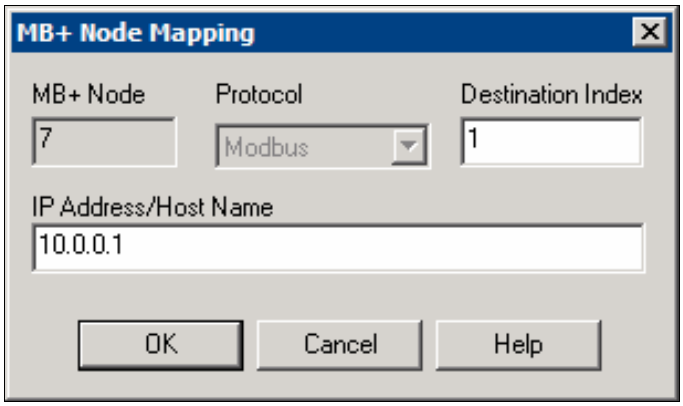
The *Destination Index* value is ignored when communicating to Modicon PLCs. If your destination is another Ethernet MBX node, you may need to select the *Destination Index* value. This value is used for mapping into the *Data/Program Slave* path number of your destination Ethernet MBX device. Refer to the Theory of Operation section for details.

This Node

Because the Ethernet MBX device emulates the behavior of a Modbus Plus adapter card, you must assign a Modbus Plus node address to this device by selecting the address from the *This Node* drop-down list. By default, node address *64* is used.

Setting Indirect Address Mapping

Select a node address to map to an IP/Host Name address. Click *Edit* or double-click the selected entry. You will see the following dialog.



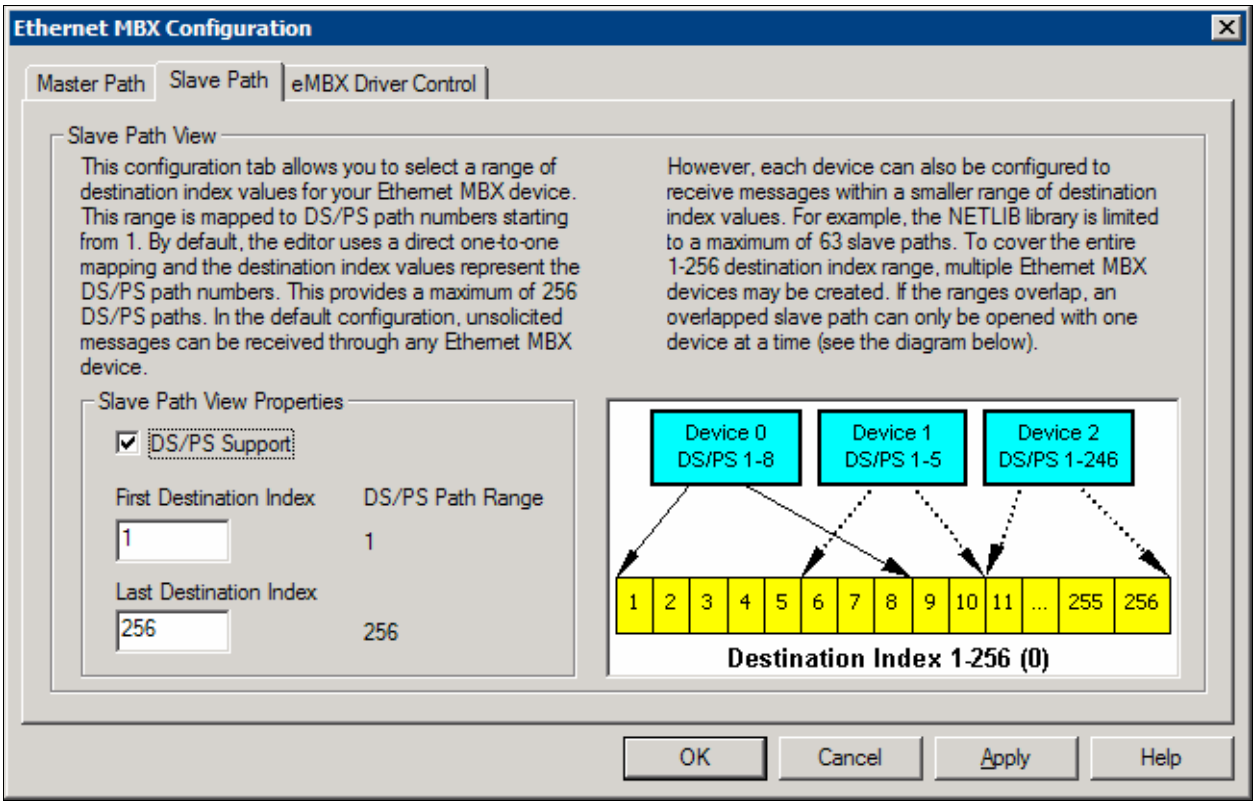
Enter the *Destination Index* value and then enter the IP address or Host Name in the *IP Address/Host Name* box. Click *OK* to return to the Ethernet MBX Configuration editor.

Slave Path Tab

The Modbus Plus architecture has a concept of Data/Program Slave Paths when receiving unsolicited messages. Modbus protocol used by Ethernet communications does not have a concept of these communication paths. Instead, any application intending to receive unsolicited messages must listen to TCP/IP port 502. The Ethernet MBX Driver resolves this problem by emulating Modbus Plus.

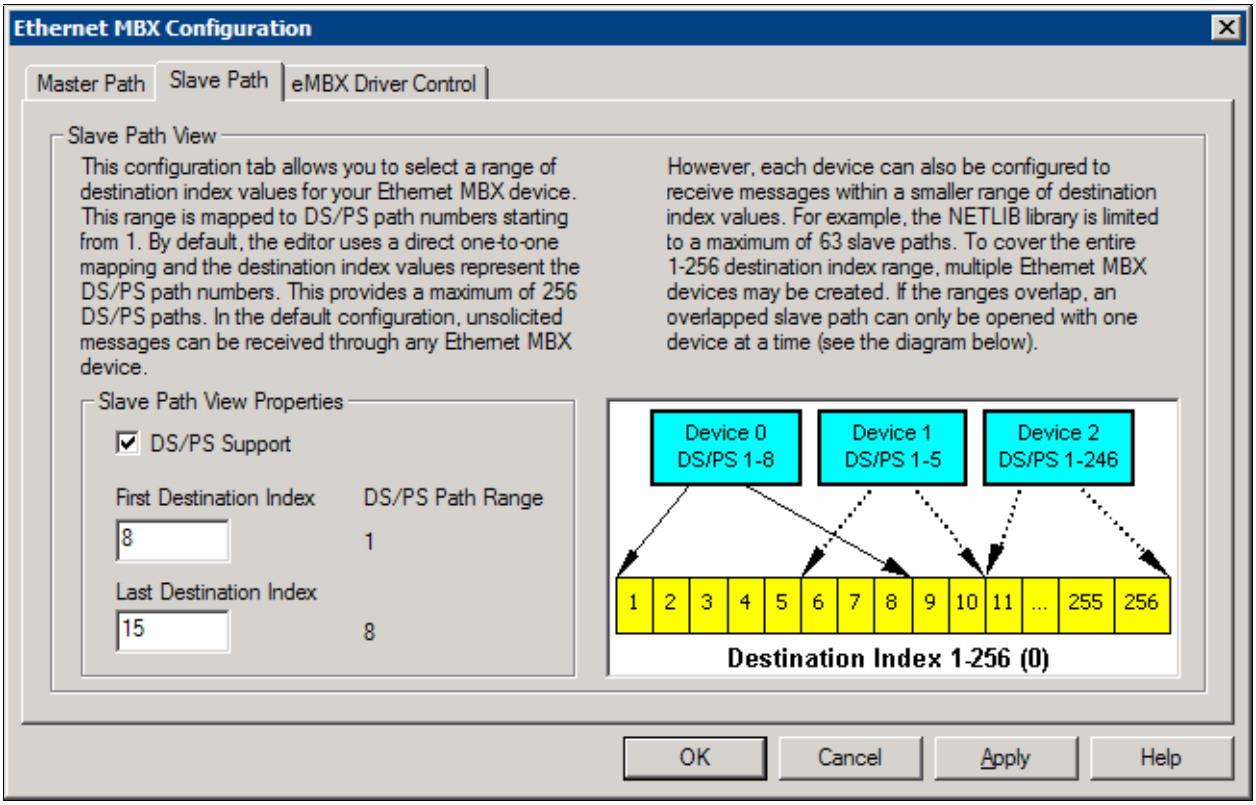
To accommodate the Modbus Plus architecture, Ethernet MBX uses the destination index byte value sent along with the message for mapping into the Data/Program Slave Path number. The type of path is determined by the Modbus command code embedded in the message. Data access messages are routed to DS paths while programming messages are routed to PS paths.

The *Slave Path* tab allows the user to select a range of destination index values that are mapped to DS/PS path numbers starting from path number 1. By default, a direct one-to-one mapping is used and a destination index value represents a DS/PS path number. That provides a maximum of 256 DS/PS paths. By comparison, Modbus Plus supports a maximum of eight DS/PS paths.



In the default configuration, unsolicited messages can be received through any logical device. If broken up into ranges, each device can receive messages with a certain range of destination index values. Since the NETLIB library supports a maximum of 63 Slave Paths, multiple logical devices may be required to cover the entire 1-256 destination index range. When ranges overlap, an overlapped Slave Path can be opened through only one device at a time.

Slave Path support should be enabled only if your application supports and requires unsolicited (Data/Program Slave Path) communications. By default, support for this type of communication is disabled. To enable it, check the *DS/PS Support* checkbox.



Setting DS/PS mapping

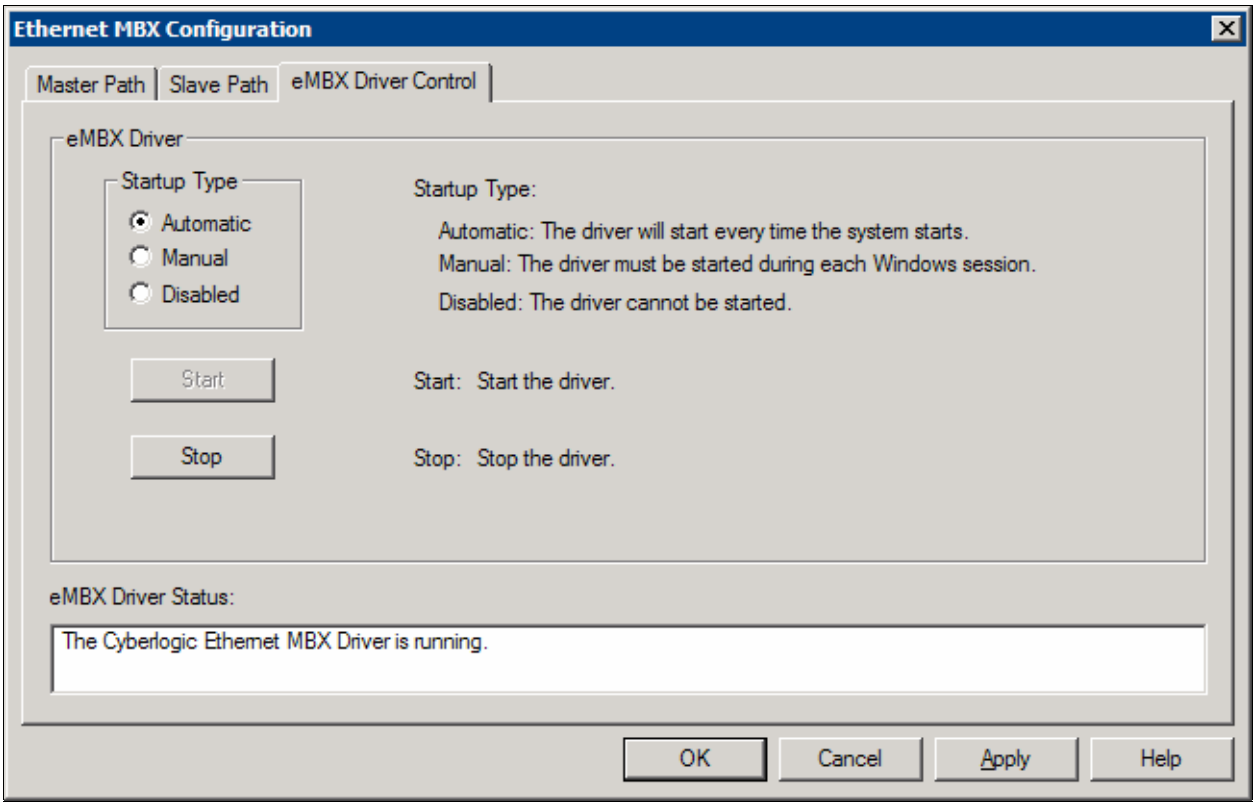
Select the *DS/PS Support* checkbox.

Note:	Checking/clearing the *DS/PS Support* box enables/disables support for unsolicited communications for all Ethernet MBX devices on the system.

Enter the proper values for the *First Destination Index* and *Last Destination Index*. The *First Destination Index* and *Last Destination Index* fields allow users to properly map messages with the selected destination index values into the Data/Program Slave Path numbers. Refer to the Theory of Operation section for more information.

eMBX Driver Control Tab

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



By default, the Ethernet MBX Driver is configured for the Automatic startup type. In this mode, the driver starts automatically when the operating system boots. Most users should select the Automatic startup type.

Note:	These settings are global and common to all Ethernet MBX devices.

Selecting the Startup Type

In the *Startup Type* section, select *Automatic*, *Manual* or *Disabled*.

Start/Stop the Ethernet MBX Driver

Click *Start* to start the driver or *Stop* to stop the driver.

VALIDATION & TROUBLESHOOTING

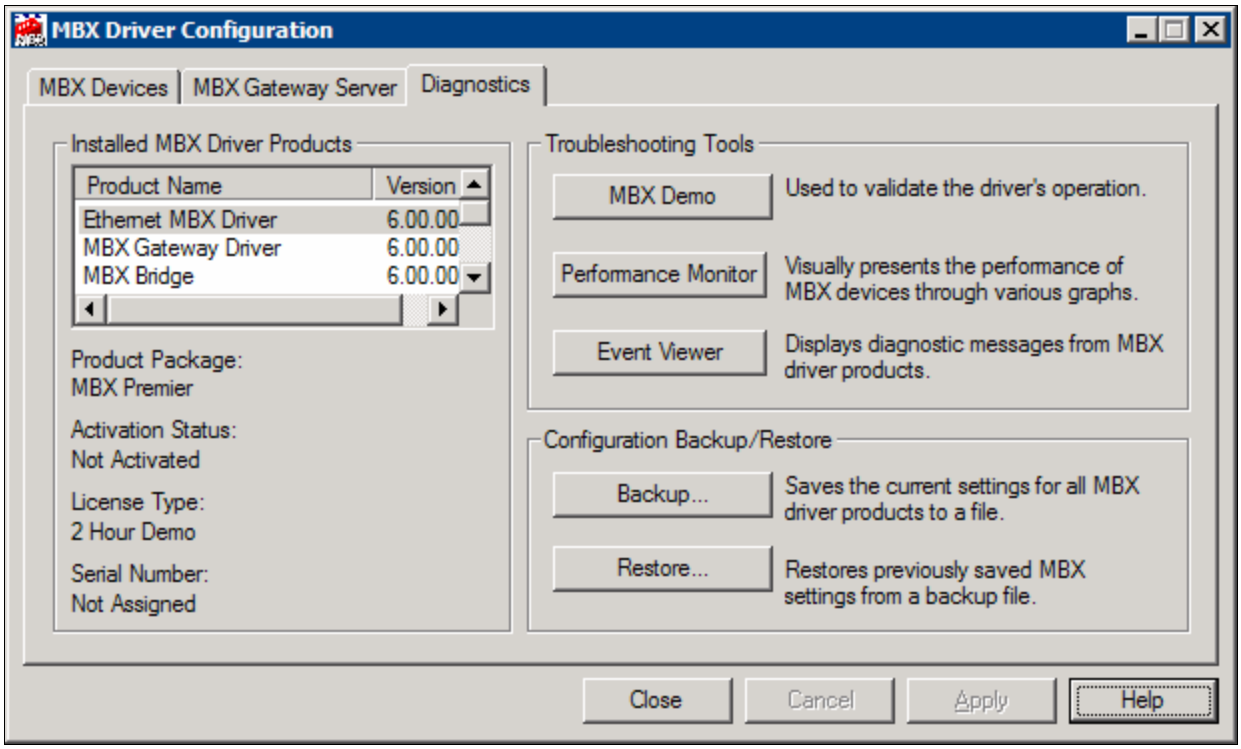
The following sections describe how the MBX Demo and Performance Monitor are used to verify that the Ethernet MBX devices are configured correctly.

If you are having difficulties communicating through an Ethernet 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 Ethernet MBX Server 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 activate the program, open the Windows *Start* menu and locate the MBX Driver submenu. From that menu, select *MBX Demo*.

Alternatively an MBX Demo button is located in the Diagnostics tab of the MBX Driver Configuration Editor:



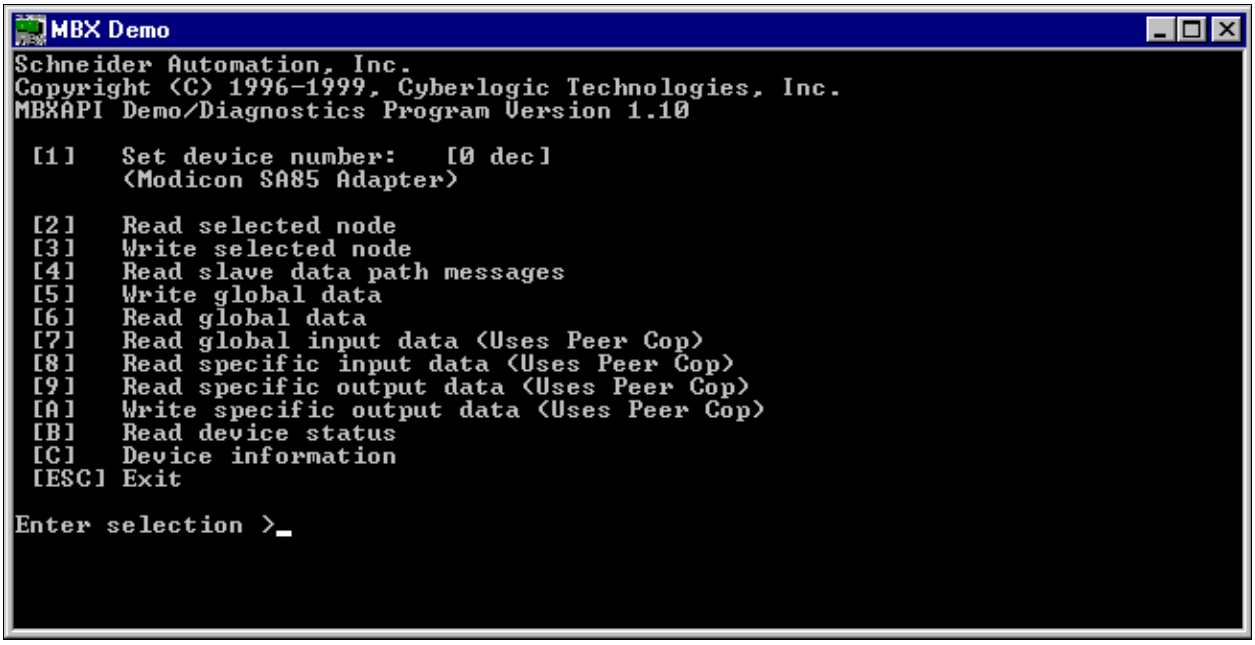
How to Use the MBX Demo Program

Although its user interface resembles a DOS-based legacy program, the MBX Demo is a 32-bit program. It will quickly access all available features of the configured MBX devices in a system for validation of driver operation.

The simple command-line interface is designed to mimic earlier tools familiar to most users. It displays numbered menu choices taking the user to secondary level screens. Pressing *Esc* at any screen returns the user to the main window shown below. Pressing *Esc* in the main window exits the program.

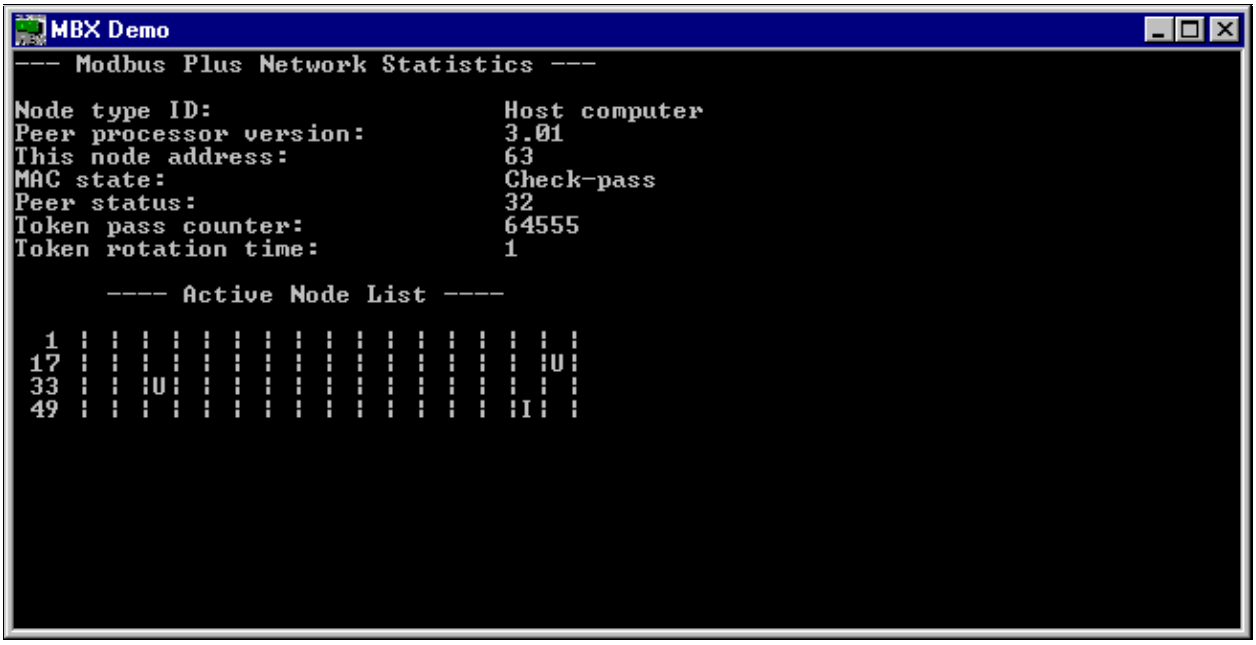
Device Number

When the MBX Demo program starts, the Device Number defaults to device 0. It may be changed by choosing the *Set device number* option.



Read Device Status

This screen shows all active nodes on the network.



Device Information

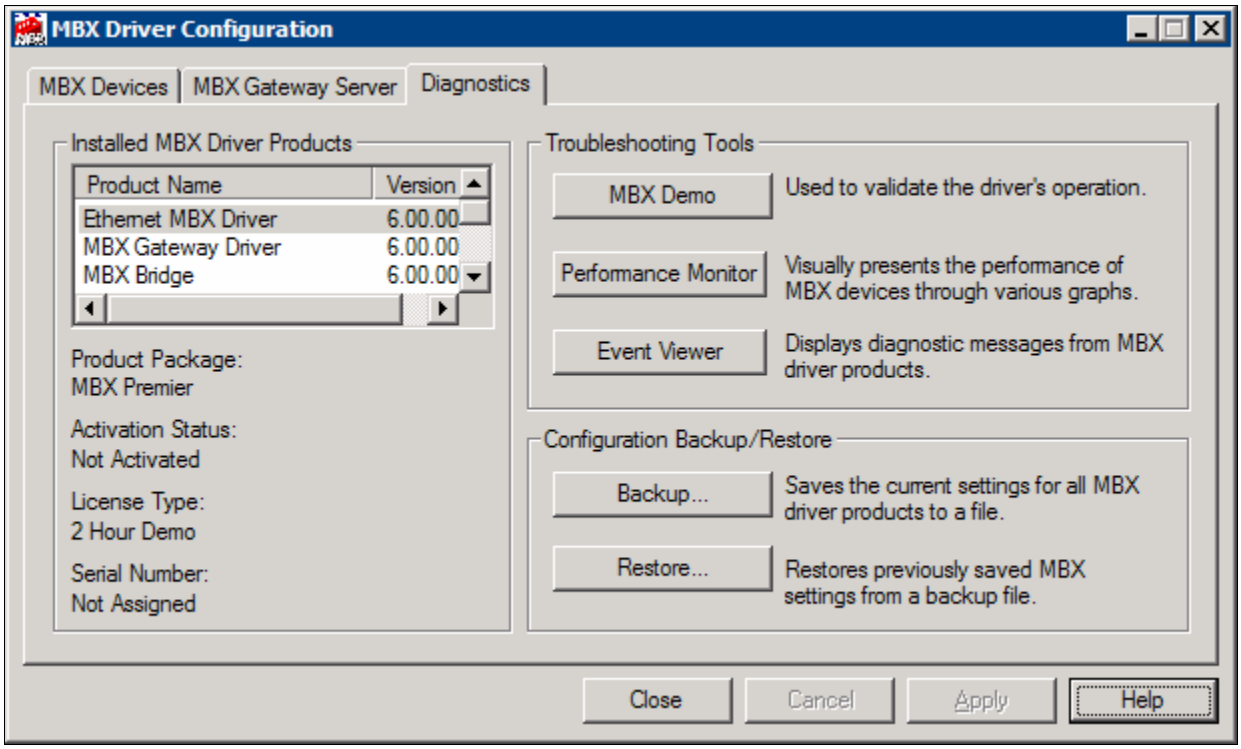
The Device information option shows configuration, statistical and diagnostic information about the driver, the device and the network.

Performance Monitor

Microsoft provides a diagnostic tool, the Performance Monitor, as part of the Windows XP/2000/NT operating system. Applications supporting the Performance Monitor allow users to monitor relevant performance information. The MBX Driver supports the Performance Monitor. Multiple devices can be monitored simultaneously for comparison.

To start this program, click on its icon from Start/Settings/Control Panel/Administrative Tools group.

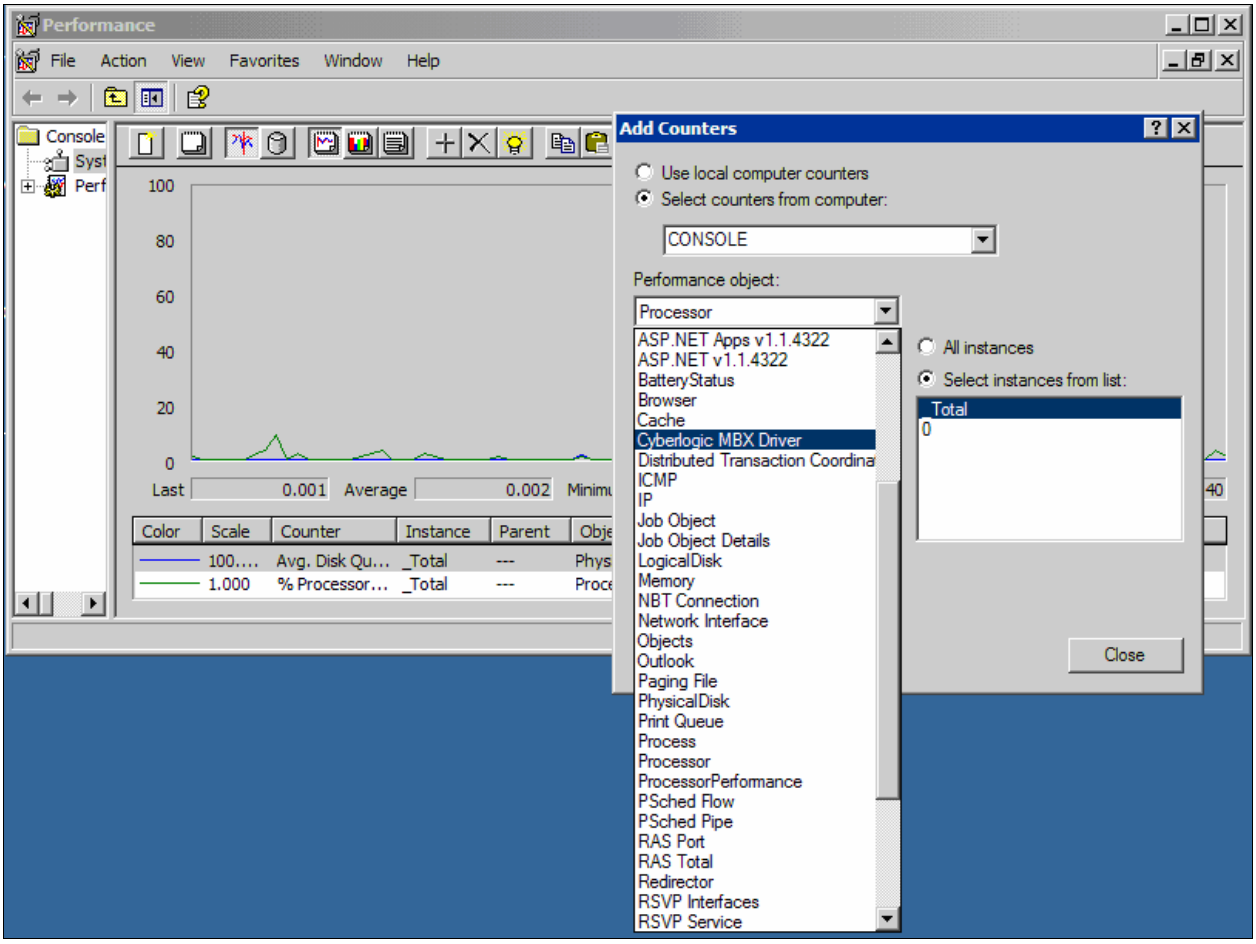
Alternatively, a Performance Monitor button is located in the Diagnostics tab of the MBX Driver Configuration Editor:



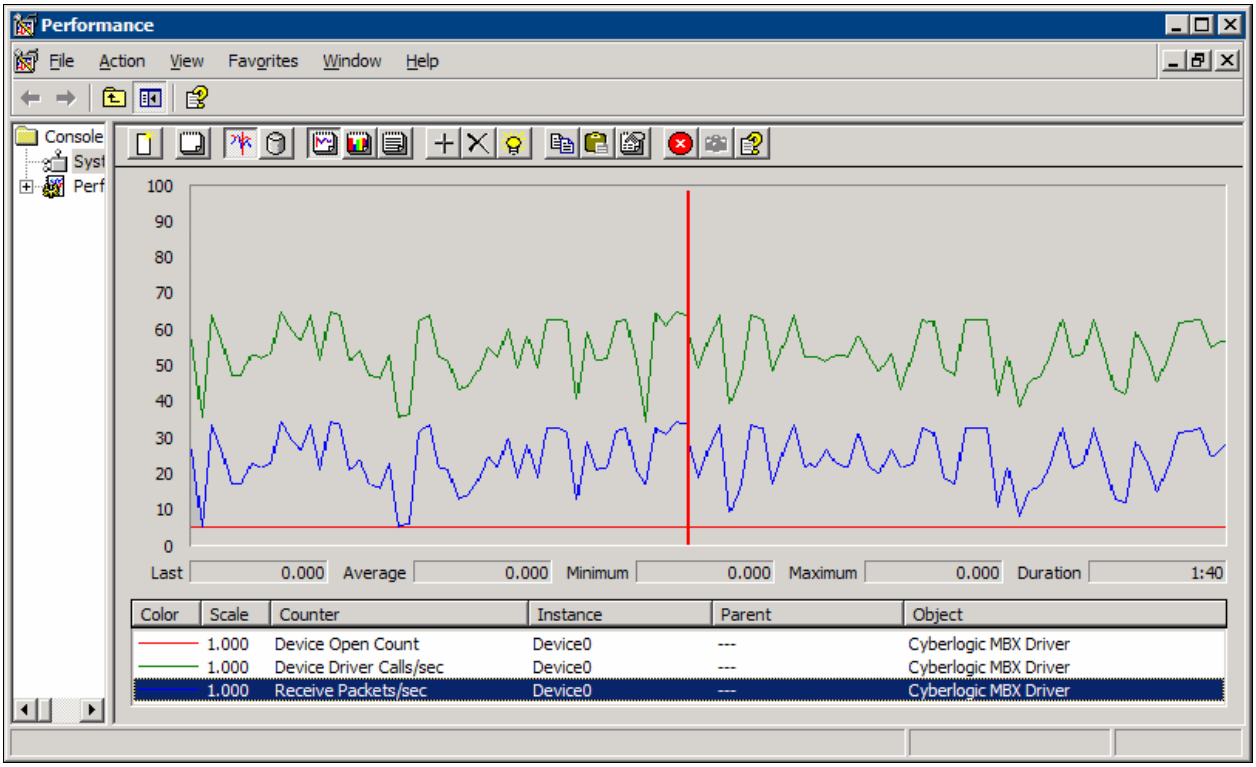
How to Use the Performance Monitor

Since extensive help is provided for this program by Microsoft, only few points relevant to the MBX Driver will be shown here.

When the Performance Monitor program starts, select the *Add to Chart* option from the Edit menu (or click the + button on the tool bar) and then select *Cyberlogic MBX Driver* from the Object list. After choosing a monitoring option, click *Add* and then *Done*.



Shown below are three of the many monitoring options.

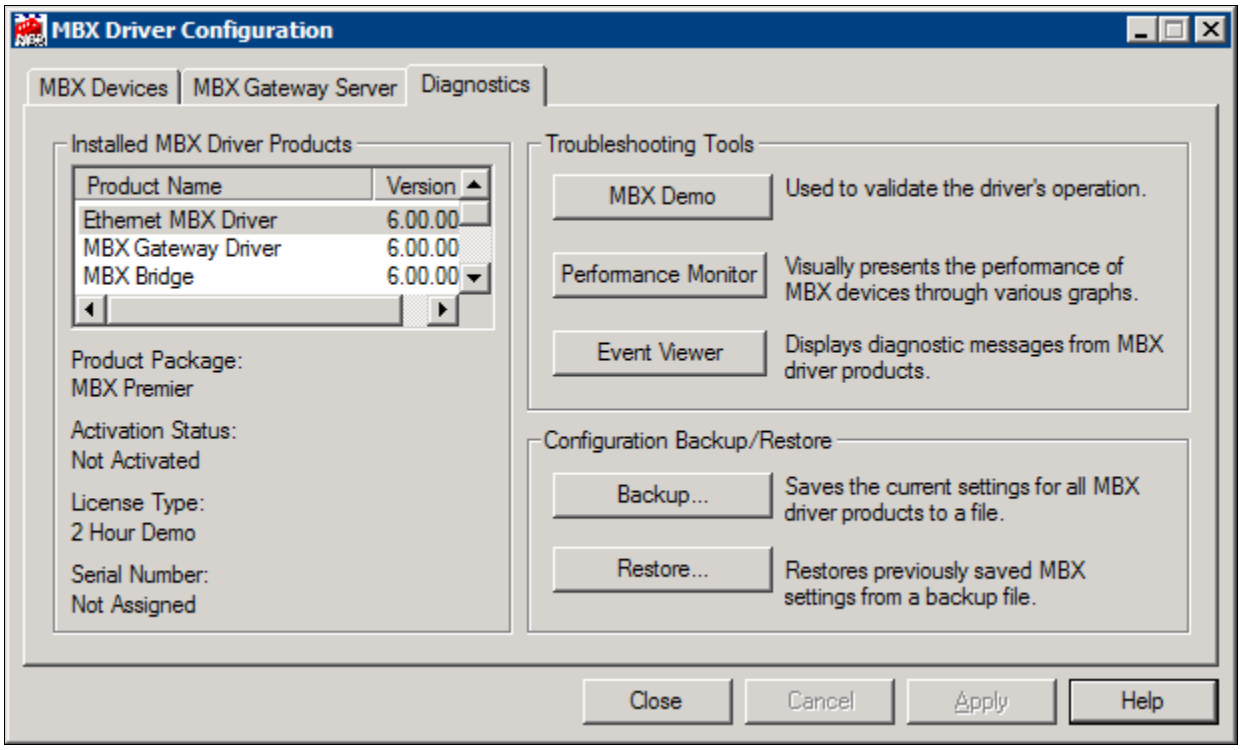


Event Viewer

During system startup, the MBX Driver may detect configuration problems. When a problem is detected, the driver sends an appropriate message to the Windows XP/2000/NT Event Logger. To view the error log messages:

1. From the Administrative Tools group run *Event Viewer*.

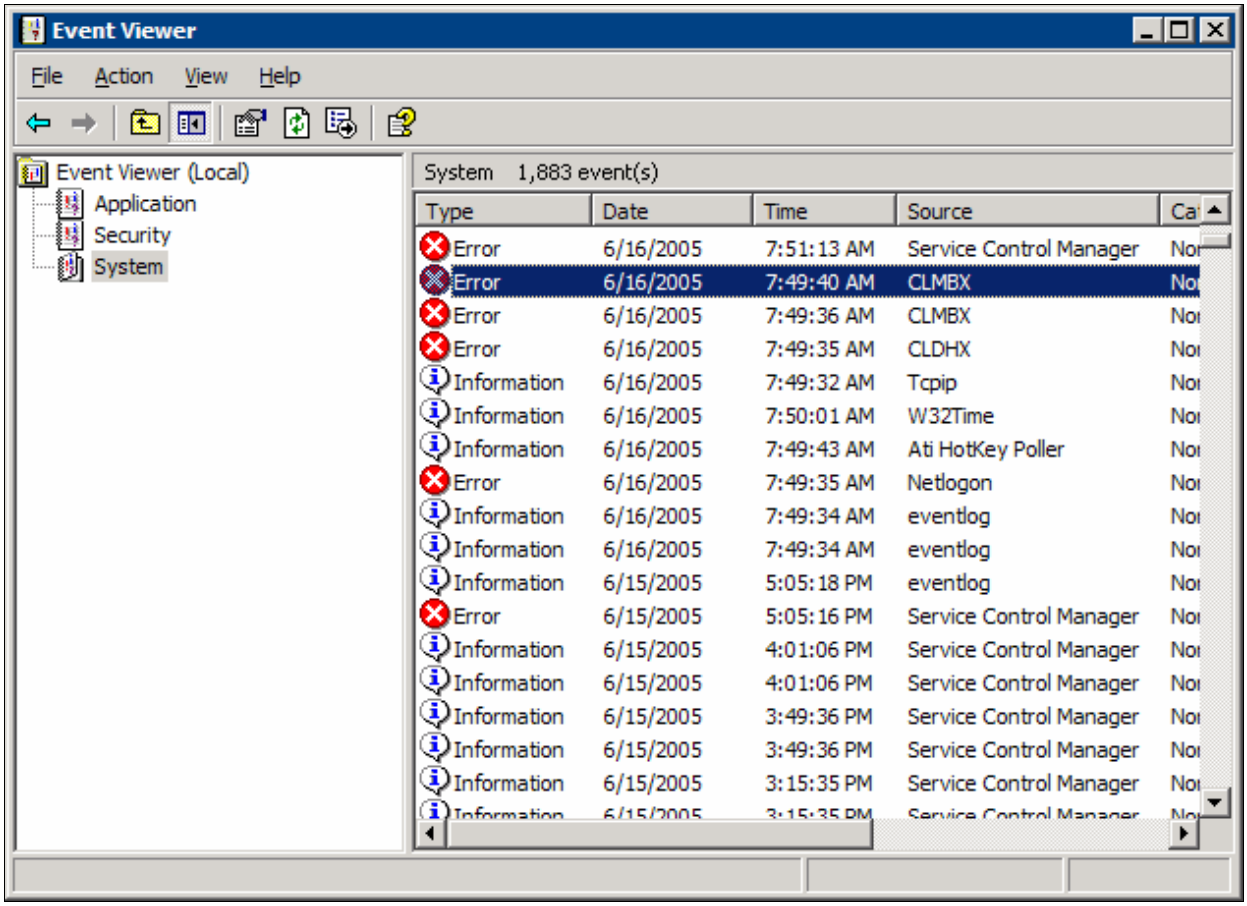
   Alternatively, an Event Viewer button is located in the Diagnostics tab of the MBX Driver Configuration Editor.



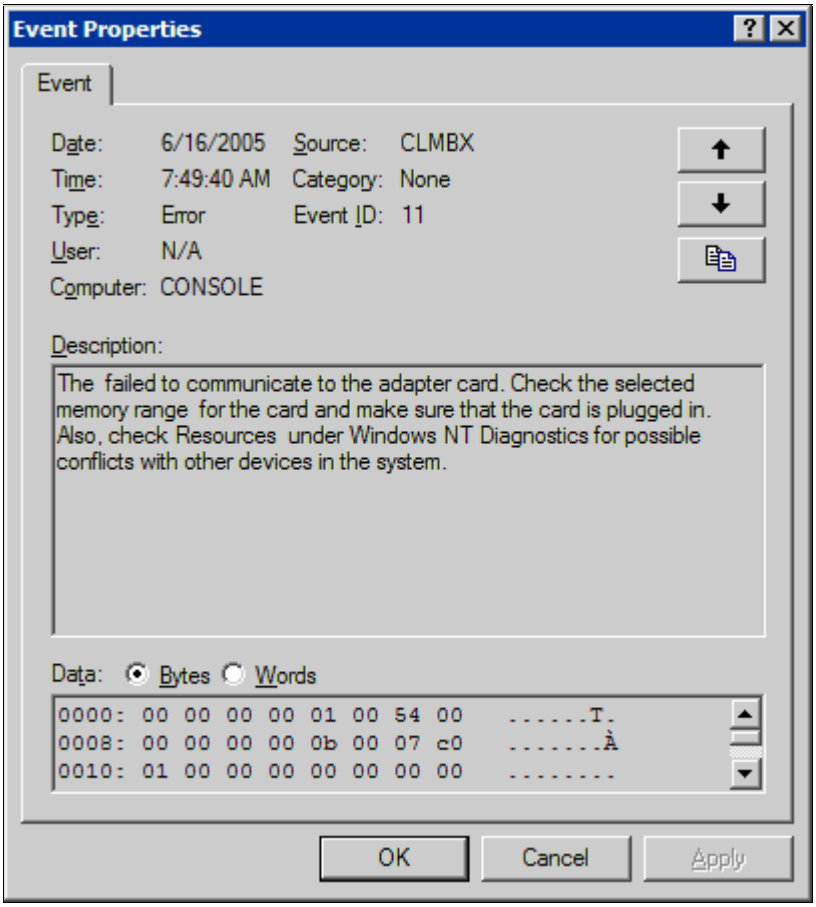
2. Select *System* from the Log menu.

3. Look for entries with *CLMBX* in the Source column.

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.

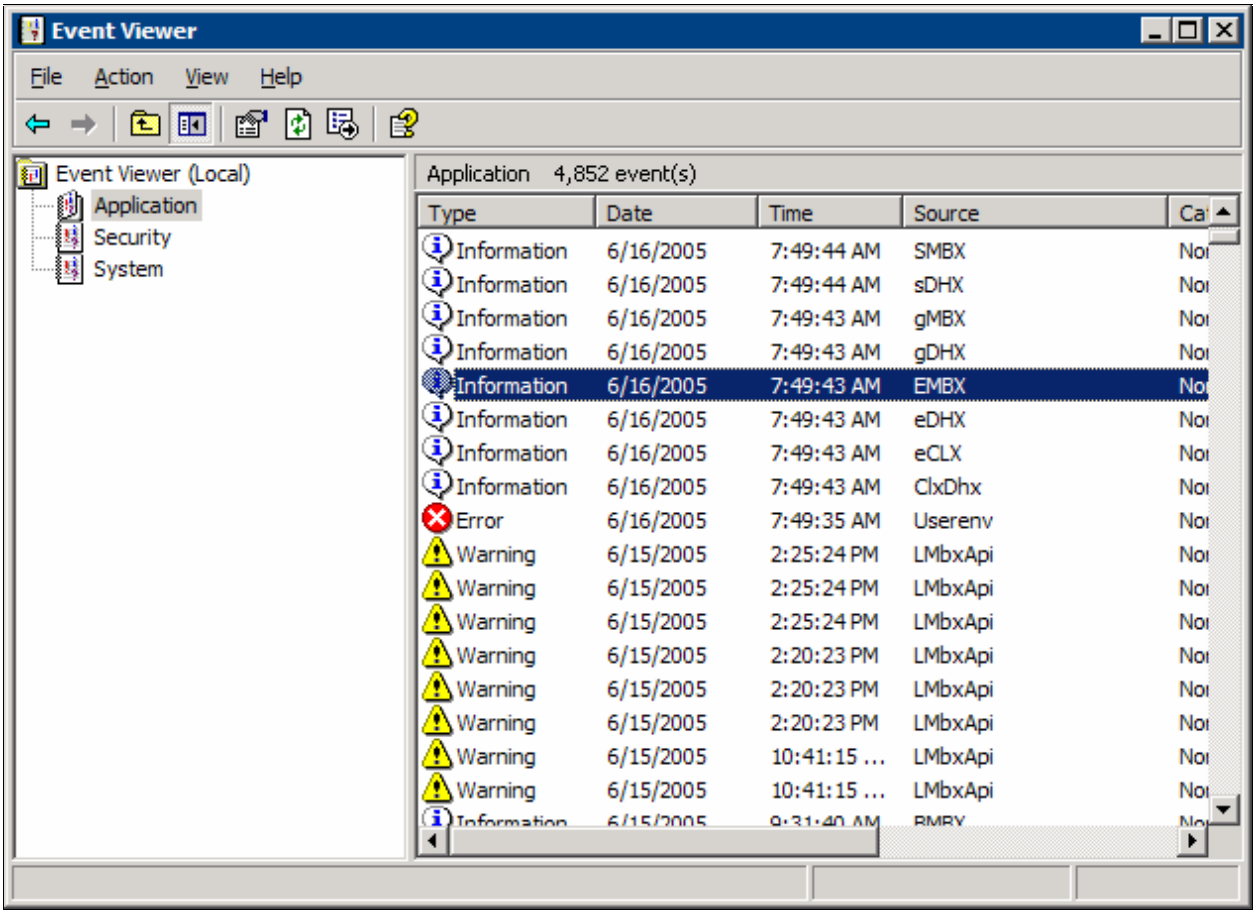


4. Double-click on the selected entry to display a complete event message as seen below.



For further descriptions of the error log messages, refer to the Help file for your specific MBX product.

5. You can also select Application events, then look for MBX entries such as sMBX and eMBX.



6. For further descriptions of the error log messages, refer to the Ethernet MBX Server Messages section.

Ethernet MBX Server Messages

<function> failed - <error text>. Data buffer contains the <data ID> data.

This is a general error message. It indicates that a problem occurred while trying to run the Ethernet MBX as a service.

Registration DLL failed to load. Reinstall the product.

This DLL should have been copied into the Windows XP/2000/NT system32 directory by the installation program. Reinstall the product.

Registration verification failed. Reinstall the product.

The registration information could not be accessed. The registration information is gathered and stored during the installation process. Reinstall the product.

EMBXAPIM.DLL failed to load. Reinstall the product.

This DLL should have been copied into the Windows XP/2000/NT system32 directory by the installation program. Reinstall the product.

Gateway server of <Server Name> name already exists! Change server name.

Only one copy of the Ethernet MBX Driver is allowed to run on a system and another copy is already running.

<function> failed - <error text>.

This is a general error message. It indicates a problem with setting up the communication between Ethernet MBX and its clients.

Cyberlogic eMBX Server service <Service Name> started.

The Ethernet MBX Driver started successfully.

Winsock version <Version Number> detected. Winsock 2.0 or greater is required.

The Ethernet MBX Driver requires Winsock 2.0 or higher. Check with Microsoft for the availability of newer versions of Winsock.

Unable to initialize Winsock.

An internal Winsock error occurred.

Unable to initialize global system resources.

The Ethernet MBX Driver was unable to allocate some of the resources that it requires. Try closing some open applications to free up more memory. If that doesn't solve the problem, it may be necessary to add more memory to the system.

The target node <IP Address> closed the connection.

This may indicate too many open connections to the target node.

Frequently Asked Questions

Installation & Configuration

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

The next step is to configure a logical device (Ethernet MBX). See the Configuration section. After this is done, run the MBX Demo to test the driver.

I've configured an Ethernet MBX device. How do I know that it's working?

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

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 Ethernet MBX device has been configured. If not, see the Typical Driver Configuration for details on setting up an Ethernet MBX device.

Cause 2: The Ethernet MBX Driver could not find the MBX device specified under Device Number. See the Typical Driver Configuration for details on finding and entering this information.

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

MBX Demo uses the device number to determine which card to use. "Set device number" lets you choose which configured Ethernet MBX device the demo will use. If you are using some other software product, contact the manufacturer for more information on using multiple devices.

Miscellaneous

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

Ethernet communications do not support Peer Cop.