



# Mbx.Net Help

---

*Mbx.Net Configuration Editor*

---

Version 1

## **MBX.NET CONFIGURATION EDITOR HELP**

### **Version 1**

Copyright © 1994-2017, Cyberlogic® Technologies Inc. All rights reserved.

This document and its contents are protected by all applicable copyright, trademark and patent laws and international treaties. No part of this document may be copied, reproduced, stored in a retrieval system or transmitted by any means, electronic, mechanical, photocopying, recording or otherwise, without the express written permission of Cyberlogic Technologies Inc. This document is subject to change without notice, and does not necessarily reflect all aspects of the mentioned products or services, their performance or applications. Cyberlogic Technologies Inc. is not responsible for any errors or omissions in this presentation. Cyberlogic Technologies Inc. makes no express or implied warranties or representations with respect to the contents of this document. No copyright, trademark or patent liability or other liability for any damages is assumed by Cyberlogic Technologies Inc. with respect to the use of the information contained herein by any other party.

Cyberlogic®, DHX®, MBX®, WinConX® and Intelligent • Powerful • Reliable® are registered trademarks and DirectAccess™, OPC Crosslink™, OPC Datacenter™, DevNet™ and C-logic™ are trademarks of Cyberlogic Technologies Inc. All other trademarks and registered trademarks belong to their respective owners.

Document last revision date October 20, 2017

## TABLE OF CONTENTS

<b>Introduction .....</b>	<b>5</b>
Compatibility.....	5
Blending MBX-Supported Networks.....	5
<b>What Should I Do Next?.....</b>	<b>8</b>
Learn How the Mbx.Net Platform Works.....	8
Read a Quick-Start Guide.....	8
Get Detailed Information on the Configuration Editor .....	8
Verify That It's Working or Troubleshoot a Problem.....	8
Get Information on Related Products .....	8
Print a Copy of This Document .....	8
Contact Technical Support .....	8
<b>Communication Using the Mbx.Net Platform.....</b>	<b>9</b>
Mbx.Net Server Connections .....	9
Communication Protocols.....	9
Named Pipes (Net.Pipe).....	9
TCP (Net.Tcp) .....	9
HTTP .....	9
Mbx.Net Examples.....	10
Using MbpStat.Net .....	10
Typical Mbx.Net Gateway Application.....	10
<b>Quick-Start Guide.....</b>	<b>12</b>
Enabling Endpoint Discovery Service.....	13
Configuring HTTP Protocol .....	13
Configuring Net.Pipe Protocol.....	14
Configuring Net.Tcp Protocol.....	15
Configuring the Startup Mode.....	17
Configuring the Firewall .....	18
Adding Server Connections .....	19
Running MbpStat.Net .....	25
<b>Configuration Editor Reference .....</b>	<b>28</b>
Client Tab.....	28
Server Connection Editor .....	30
Server Tab .....	33
Configuration Tab .....	34
Connected Clients Tab.....	36
Server Control Tab.....	38
Registration Information Tab .....	39
<b>Validation &amp; Troubleshooting .....</b>	<b>41</b>
MbpStat.Net .....	41
How to Navigate the Options .....	42
Set Routing Parameters.....	43
Monitor Network Activity.....	44
Read Global Data .....	45
Global Data Present Table .....	46
Node Active Station Table.....	46
Node Error Statistics .....	47
Token Station Table .....	47
Token Owner Work Table .....	48

Current Internal Path Transactions .....	49
Show Node Personality .....	49
Show Node Peer Cop Configuration .....	50
Show Node Peer Cop Health .....	50
Event Viewer .....	51
Frequently Asked Questions .....	52
<b>Appendix: MBX Architecture and Companion Products .....</b>	<b>54</b>
MBX Driver .....	54
Ethernet MBX Driver .....	55
Serial MBX Driver .....	55
MBX Gateway Driver .....	56
Mbx.Net Gateway Driver .....	56
Virtual MBX Driver .....	56
MBX Bridge .....	57
MBX OPC Server .....	57
MbpStat.Net .....	58
Mbx.Net Server .....	58
MBX SDK .....	59

## INTRODUCTION

The Cyberlogic Mbx.Net architecture provides a secure and reliable remote connectivity infrastructure for the Cyberlogic MBX family of products. It is also a platform for the Mbx.Net applications, such as the MbpStat.Net and the Mbx.Net Gateway Driver.

At the heart of this platform is the Mbx.Net Server. This is a Windows Communication Foundation (WCF) server that supports the Http, Net.Tcp and Net.pipe protocols. The Mbx.Net client applications connect securely to the server, which in turn exposes the functionality of all MBX devices configured on the server node. The client applications and the server can be on the local network or connected through the Internet.

The Mbx.Net server and client components are part of Cyberlogic's Mbx.Net Suite and Mbx.Net Premier Suite. The Mbx.Net Server and the Mbx.Net Gateway Driver also require one of the MBX suites to be installed. However, the MbpStat.Net does not need any of the MBX suites to operate.

Only the Mbx.Net Server is a licensed component that requires activation. The client components do not require activation and can freely be installed on any system.

## Compatibility

The Cyberlogic Mbx.Net platform does not replace, but rather complements the Cyberlogic MBX architecture. The Mbx.Net Suite and Mbx.Net Premier Suite can work with any version of the MBX products, as long as they are installed on the supported Windows platforms.

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 also run on 32-bit and 64-bit Windows, and it does not require the Virtual MBX Driver.

The Mbx.Net Gateway Driver provides similar functionality as the MBX Gateway Driver. However, the Mbx.Net Gateway Driver uses a newer and more secure infrastructure. It also supports more communication protocols.

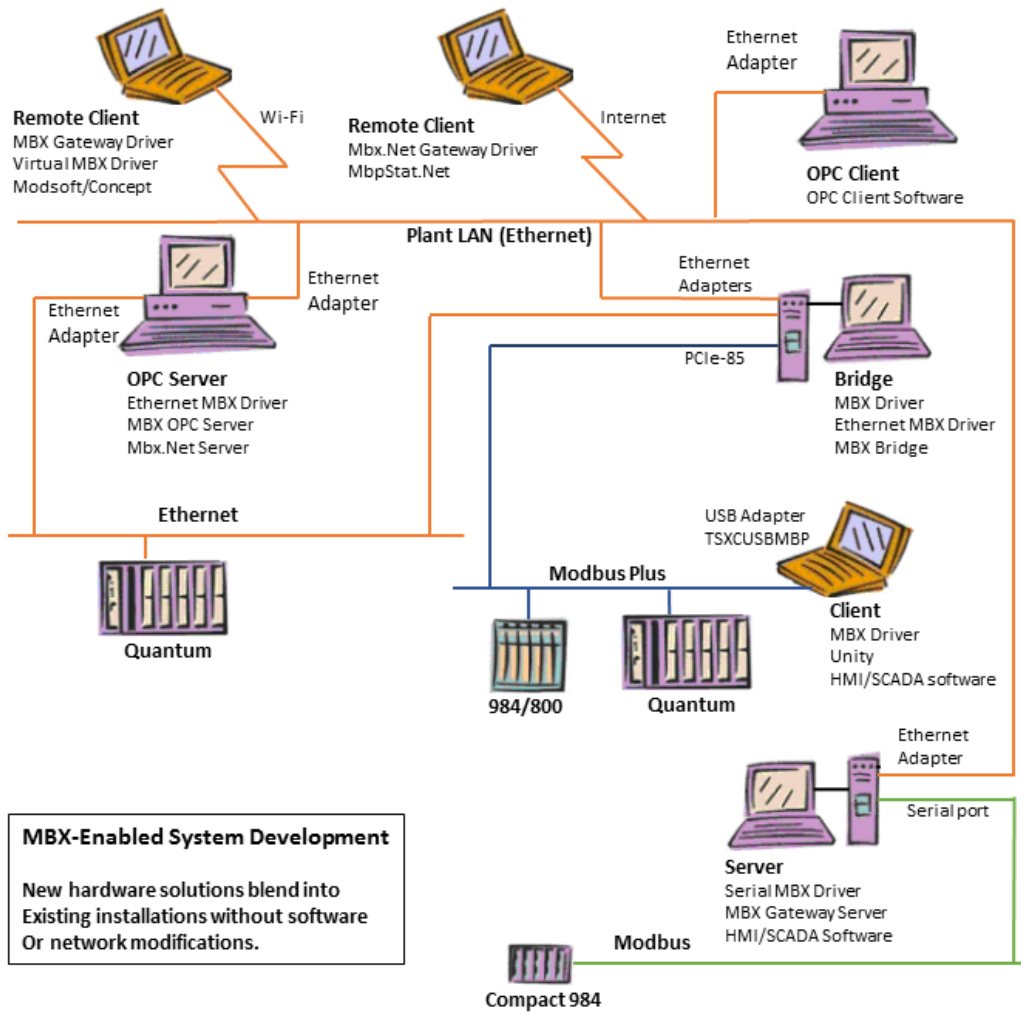
## Blending MBX-Supported Networks

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 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.
- [MbxStat.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 needed to configure, use and troubleshoot the Mbx.Net components.

### Learn How the Mbx.Net Platform Works

If you are not familiar with the Mbx.Net architecture, you should begin by reading [Communication Using the Mbx.Net Platform](#).

### Read a Quick-Start Guide

First-time users of an Mbx.Net-based product will want to read the [Quick-Start Guide](#), which walks through a typical configuration session, step-by-step.

### Get Detailed Information on the Configuration Editor

Experienced users who want specific information on features of the configuration editor will find it in the [Configuration Editor Reference](#) section.

### Verify That It's Working or Troubleshoot a Problem

If you have already completed the Mbx.Net configuration, 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 the product suite you have installed. From there, select **Product Information**.



## COMMUNICATION USING THE MBX.NET PLATFORM

At the heart of the Mbx.Net platform is the Mbx.Net Server. This is a Windows Communication Foundation (WCF) server that supports the Http, Net.Tcp and Net.pipe protocols. The Mbx.Net client applications connect securely to the server, which in turn exposes the functionality of all MBX devices configured on the server node. The client applications and the server can be on the local network or connected through the Internet.

### Mbx.Net Server Connections

The Mbx.Net client applications (e.g. MbpStat.Net, Mbx.Net Gateway Driver) connect to the Mbx.Net Server through communication conduits called Mbx.Net Server Connections. Each Server Connection has a user-assigned name and a server endpoint that defines the server location and the selected protocol details. Once configured, the client applications select which Server Connection to use by its name.

On systems with the Mbx.Net Server installed, a default Server Connection called ***My Computer (net.pipe)*** is already configured. This Server Connection allows connection to the local Mbx.Net Server.

### Communication Protocols

The Mbx.Net Server supports three transport protocols: Named Pipes, TCP, and HTTP.

#### Named Pipes (Net.Pipe)

This transport protocol allows communication to the local Mbx.Net Server, and cannot be used to connect to the remote servers. A typical example for using this protocol would be when running the MbpStat.Net on the server system.

#### TCP (Net.Tcp)

TCP is a reliable delivery service with end-to-end error detection and correction. Also, it allows connections to be authenticated with a user name and password. This protocol should be used when connecting to the Mbx.Net Server over a standard Ethernet LAN.

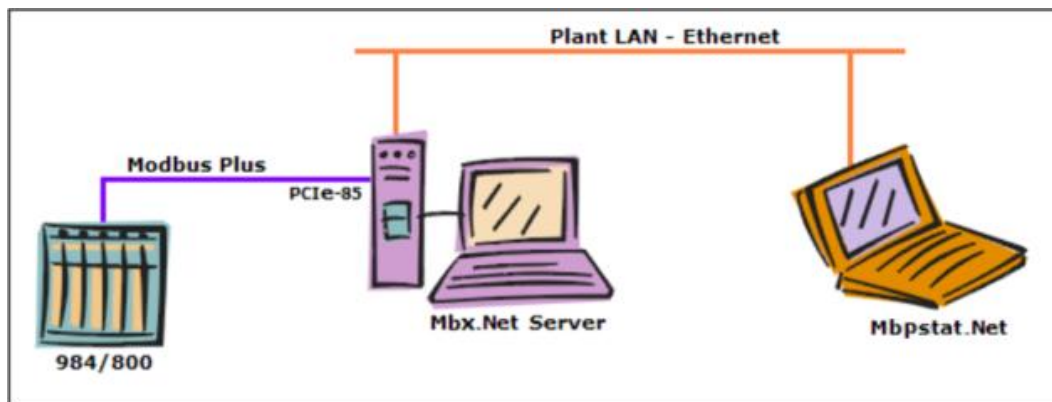
#### HTTP

HTTP is the underlying protocol used on the Internet. This protocol should typically be used when the Mbx.Net Server runs on a system that is outside of your local network and is accessible over the Internet.

## Mbx.Net Examples

### Using MbpStat.Net

The figure below shows a simple example for how the MbpStat.Net diagnostic utility can be used.

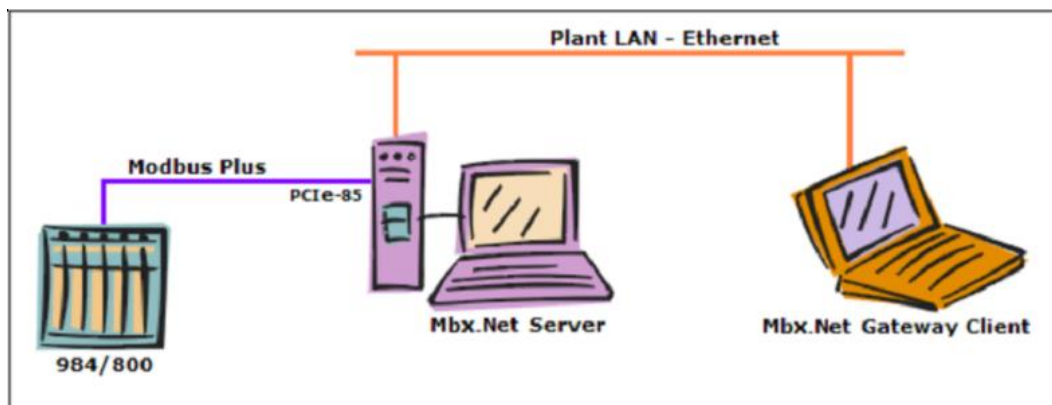


In this setup, the tower system uses the MBX Driver and a PCIe-85 card to communicate over Modbus Plus to the 984 controller. The tower is also running the Mbx.Net Server, making it the server system.

The MbpStat.Net runs on the notebook system and connects to the Mbx.Net Server over a standard Ethernet LAN. Notice that the MbpStat.Net does not require the MBX Driver or the Mbx.Net Gateway Driver to run.

### Typical Mbx.Net Gateway Application

The figure below shows a typical application of the Mbx.Net Gateway Driver.



In this setup, the tower system uses the MBX Driver and a PCIe-85 card to communicate over Modbus Plus to the 984 controller. The tower is also running the Mbx.Net Server, making it the server system.

The notebook connects to the tower over a standard Ethernet LAN, and it intends to communicate with the 984. Rather than installing a Modbus Plus interface adapter and running additional cable to connect directly to the Modbus Plus network, the notebook can simply become a client system by running the Mbx.Net Gateway Driver.

To make the connection, you would configure an Mbx.Net Gateway device on the notebook, designating the tower system as the server and selecting the tower's MBX device that corresponds to its PCIe-85 card. Applications on the notebook can now access the Mbx.Net Gateway device, which will function just as though the notebook was directly connected to the Modbus Plus through the PCIe-85.

**Note**

The Mbx.Net Gateway Driver works with all networks supported by the MBX family of drivers, not just Modbus Plus. If the server is connected to controllers over Modbus or Modbus TCP, it can also make these connections available to client systems.

The above example is a very simple, basic setup. More complex architectures are possible. For example:

- The server can have more than one device that it makes available to client systems. These devices can be of any mixture of adapters or network types.
- Multiple client systems can obtain access through each server and through each device on a server.
- A client can connect to more than one device on a server. To do this, simply create additional Mbx.Net Gateway Driver devices on the client and associate each with the desired device on the server.
- A client can connect to more than one server. Again, this is achieved by creating additional Mbx.Net Gateway Driver devices, one for each desired device on each server.

## QUICK-START GUIDE

The Mbx.Net Configuration Editor is a common component of all products in the Mbx.Net family. It is used to create the Mbx.Net Server Connections, and to configure and monitor the Mbx.Net Server.

The following guide shows a typical configuration session. Use it only as a guideline of how to configure the most common features. For detailed descriptions of all available features, refer to the [Configuration Editor Reference](#) section.

During the Mbx.Net product installation, the installer allows you to install either the client applications (e.g. MbpStat.Net and Mbx.Net Gateway) or the Mbx.Net Server, or both. The required configuration may vary depending on which components have been installed. This guide assumes that all client and server components have been installed.

In case you decide to skip some sections, this guide has been divided into three main procedures: Server-Side Configuration, Client-Side Configuration, and Verifying Your Configuration. Each procedure is broken into several short segments. Follow the segments that are relevant to you.

### ***Server-Side Configuration***

- [Enabling Endpoint Discovery Service](#)
- [Configuring HTTP Protocol](#)
- [Configuring Net.Pipe Protocol](#)
- [Configuring Net.Tcp Protocol](#)
- [Configuring the Startup Mode](#)
- [Configuring the Firewall](#)

### ***Client-Side Configuration***

- [Adding Server Connections](#)

### ***Verifying Your Configuration***

- [Running MbpStat.Net](#)

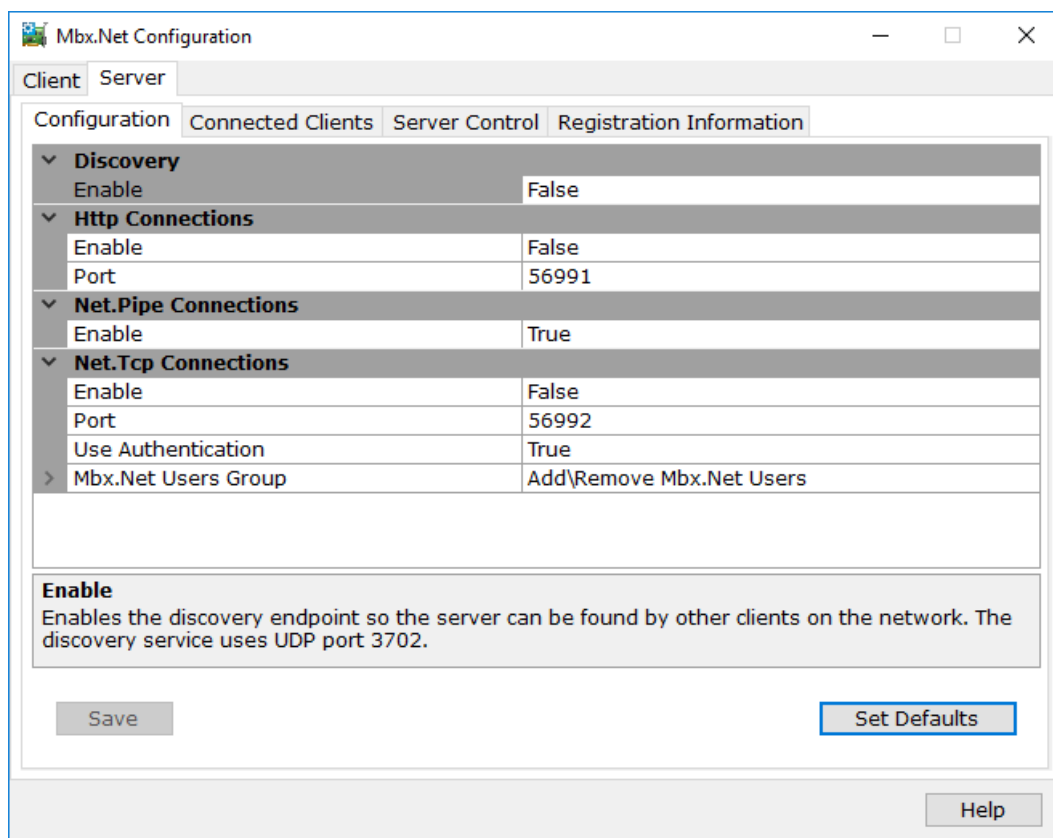
After completing this procedure, you will have a fully-configured Mbx.Net Server and the Server Connections.

To begin, go to [Enabling Endpoint Discovery Service](#).

## Enabling Endpoint Discovery Service

The Mbx.Net Server supports the WS-Discovery protocol. When enabled, the server can announce the availability of the supported protocols over the local network. The discovery service uses UDP port 3702. By default, the Discovery service is disabled. The following procedure shows you how to enable it.

1. From the Windows **Start** menu, go to the **Cyberlogic Mbx.Net Suites** submenu and select **Configuration**. Next, click on the **Server** tab.

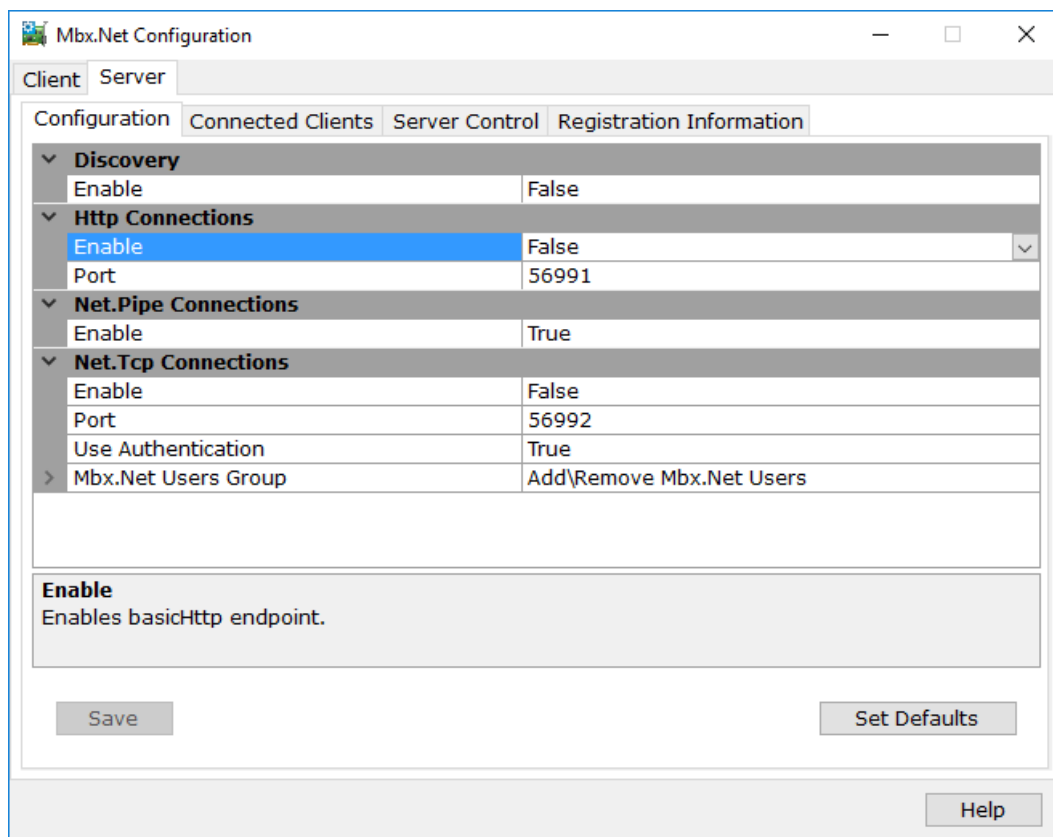


2. In the **Discovery** section, select the **Enable** field.
3. If you want to enable the Discovery service, change the **Enable** field from False to True.
4. Click the **Save** button to save the changes.

For the next part of the procedure, go to the [Configuring HTTP Protocol](#) section.

## Configuring HTTP Protocol

HTTP is the underlying protocol used on the Internet. This protocol should typically be used when connecting to the Mbx.Net Server over the Internet. By default, the HTTP protocol is disabled. The following procedure shows you how to enable it.

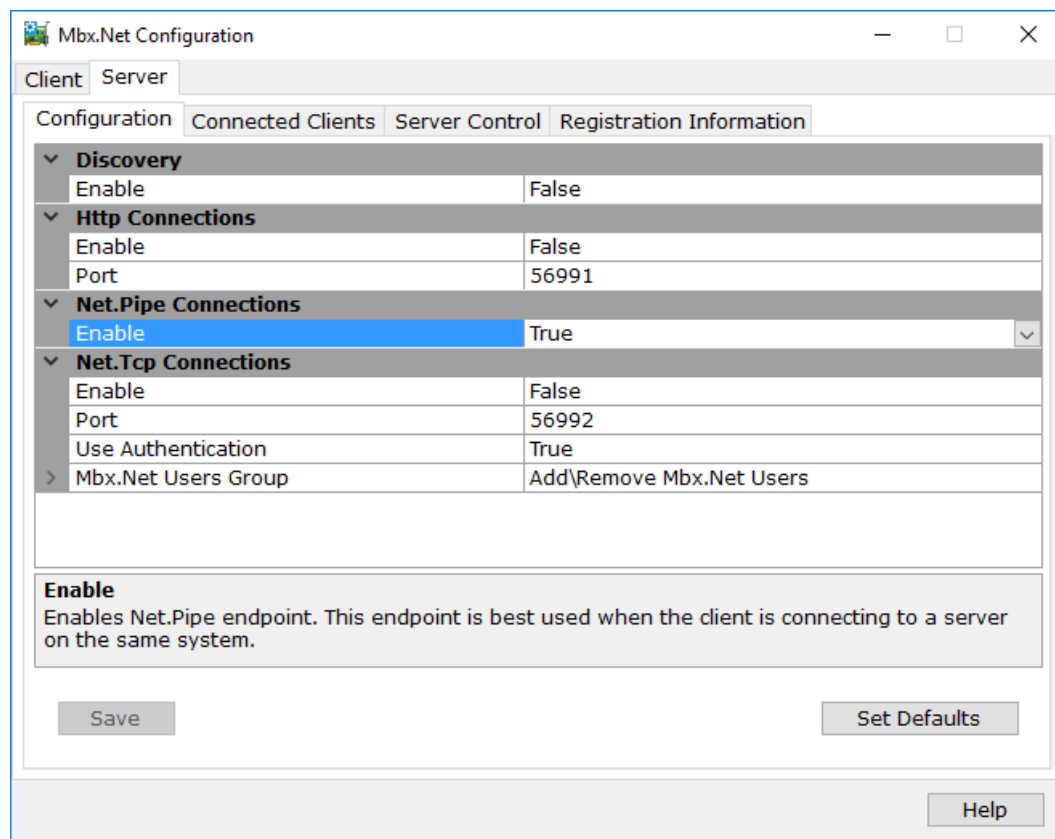


1. In the **HTTP Connections** section, select the **Enable** field.
2. If you want to enable the HTTP protocol, change the **Enable** field from False to True.
3. Enter the **Port** to be used with the HTTP connections. The default port number is 56991.
4. Click the **Save** button to save the changes

To continue, proceed to the [Configuring Net.Pipe Protocol](#) section.

## Configuring Net.Pipe Protocol

This transport protocol allows communication to the local Mbx.Net Server and cannot be used to connect to the remote servers. A typical example for using this protocol would be when running the MbpStat.Net on the server system. By default, the Net.Pipe protocol is enabled, and most users should leave it enabled. The following procedure shows you how to disable it, but we recommend skipping it.

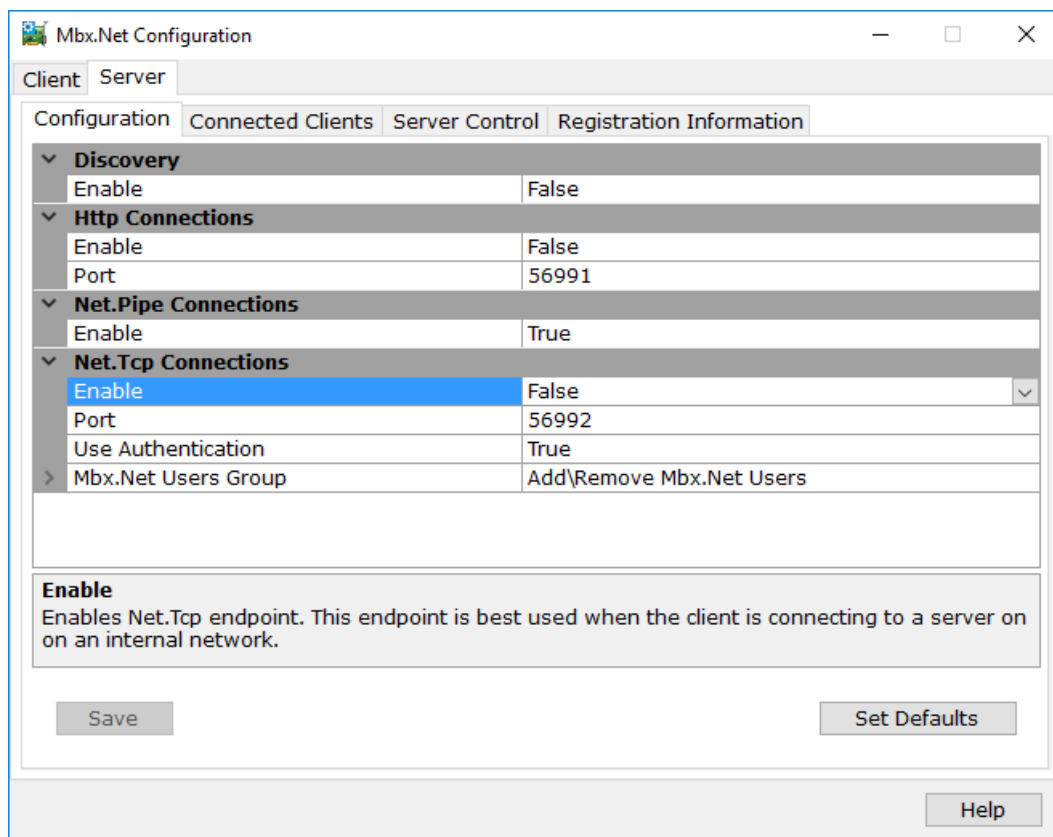


1. In the **Net.Pipe Connections** section, select the **Enable** field.
2. If you want to disable the Net.Pipe protocol, change the **Enable** field from True to False.
3. Click the **Save** button to save the changes.

To continue, proceed to the [Configuring Net.Tcp Protocol](#) section.

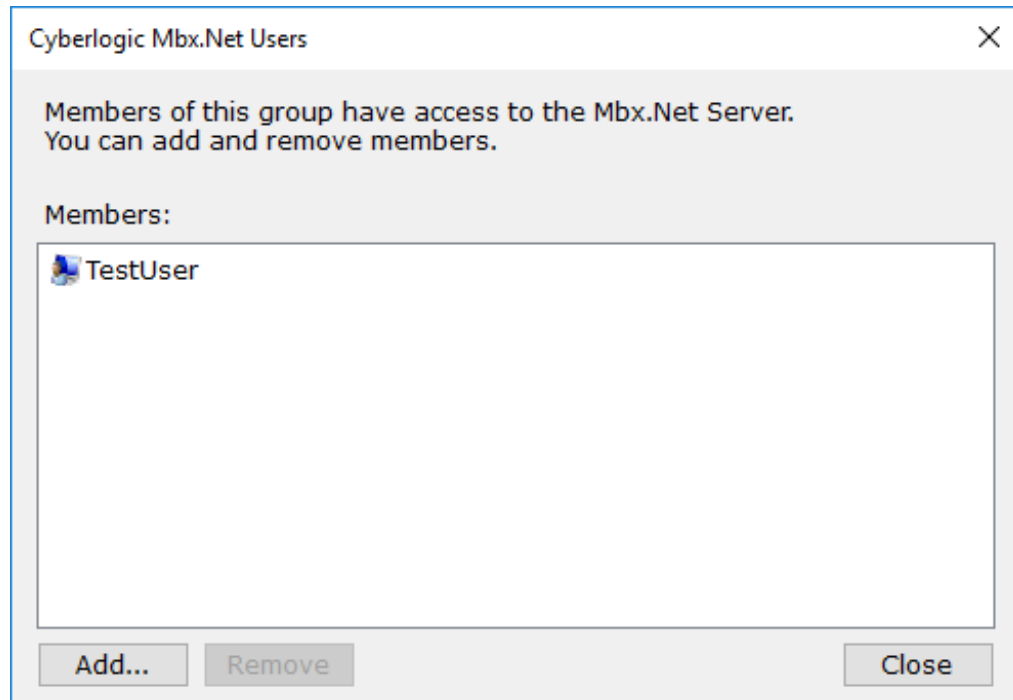
## Configuring Net.Tcp Protocol

TCP is a reliable delivery service with end-to-end error detection and correction. Also, it allows connections to be authenticated with a user name and password. This protocol should be used when connecting to the Mbx.Net Server over a standard Ethernet LAN. By default, the Net.Tcp protocol is disabled. The following procedure shows you how to enable it.



1. In the **Net.Tcp Connections** section, select the **Enable** field.
2. If you want to enable the Net.Tcp protocol, change the **Enable** field from False to True.
3. Enter the **Port** to be used with the Net.Tcp connections. The default port number is 56992.
4. Make sure that the **Use Authentication** field is set to True. This will enable the user authentication.
5. Select the **Mbx.Net Users Group** field. Here you can set a group of users that are authorized to connect to the Mbx.Net Server. The name of this group is **Cyberlogic Mbx.Net Users**. Click the edit button to edit this group.





6. By default, the user that installed the Mbx.Net Suite is already part of the group. Click the **Add...** button to add more members to the group. Click **Close** when finished.
7. Click the **Save** button to save the changes.

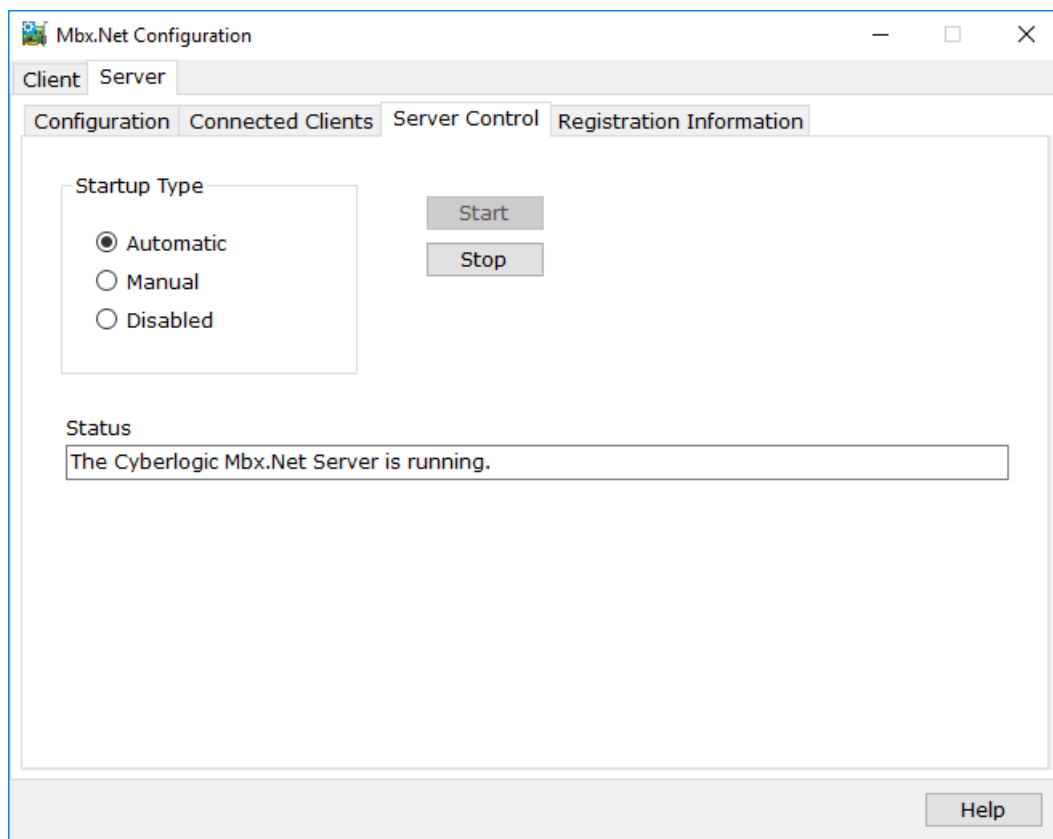
To continue, proceed to the [Configuring the Startup Mode](#) section.

## Configuring the Startup Mode

By default, the Mbx.Net Server starts automatically when Windows boots. Other options include Manual or Disabled startup.

In Manual startup mode, the Mbx.Net Server will not start when Windows boots, but you can control it manually using the Start and Stop buttons.

When the startup mode is set to Disabled, the Mbx.Net Server will not run.



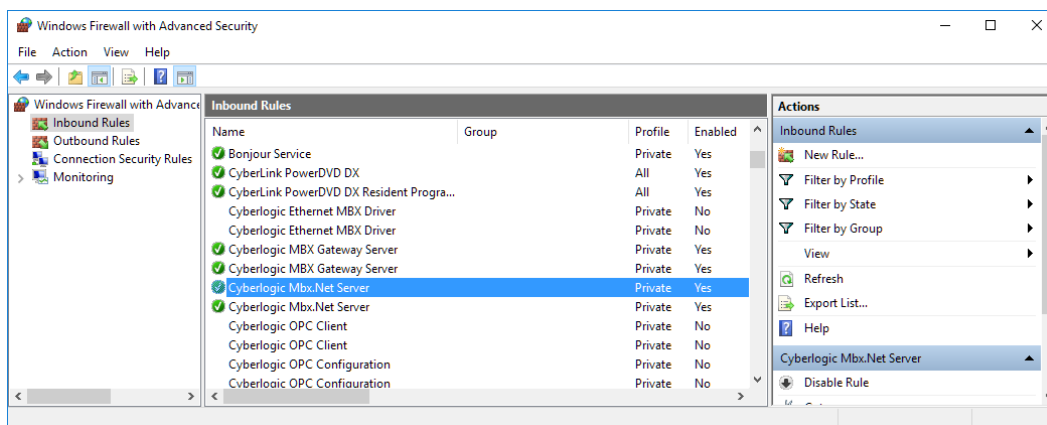
1. Click on the **Server Control** tab.
2. Select the **Startup Mode**. We recommend using the Automatic (default) startup mode.

To continue, proceed to the [Configuring the Firewall](#) section.

## Configuring the Firewall

If your system uses a firewall, you must configure it to permit Mbx.Net Server communication. The procedure shown here is for the Windows 10 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 the Windows 10 firewall, go to the **Control Panel** and open **Windows Firewall** and select **Advanced Settings**.
2. Select **Inbound Rules**.



3. Verify that the rules for **Cyberlogic Mbx.Net Server** are enabled. If not, enable them.
4. Close the window to exit.

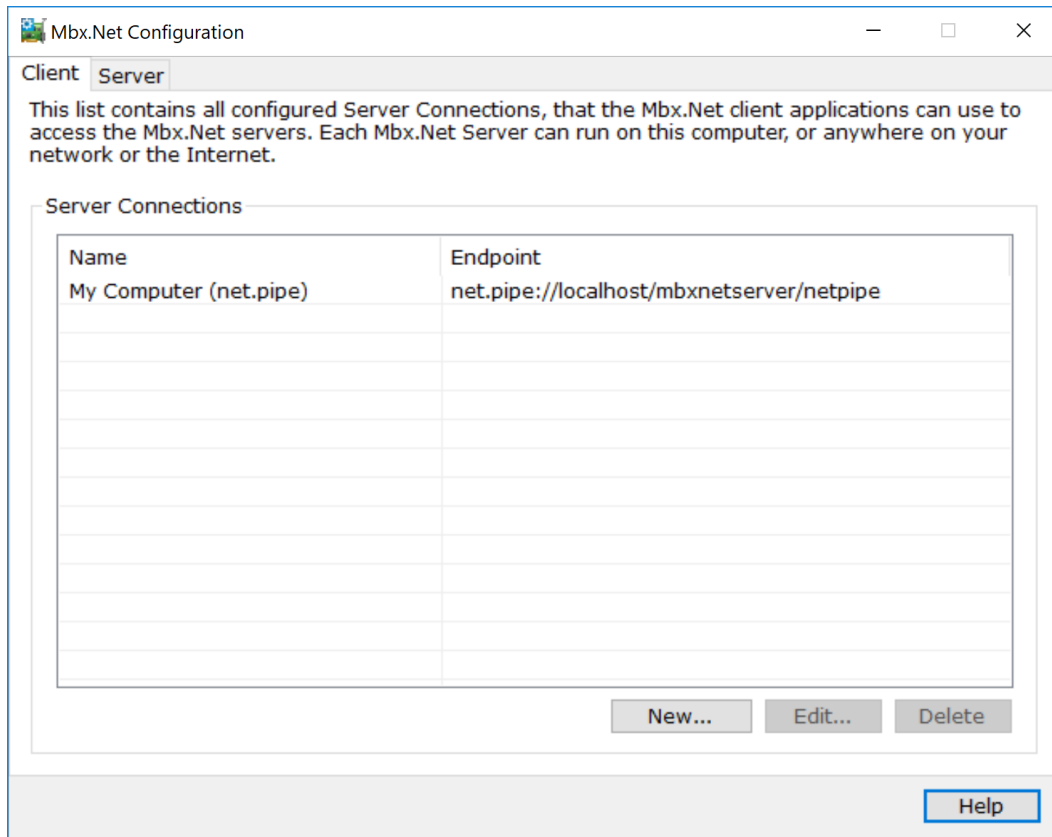
To continue, proceed to the [Adding Server Connections](#) section.

## Adding Server Connections

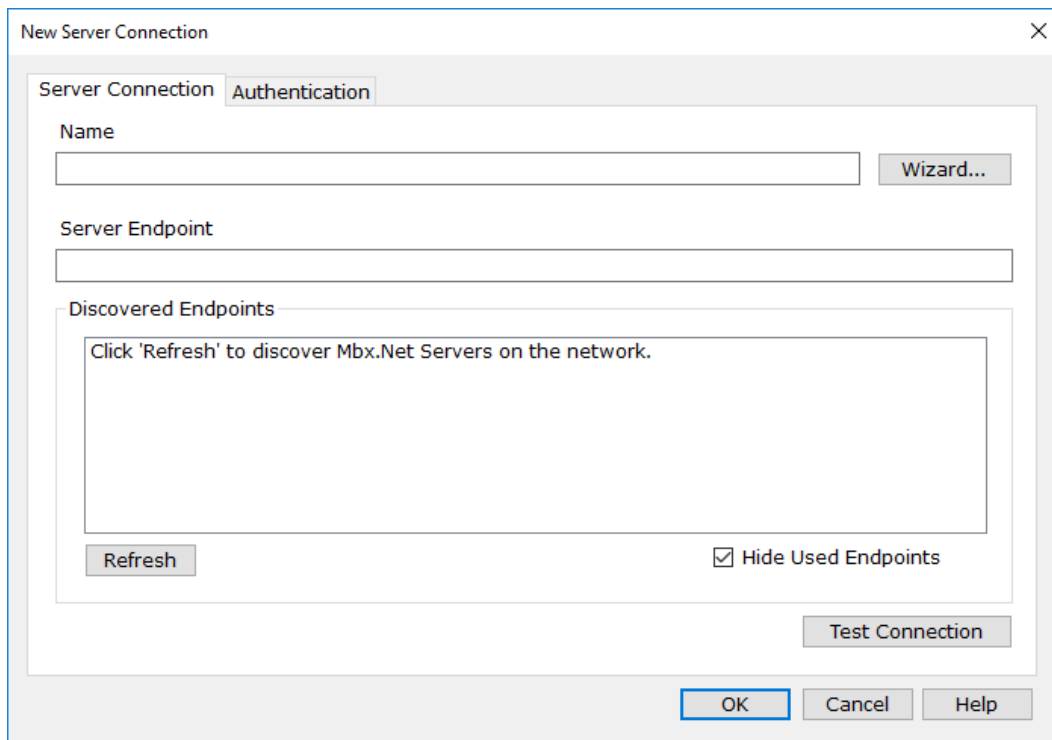
The Mbx.Net client applications (e.g. MbpStat.Net, Mbx.Net Gateway Driver) connect to the Mbx.Net Server through communication conduits called Mbx.Net Server Connections. Each Server Connection has a user-assigned name and a server endpoint that defines the server location and the selected protocol details. Once configured, the client applications select which Server Connection to use by its name.

On systems with the Mbx.Net Server installed, a default Server Connection called **My Computer (net.pipe)** is already configured. This Server Connection allows connection to the local Mbx.Net Server. The following procedure shows how to create a new Server Connection using the Net.Tcp protocol.

1. From the Windows **Start** menu, go to the **Cyberlogic Mbx.Net Suites** submenu and select **Configuration**. Make sure the **Client** tab is selected.

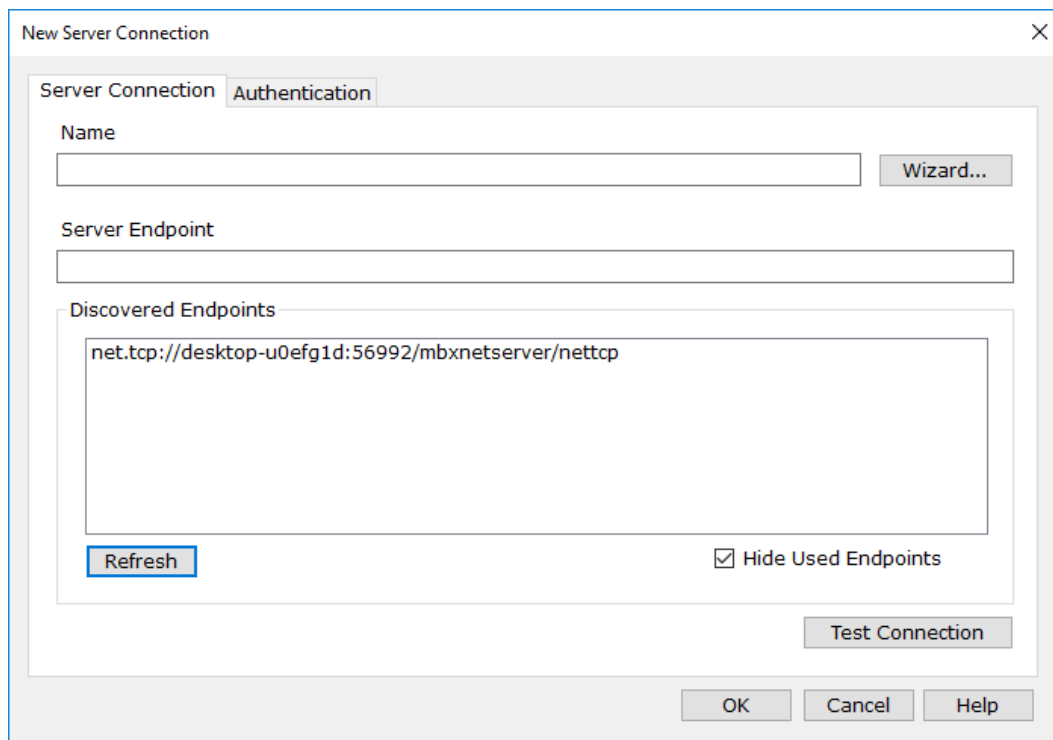


2. Click the **New...** button.



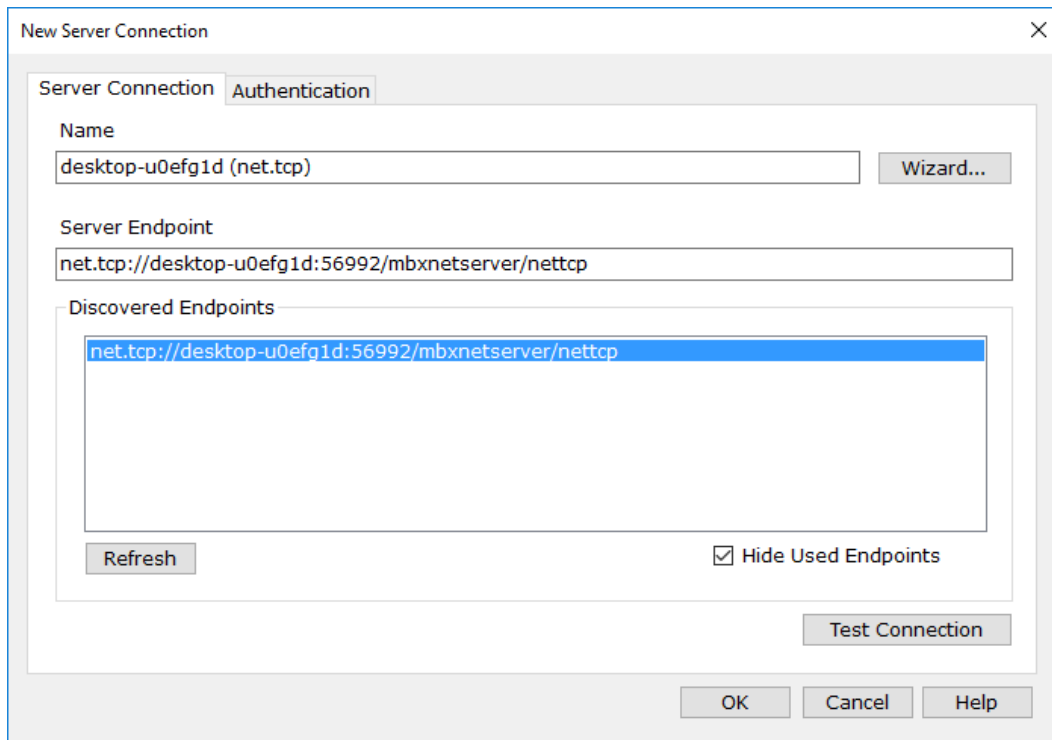
When the Server Connection editor is launched for the first time, the discovered endpoints list is empty.

3. Click the **Refresh** button to discover available endpoints from remote Mbx.Net Servers.

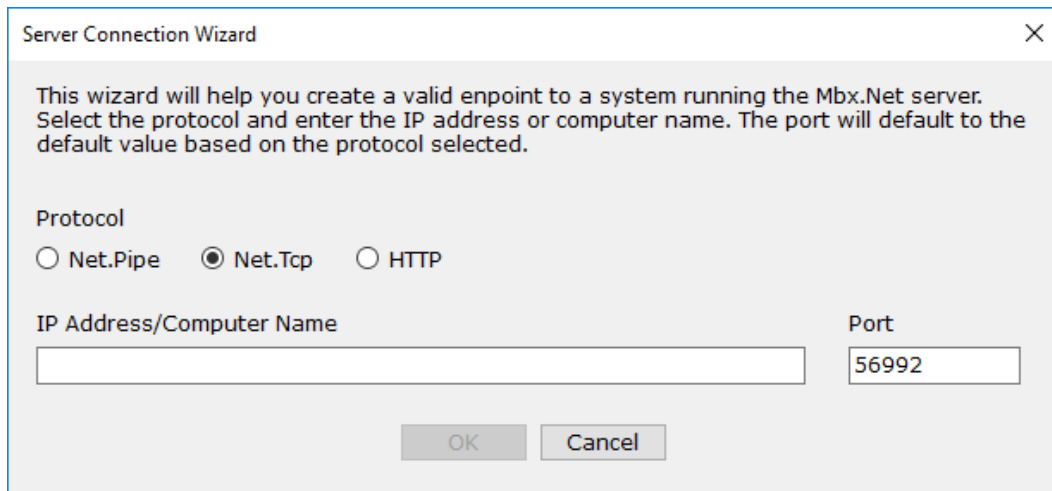


In our case, we discovered one endpoint, but you may have more. If **Refresh** doesn't find any endpoints, make sure that the Discovery service at each remote Mbx.Net Server is enabled. For information on how to do that, go to [Enabling Endpoint Discovery Service](#).

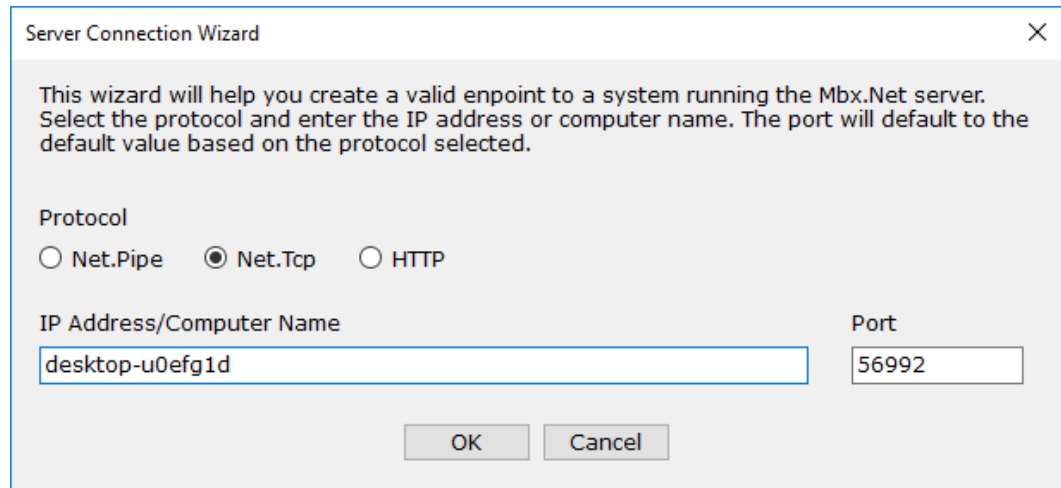
4. If **Refresh** did not find any endpoints, go to step 6 to see how to set the endpoint manually.
5. Select an endpoint that starts with "net.tcp" from the **Discovered Endpoints** list. Go to step 9 to continue the procedure.



6. Click the **Wizard** button.



7. Select Net.Tcp protocol and enter the name of the remote computer that runs the Mbx.Net Server.



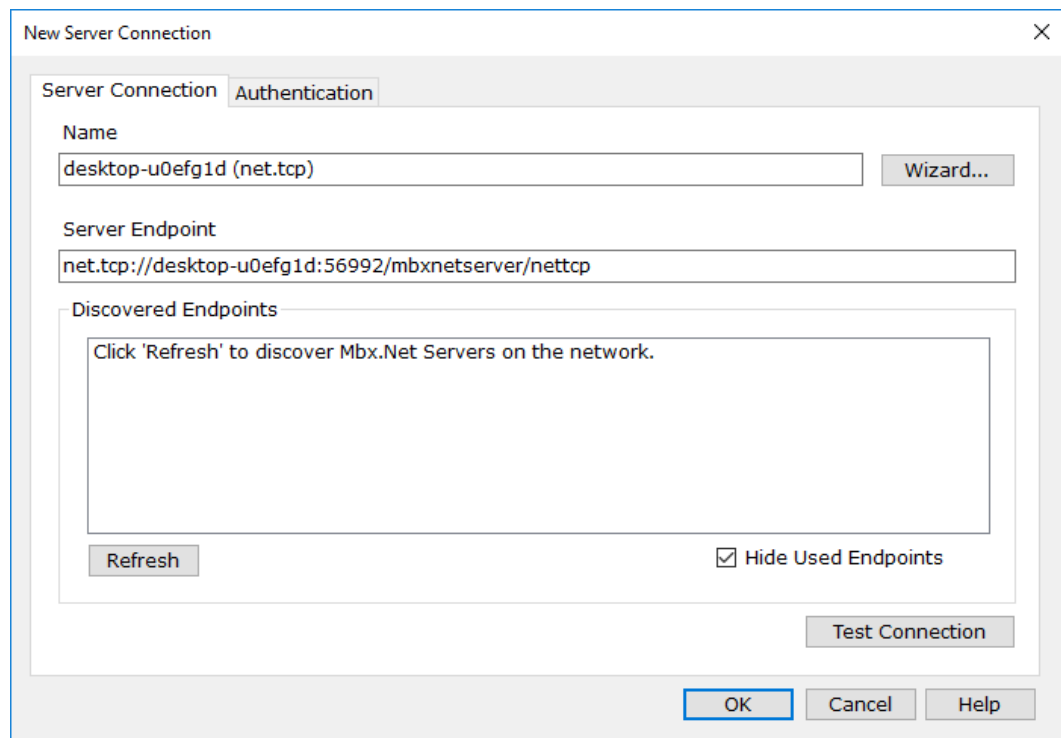
The screenshot shows the 'Server Connection Wizard' dialog box. It has a title bar with a close button (X). The main text reads: 'This wizard will help you create a valid endpoint to a system running the Mbx.Net server. Select the protocol and enter the IP address or computer name. The port will default to the default value based on the protocol selected.'

Under the heading 'Protocol', there are three radio buttons: 'Net.Pipe' (unselected), 'Net.Tcp' (selected), and 'HTTP' (unselected).

Below this, there are two input fields. The first is labeled 'IP Address/Computer Name' and contains the text 'desktop-u0efg1d'. The second is labeled 'Port' and contains the text '56992'.

At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

8. Click OK to exit the wizard.



The screenshot shows the 'New Server Connection' dialog box. It has a title bar with a close button (X). The dialog is divided into two tabs: 'Server Connection' (selected) and 'Authentication'.

Under the 'Server Connection' tab, there are three main sections:

- Name:** A text box containing 'desktop-u0efg1d (net.tcp)' and a 'Wizard...' button to its right.
- Server Endpoint:** A text box containing 'net.tcp://desktop-u0efg1d:56992/mbxnetserver/nettcp'.
- Discovered Endpoints:** A large text area containing the instruction 'Click 'Refresh' to discover Mbx.Net Servers on the network.' Below this area is a 'Refresh' button and a checked checkbox labeled 'Hide Used Endpoints'.

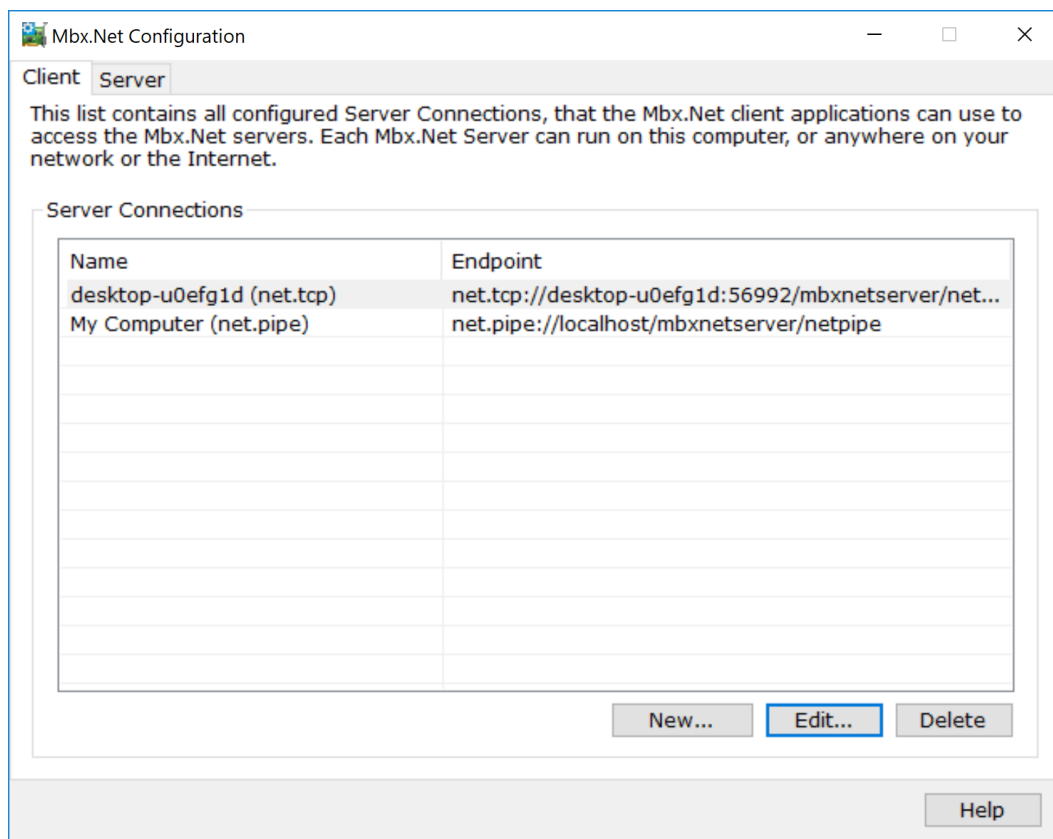
At the bottom right of the dialog is a 'Test Connection' button. At the very bottom are three buttons: 'OK' (highlighted with a blue border), 'Cancel', and 'Help'.

9. Select the **Authentication** tab.

The screenshot shows a dialog box titled "New Server Connection" with a close button (X) in the top right corner. It has two tabs: "Server Connection" and "Authentication". The "Authentication" tab is active and contains the following text: "Endpoints that are configured to use net.tcp protocols can use authentication to ensure the identity of the user making the connection. If the Mbx.Net server you are connecting to requires authentication, you can enter the user name\password here. If you do not enter the user name\password here you will be prompted at runtime when the connection is made." Below this text is a checkbox labeled "Use Authentication" which is currently unchecked. Underneath the checkbox are two text input fields: "User Name" and "Password". At the bottom right of the dialog box are three buttons: "OK", "Cancel", and "Help".

10. If the selected endpoint requires authentication, check the ***Use Authentication*** checkbox, and enter the ***User Name*** and ***Password***. We strongly recommend using authentication. Next, click ***OK*** to finish.





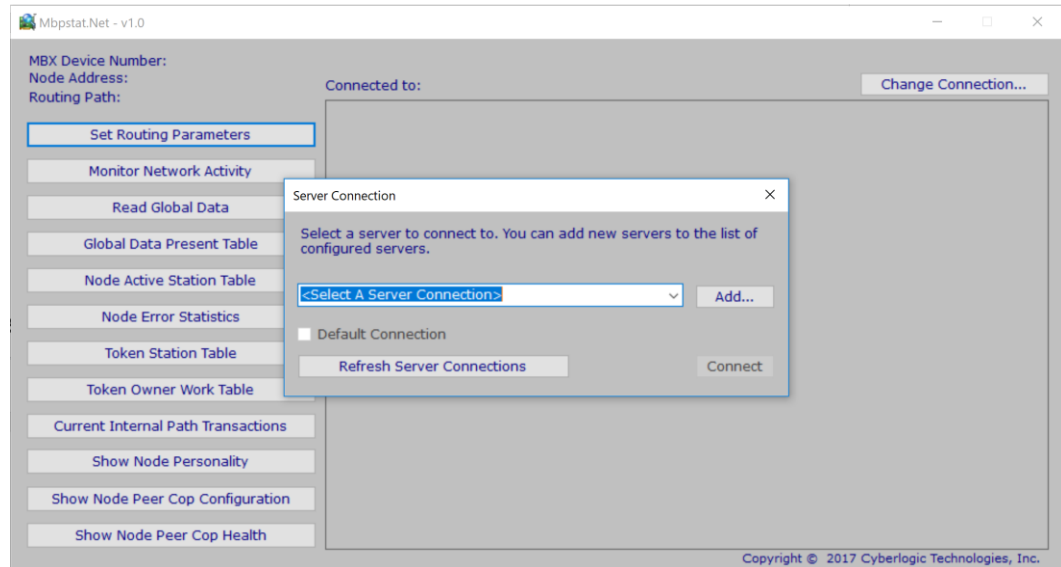
11. Repeat this procedure for each Server Connection that you want to use.

To continue, proceed to the [Running MbpStat.Net](#) section.

## Running MbpStat.Net

The MbpStat.Net utility 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.

1. From the Windows **Start** menu, locate the **Cyberlogic Mbx.Net Suites**. From there, go to the **Diagnostics** submenu and select **MbpStat.Net**.



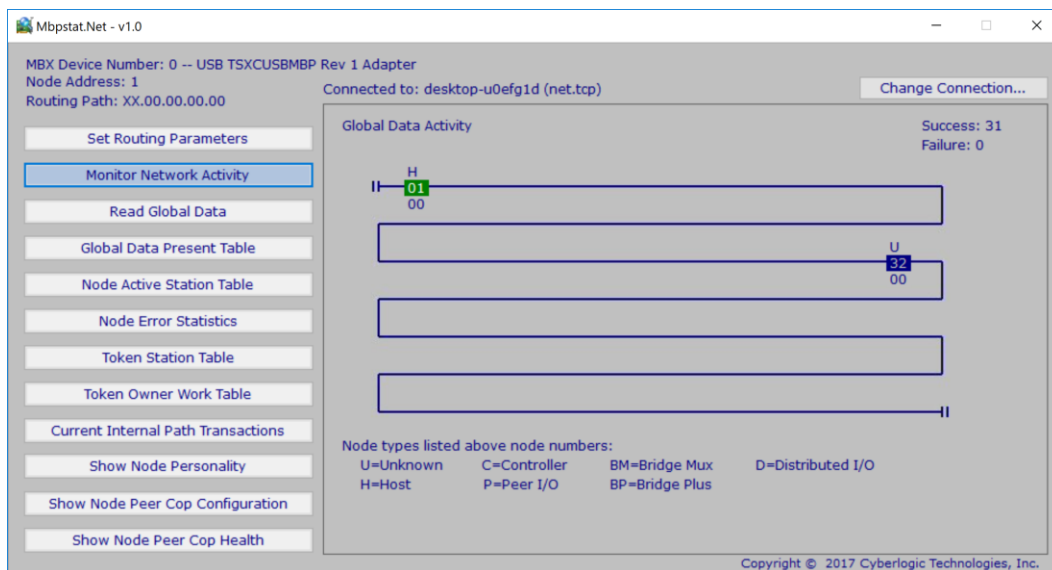
2. Select the **Server Connection** and click the **Connect** button.



3. Enter the MBX Device number that you want to use and then press **Enter**.



4. Enter the routing path of the Modbus Plus node you want to monitor. The last routing byte should be zero.



5. Use the mouse or the **Up/Down** and **Enter** keys to select the various diagnostics from the menu. All diagnostics provide the same information as the original DOS-based MBPSTAT application.

## CONFIGURATION EDITOR REFERENCE

The Mbx.Net Configuration Editor is a common component of all products in the Mbx.Net family. It is used to create the Mbx.Net Server Connections, and to configure and monitor the Mbx.Net Server.

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](#).

The Mbx.Net Configuration Editor consists of two tabs:

- [Client Tab](#)
- [Server Tab](#)

The following sections provide complete descriptions of these tabs.

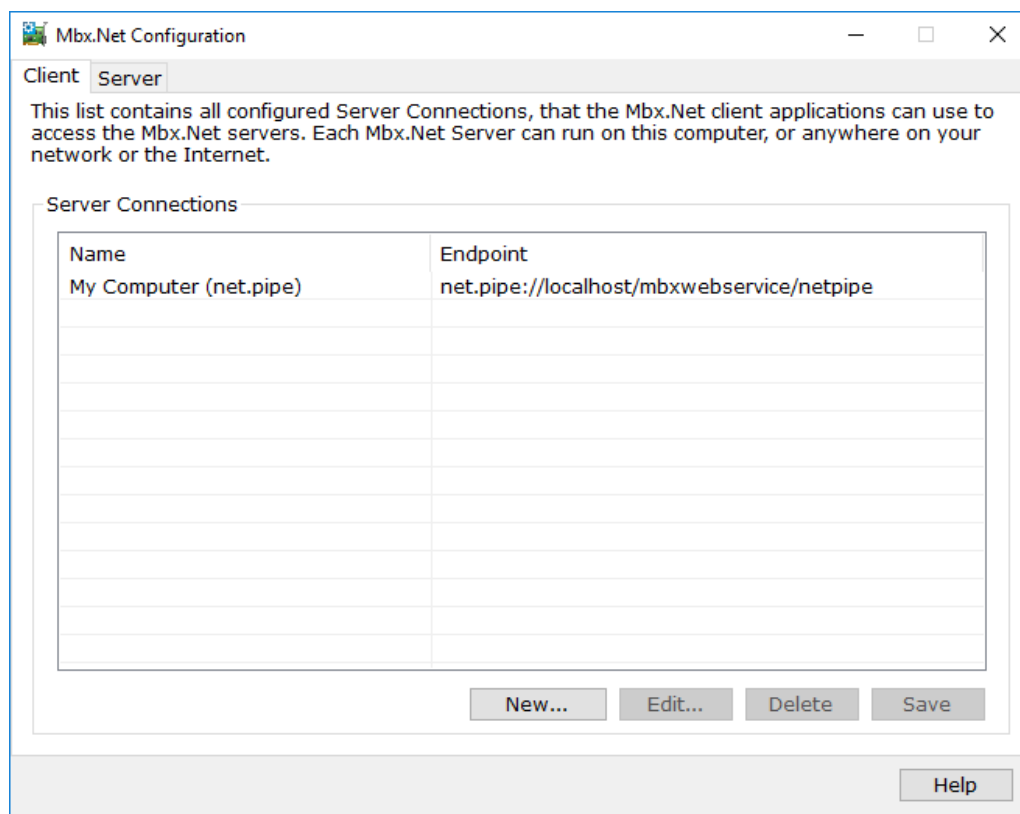
### Client Tab

**Note**

The Client Tab is only available on systems that have the Mbx.Net clients (MbpStat.Net and Mbx.Net Gateway) installed.

The Client Tab allows you to configure the Server Connections that the Mbx.Net clients running on this system will use to connect to the Mbx.Net Servers. A Server Connection defines the server location, the protocol, and, in some cases, the parameters to be used with the selected protocol. Once configured, the client applications select which Server Connection to use by its name.

The Client Tab lists the currently configured Server Connections. The information is provided in two columns: Name and Endpoint.



### Name

This column contains the friendly name that clients will see when browsing for Server Connections to connect to. You should use a name that will easily identify the connection.

### Endpoint

This column contains the actual endpoint that the client will use when connecting to the Mbx.Net Server. It defines the server location and the selected protocol details.

### New...

Click this button to create a new Server Connection.

### Edit...

Select a Server Connection and click this button to edit it.

### Delete

Select a Server Connection and click this button to delete it.

### ***Creating a New Server Connection***

Click the **New** button or right-click inside the Server Connections window and select **New** from the context menu. This will bring up the Server Connection editor.

### ***Editing an Existing Server Connection***

Select the Server Connection and click the **Edit** button or right-click and select **Edit** from the context menu. This will bring up the Server Connection editor.

### ***Deleting an Existing Server Connection***

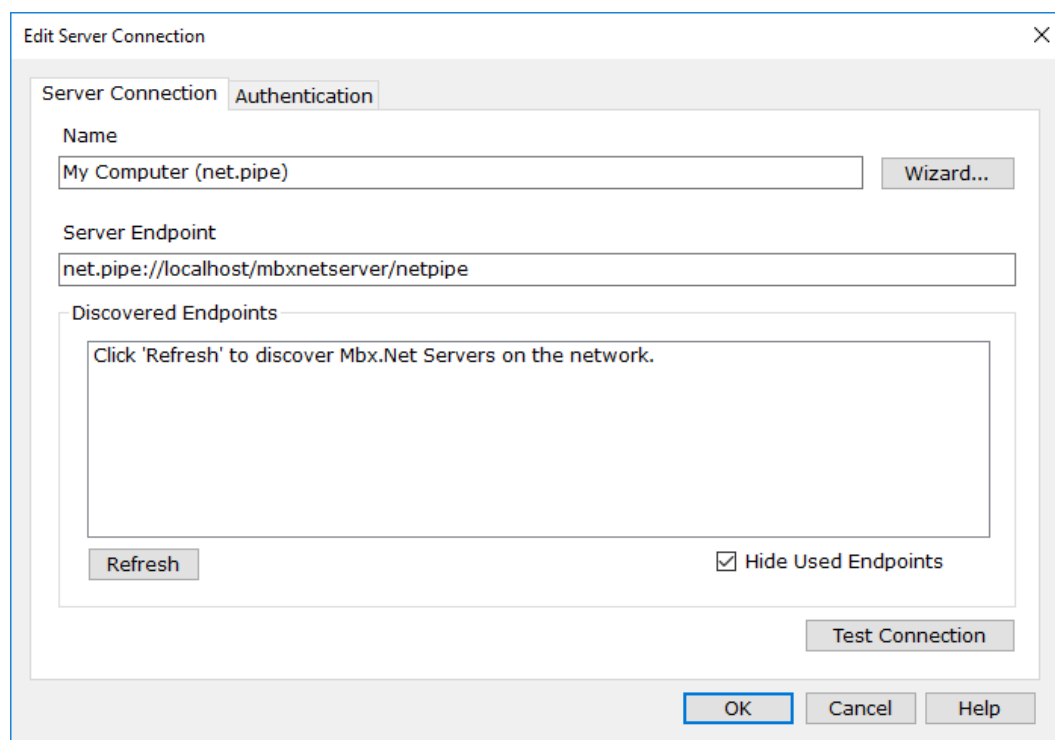
Select the Server Connection and click the **Delete** button or right-click and select **Delete** from the context menu.

## **Server Connection Editor**

This editor allows you to create new or to modify existing Server Connections. The Server Connection editor consists of two tabs: [Server Connection Tab](#) and [Authentication Tab](#).

### ***Server Connection Tab***

On this tab, you can edit the Server Connection name and the server endpoint that is assigned to it. A wizard can be used to assist in creating a new Server Connection. A list of available server endpoints can be discovered to help in the process.



The screenshot shows the 'Edit Server Connection' dialog box with the 'Server Connection' tab selected. The dialog has a title bar with a close button (X). Inside, there are two tabs: 'Server Connection' (active) and 'Authentication'. The 'Server Connection' tab contains the following elements:

- Name:** A text box containing 'My Computer (net.pipe)' and a 'Wizard...' button to its right.
- Server Endpoint:** A text box containing 'net.pipe://localhost/mbxnetserver/netpipe'.
- Discovered Endpoints:** A large text area containing the instruction 'Click 'Refresh' to discover Mbx.Net Servers on the network.' Below this area is a 'Refresh' button.
- Hide Used Endpoints:** A checkbox that is checked.
- Test Connection:** A button located at the bottom right of the main content area.

At the bottom of the dialog, there are three buttons: 'OK', 'Cancel', and 'Help'.

### Name

This sets the friendly name for the Server Connection that the Mbx.Net clients will see when browsing for connections to use.

### Wizard

Click this button to launch a [wizard](#) to assist in creating Server Connections.

### Server Endpoint

This is the endpoint that is associated with the Server Connection. It identifies the server location and the used protocol details.

### Discovered Endpoints

The Mbx.Net Server supports the WS-Discovery protocol. If Mbx.Net Servers on the local network have this service enabled, clients can search the network to discover available servers and their endpoints. This list shows the endpoints that were discovered on the local network.

### Refresh

Click this button to update the list of available Mbx.Net Server endpoints from the local network. The discovered endpoints list is empty when the Server Connection editor is launched for the first time. Once server endpoints are discovered, the list is maintained and available for subsequent editor use. By refreshing the list, endpoints can be discovered that were not previously available.

### Hide Used Endpoints

Check this checkbox to show only endpoints that have not been used yet.

### **Server Connection Wizard**

This wizard assists in manually creating a Server Connection. You should use it when the server is outside of the local network, since only endpoints from Mbx.Net Servers on the local network can be discovered. The discovery service can also be blocked by routers or firewalls, or may not be enabled at the server. In these cases, the wizard can also help to create a correct Server Connection. The required information to be entered here can be found by running the Mbx.Net Configuration Editor on the system with the Mbx.Net Server, and then selecting the Server Tab.

Server Connection Wizard

This wizard will help you create a valid endpoint to a system running the Mbx.Net server. Select the protocol and enter the IP address or computer name. The port will default to the default value based on the protocol selected.

Protocol

Net.Pipe  Net.Tcp  HTTP

IP Address/Computer Name Port

### Protocol

Selects the protocol to use. The available protocols are Net.Pipe, Net.Tcp or HTTP.

### IP Address/Computer Name

This is either the IP Address or Computer Name of the system running the Mbx.Net Server. In the case of Net.Pipe connections, this field is always set to localhost.

### Port

This is the port that the system running the Mbx.Net Server is configured to listen on. Based on the protocol, a default port number is automatically selected. However, if the server is configured to use a different port number, change the value here to match it. Notice that Net.Pipe endpoint is always local and does not require a port number.

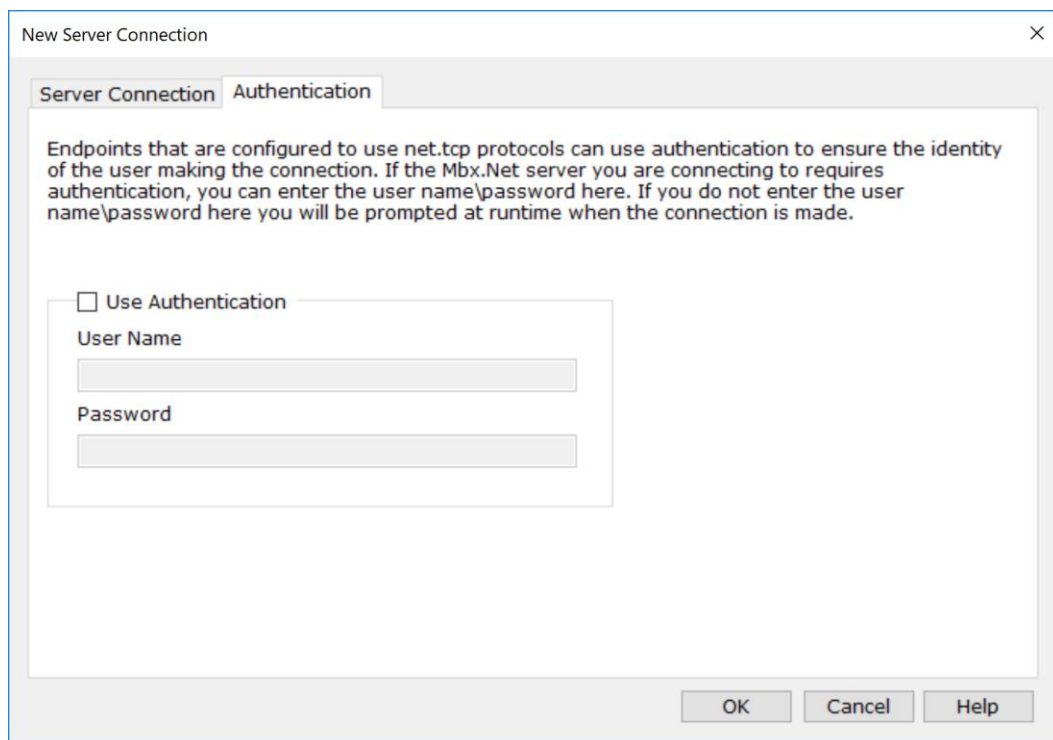
### ***Authentication Tab***

#### **Note**

This tab is only used with endpoints that use Net.Tcp protocol. Net.Pipe and HTTP protocols do not support authentication.

Endpoints that use Net.Tcp protocol can also be configured to use authentication to ensure the identity of the user making the connection. If the selected endpoint requires authentication, you can enter the user name and password here of an authorized user.





### Use Authentication

Check this checkbox to enable authentication for the selected endpoint.

### User Name

Enter a user name that is authorized to connect to the Mbx.Net Server through the selected endpoint.

If you are connecting to a computer that is in a domain, you can optionally specify the desired domain name. To specify a domain, enter it in the form <DomainName>\<UserName>. If you specify just the user name, the server will assume you are operating within the local computer's domain.

### Password

Enter the password that is associated with the selected user name.

## Server Tab

**Note** The Server Tab is only available on systems that have the Mbx.Net Server installed.

The Server Tab allows you to configure the Mbx.Net Server running on this system. It also allows you to monitor the clients connected to this server.

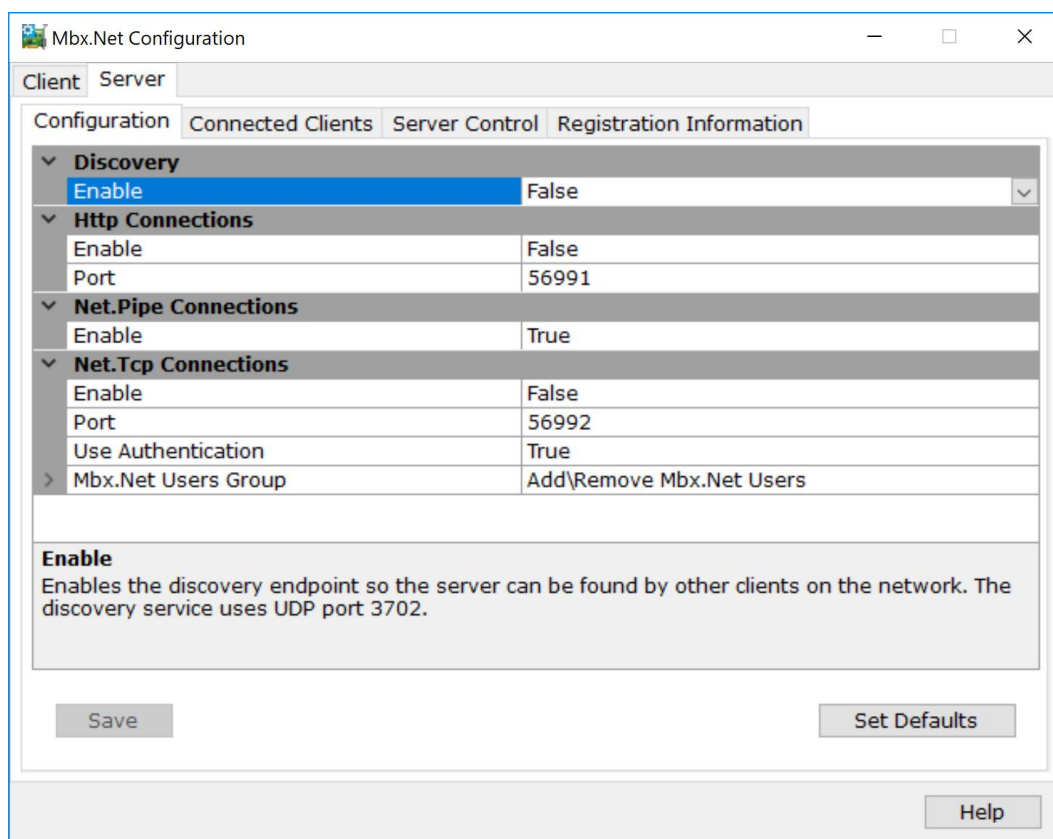
The Server Tab consists of four tabs:

- [Configuration Tab](#)
- [Connected Clients Tab](#)
- [Server Control Tab](#)
- [Registration Information Tab](#)

The following sections provide complete descriptions of these tabs.

## Configuration Tab

On this tab, you can edit most of the Mbx.Net Server-related parameters. The server supports the WS-Discovery, Http, Net.Pipe and Net.Tcp protocols. The information entered here may be needed when configuring the Server Connections on the client side.



### ***Discovery***

The Mbx.Net Server supports the WS-Discovery protocol. When enabled, the server can announce the availability of the supported protocols over the network. The discovery service uses UDP port 3702.

Enable

Set to True to enable the discovery service. By default, the discovery service is disabled.

**Http Connections**

The HTTP protocol uses BasicHttpBinding and sends SOAP 1.1 messages. Security is turned off and UTF-8 text encoding is used.

Enable

Set to True to enable the HTTP protocol. By default, the HTTP protocol is disabled.

Port

Enter the port number to be used with the HTTP connections. The default port number is 56991.

**Net.Pipe Connections**

The Net.Pipe protocol uses NetNamedPipeBinding. It can only be used for local connections.

Enable

Set to True to enable the Net.Pipe protocol. By default, the Net.Pipe protocol is enabled.

**Net.Tcp Connections**

The Net.Tcp protocol uses NetTcpBinding and may require a Windows user name and password when establishing the connection.

Enable

Set to True to enable the Net.Tcp protocol. By default, the Net.TCp protocol is disabled.

Port

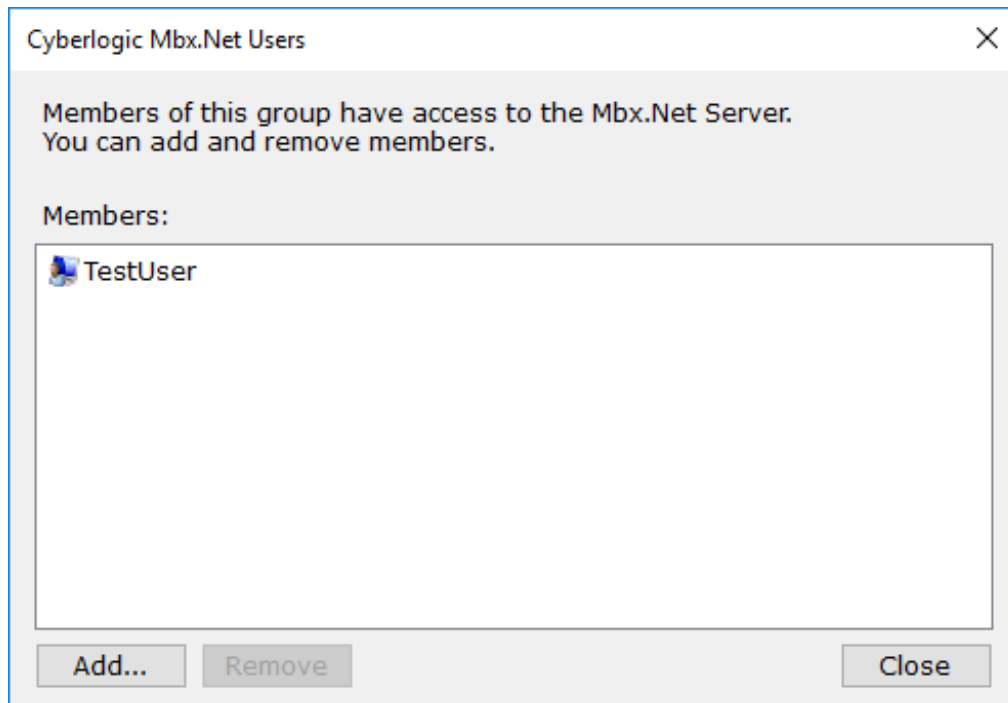
Enter the port number to be used with the Net.Tcp connections. The default port number is 56992.

Use Authentication

Set to True to enable user authentication. By default, user authentication is enabled.

### Mbx.Net Users Group

You can set a group of users that are authorized to connect to the Mbx.Net Server. The name of this group is **Cyberlogic Mbx.Net Users**. Click the edit button to edit this group.



Click the **Add...** button to add a member to the group. Select a group member and click the **Remove** button to remove a member.

### **Save**

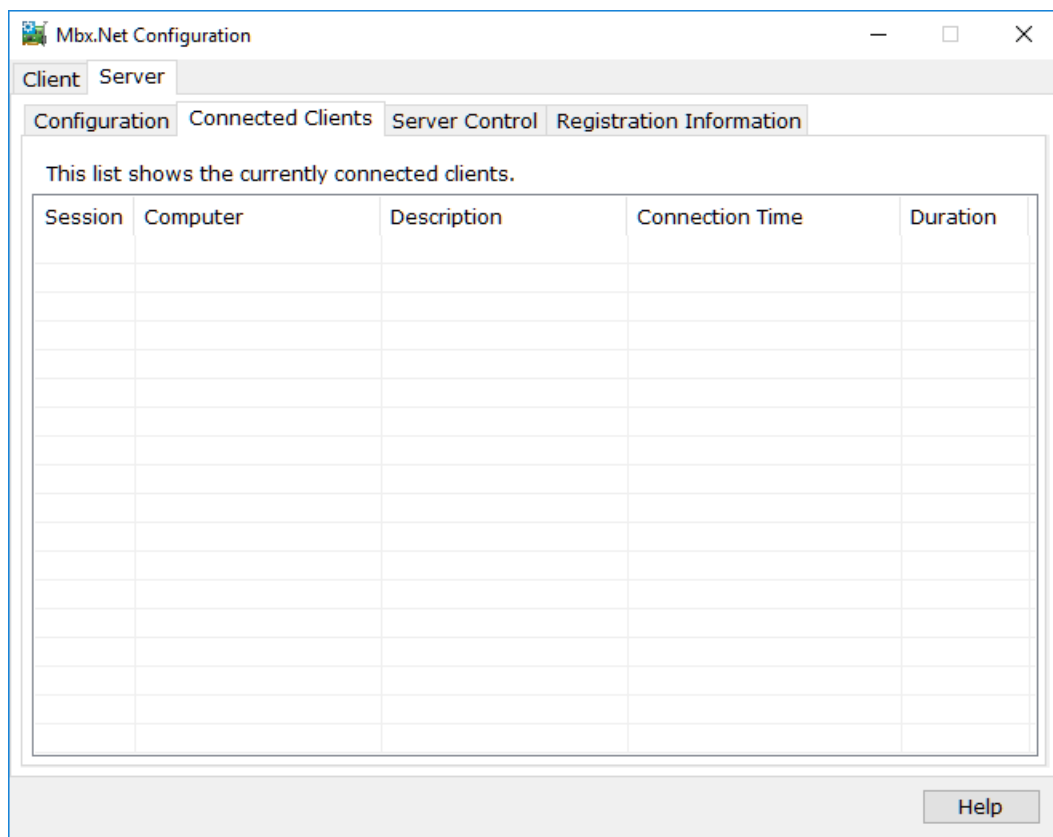
Click this button to save your changes. If the server is running, a restart will be required.

### **Set Defaults**

Click this button to restore the default settings.

## **Connected Clients Tab**

This tab shows a list of the clients that are currently connected to the Mbx.Net Server on the system.



### Session

The session number of the connection. Each connection creates a new session.

### Computer

The Name or IP Address of the computer that the connected client is running on.

### Description

The description of the client. This is a descriptive string identifying the client application that the client sent to the server during the initial connection.

### Connection Time

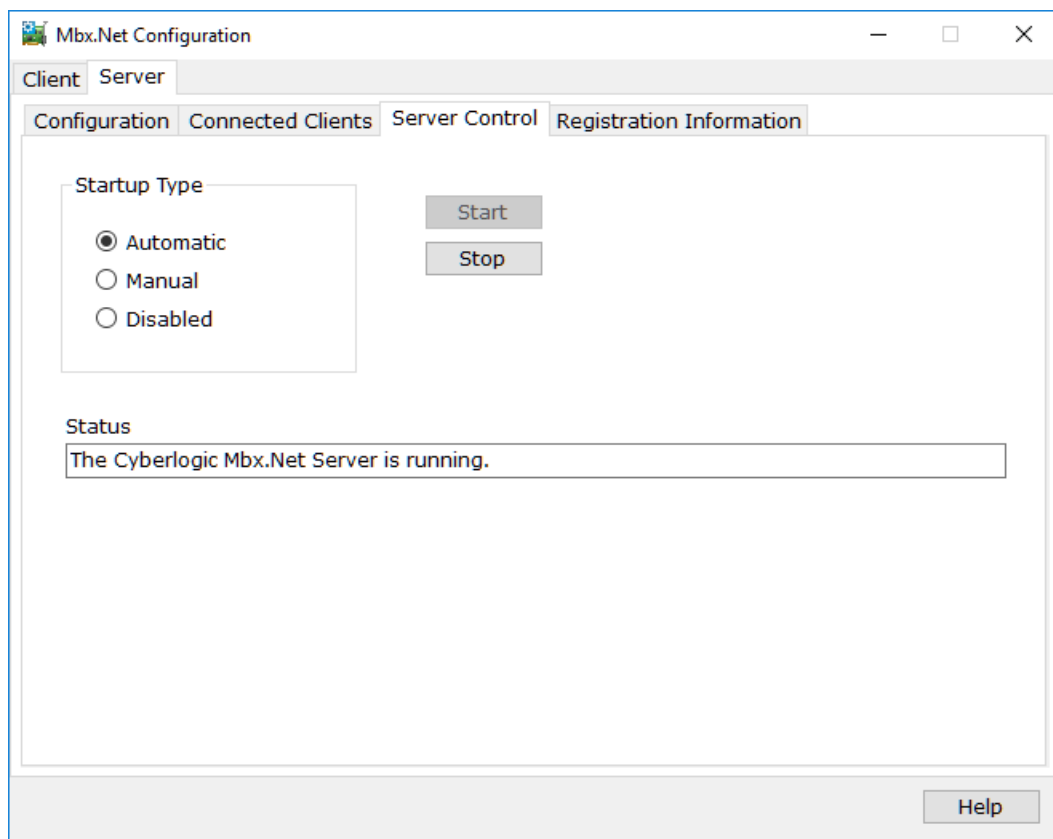
The time at which the connection from the client was established.

### Duration

The amount of time that the client has been connected.

## Server Control Tab

This tab allows you to designate if and how you want the Mbx.Net Server to start.



### Automatic

When this option is selected, the Mbx.Net Server will start when Windows boots.

### Manual

When this option is selected, the Mbx.Net 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.Net Server will not run.

### Start

In Automatic or Manual mode, click this button to start the Mbx.Net Server.

### Stop

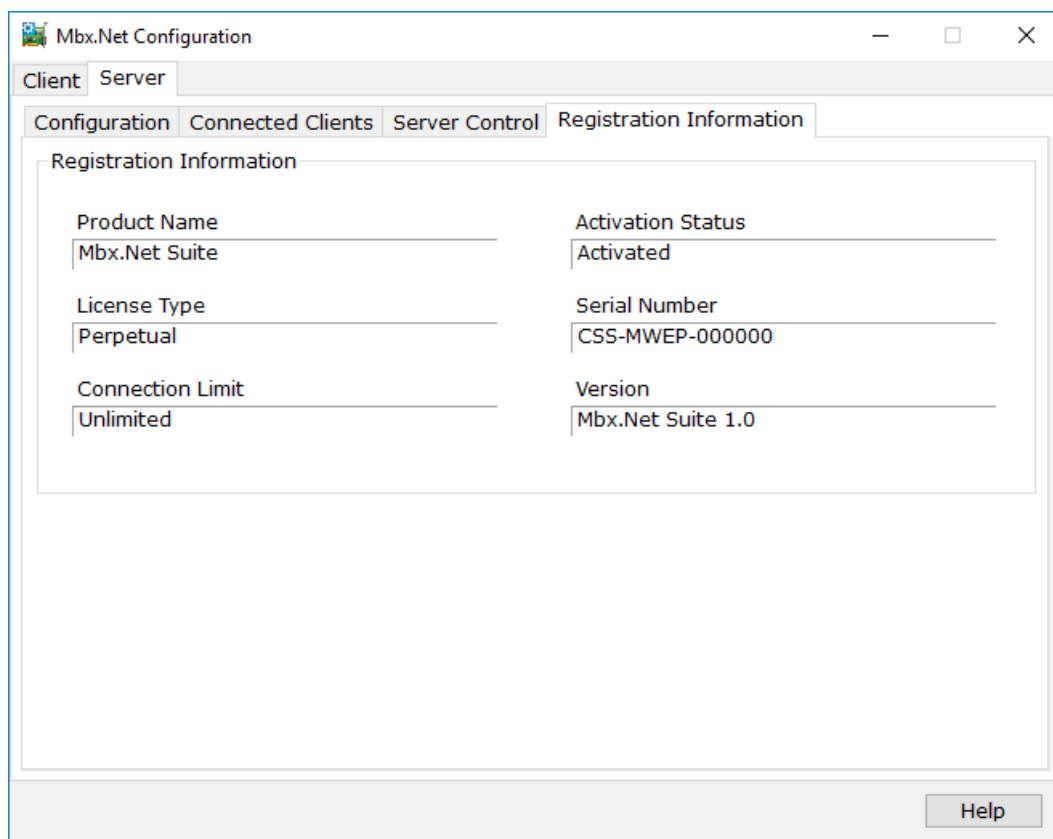
In Automatic or Manual mode, click this button to stop the Mbx.Net Server.

### Status

This tells you whether the Mbx.Net Server is running, stopped, starting or stopping.

## Registration Information Tab

The Mbx.Net Server is a licensed component and requires activation. This tab shows you the current server activation status, along with its version and serial number. This information may be requested if you call for technical support. This tab also tells you if the software has been activated or if it is running in demo mode.



### Product Name

The Mbx.Net Server is part of Cyberlogic's Mbx.Net Suite and Mbx.Net Premier Suite. This field indicates the suite that is installed on your system.

### Activation Status

The Cyberlogic software products operate in a time-limited demonstration mode until they are activated. This field tells you if 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 the product suite you have installed. You will need the serial

number and password that were assigned to you when you purchased your license for the software.

*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.

*Connection Limit*

This specifies the number of simultaneous connections that the server can support.



## VALIDATION & TROUBLESHOOTING

The following section describes how you can use the MbpStat.Net utility to verify that the Mbx.Net client and server sides are configured correctly.

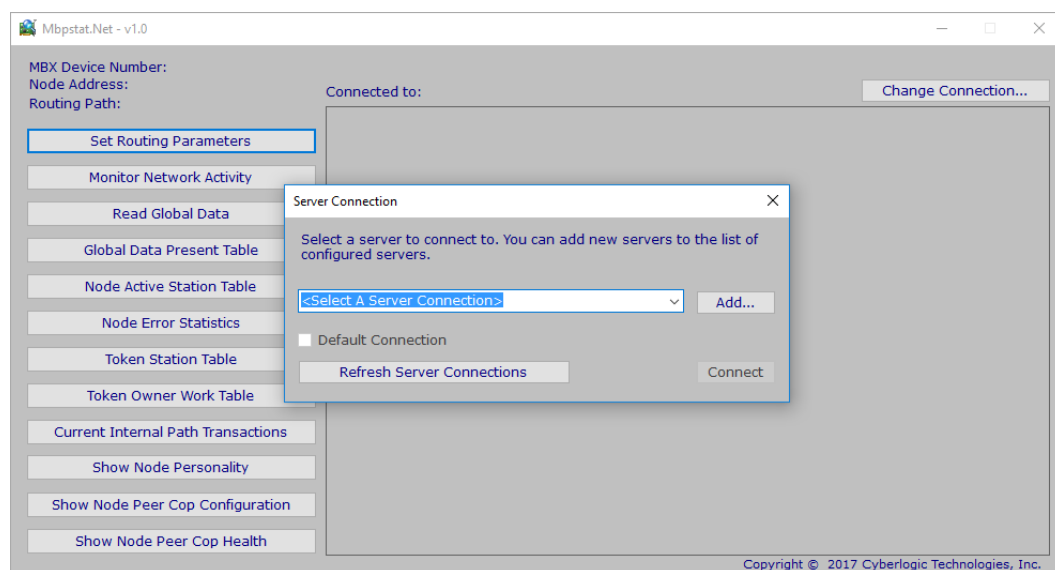
If you are having difficulties communicating, the troubleshooting sections can help you determine the nature of the problem. Included is a description of the [Event Viewer](#) along with the [Frequently Asked Questions](#).

### MbpStat.Net

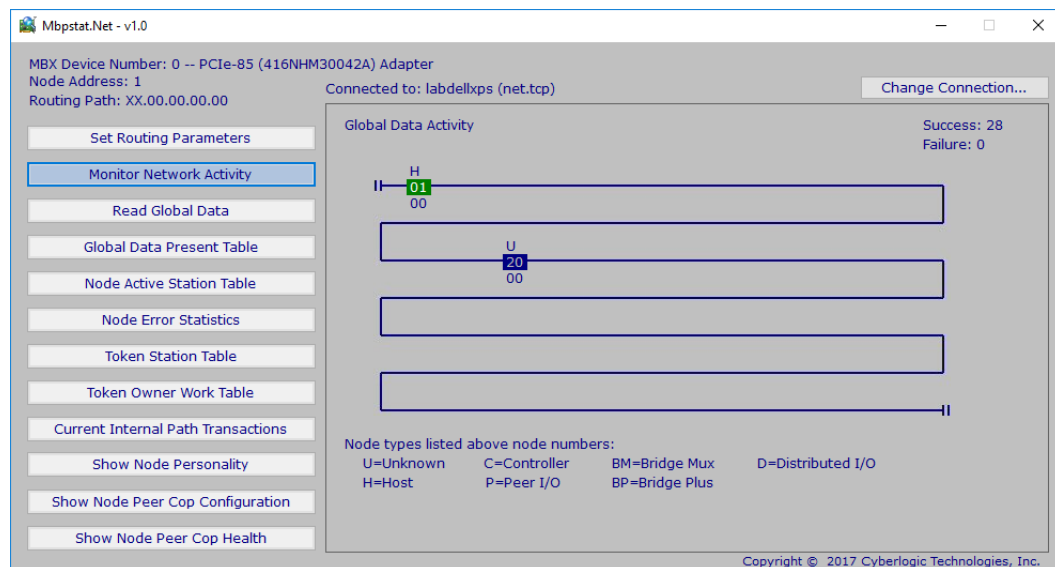
MbpStat.Net is the Modbus Plus network diagnostic utility. You can use it to view active nodes, monitor overall network activity, or view error statistics from a specific node.

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 also run on 32-bit and 64-bit Windows, and it does not require the Virtual MBX driver.

To run the program, open the Windows **Start** menu and locate the **Cyberlogic Mbx.Net Suites**. From there, go to the **Diagnostics** submenu and select **MbpStat.Net**.



When prompted, select the **Server Connection** and click the **Connect** button. Next, the program will prompt you to enter the MBX Device to use and the routing path. Following that, the program starts the Monitor Network Activity diagnostic screen.



## How to Navigate the Options

The left side of the screen includes the options menu. You can use the mouse or the **Up/Down** and **Enter** keys to select the various diagnostics from the menu.

Some selections may require a node address. The first time that it's needed, you will be prompted to enter this address. From that point on, the entered address will be used on all options where the node address is needed. To change this selection, press the **ESC** key, and you will be prompted to enter a new address.

Use the **Change Connection...** button to switch to a different Server Connection. When prompted for a new connection, you can check the **Default Connection** checkbox. This will designate this connection as the default connection, which will automatically be used when the program starts the next time.

### Options Menu

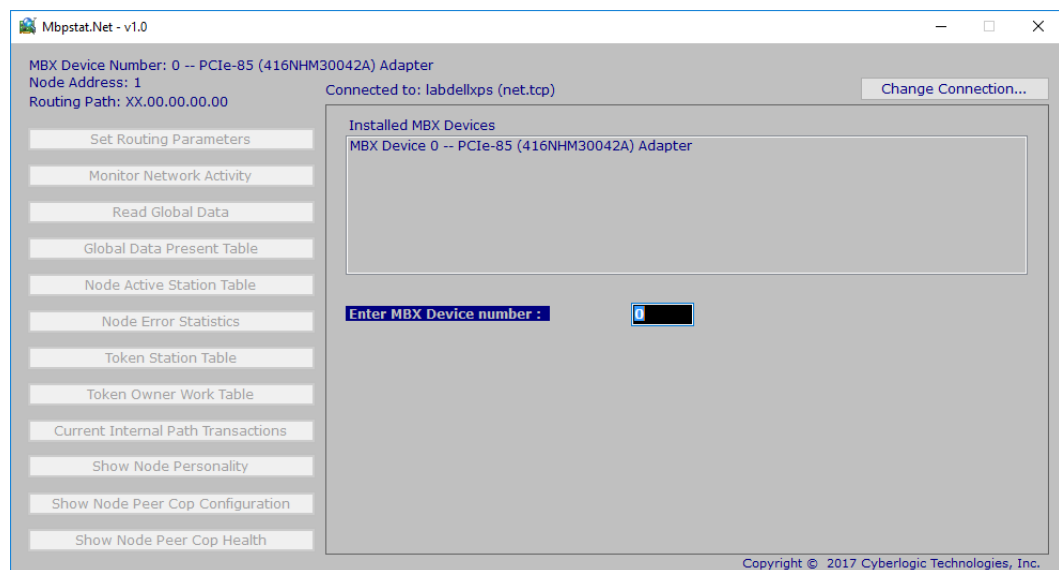
The left side of the screen includes a menu of the following options:

- [Set Routing Parameters](#)
- [Monitor Network Activity](#)
- [Read Global Data](#)
- [Global Data Present Table](#)
- [Node Active Station Table](#)
- [Node Error Statistics](#)
- [Token Station Table](#)

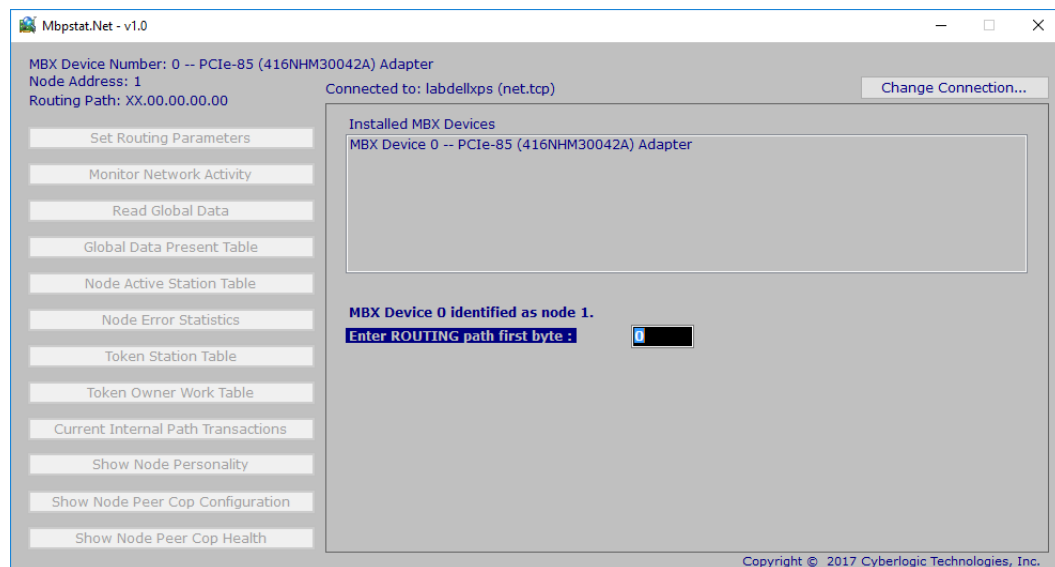
- [Token Owner Work Table](#)
- [Current Internal Path Transactions](#)
- [Show Node Personality](#)
- [Show Node Peer Cop Configuration](#)
- [Show Node Peer Cop Health](#)

## Set Routing Parameters

This menu option allows you to select the node to monitor. All subsequent menu options use this selection.



First, you are prompted to enter the MBX Device number that you want to monitor. The MBX Device identifies one of your network adapter cards through which the MbpStat.Net will communicate.



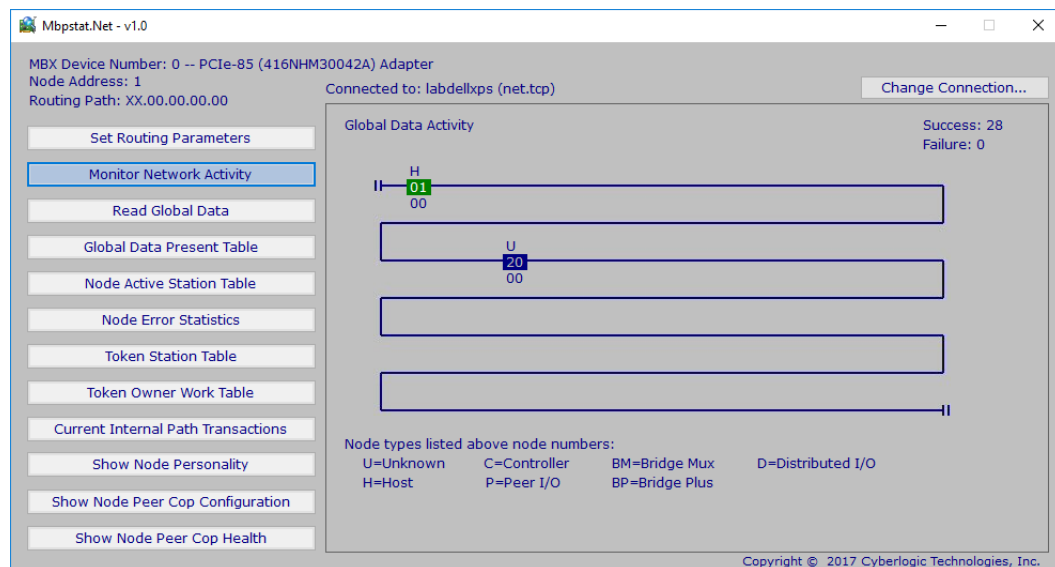
Next, the program will prompt you for a routing path to the Modbus Plus network that you want to analyze.

For example, you can examine the local network or a remote network that is accessed through a series of bridges. Notice that you are selecting only the network. The address of a specific node on the network will be entered later in one of the options that requires a node address.

You can enter up to four network routing bytes. The last non-zero entry defines the target network. The first zero entry terminates the routing. To monitor the local network, enter zero for the first routing byte.

## Monitor Network Activity

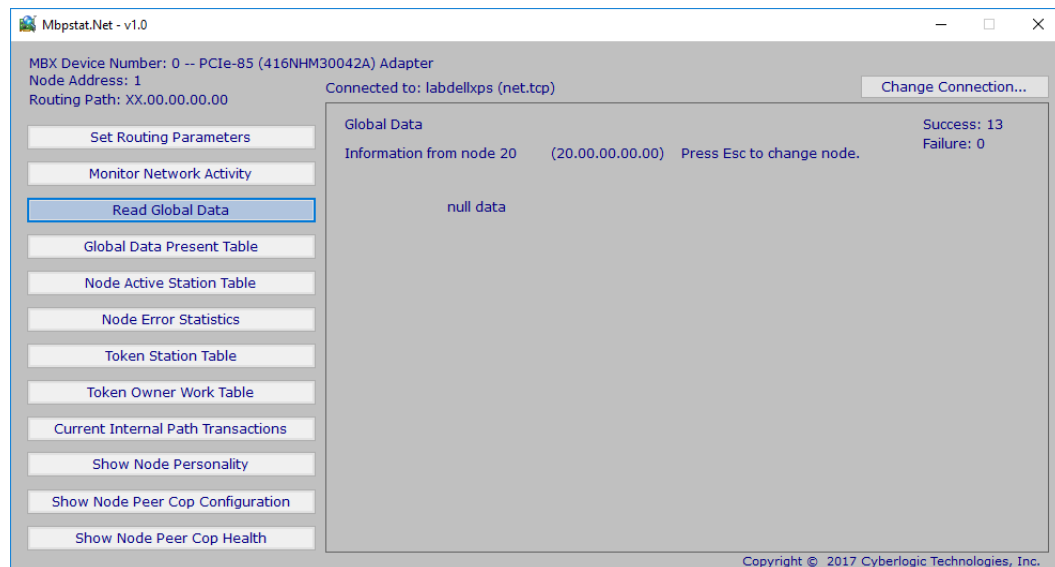
This option lists the nodes that exist on the network specified by the routing path. The program attempts to communicate with every node from 1 to 64, and checks whether a response has been received from each node.



The program also attempts to read global data from each active node and displays the number of words of global data read. The global data word count is displayed below each node's address. Zeros will be displayed if no global data was read. If you are accessing a remote network through a bridge, the program will inform you that global data is not accessible through a bridge.

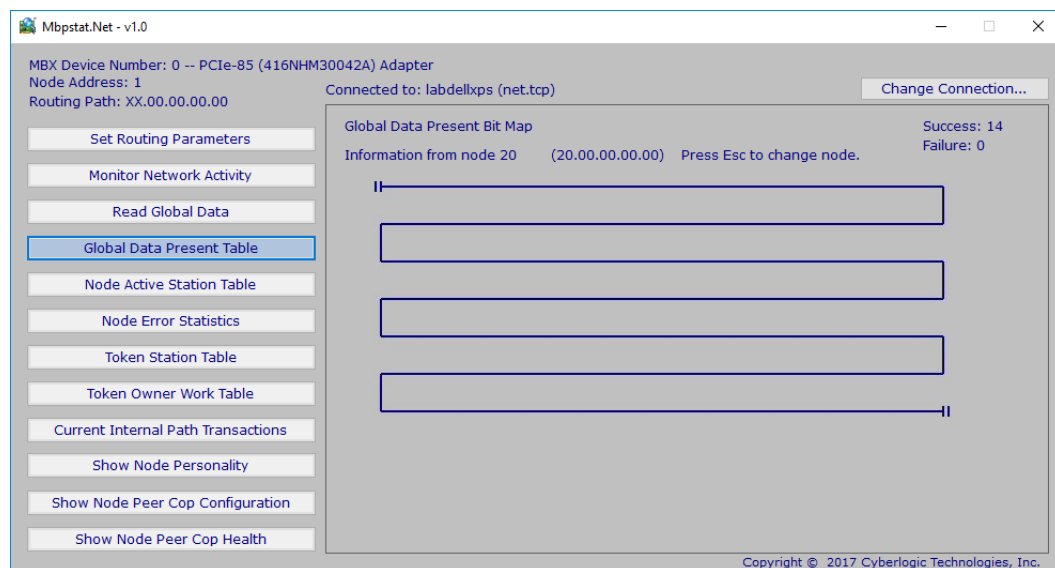
## Read Global Data

This option continuously reads and displays global data from the selected node. If global data is present but is zero words in length, a "null data" message is displayed.



## Global Data Present Table

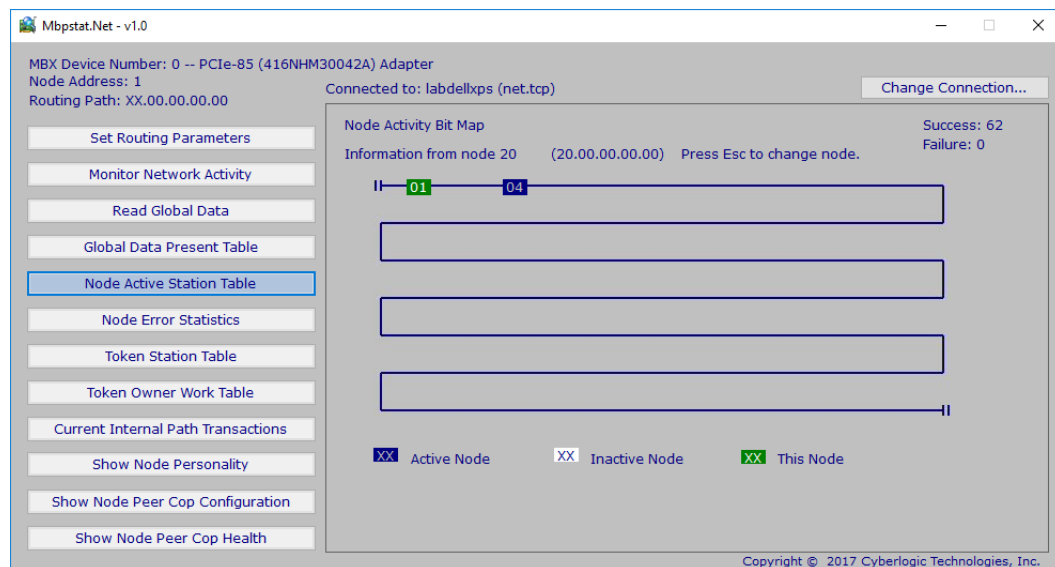
This option displays a map table of the nodes that have global data present, as seen by the specified node.



Node addresses appearing in the table are those from which the specified node has received global data since the test was started. The nodes in the table are not necessarily the only ones active on the network.

## Node Active Station Table

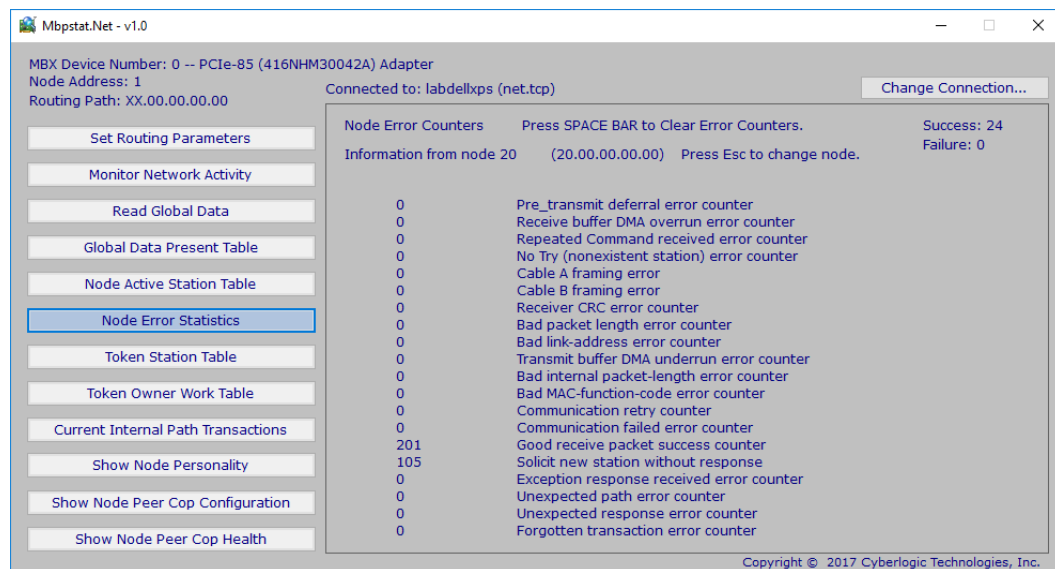
This option continuously queries the specified node for its list of active nodes and displays that list.



Node addresses shown in normal video are those seen by the node as remaining active during the test. Node addresses in reverse video are those which were active at least once during the test, but which have subsequently gone inactive.

## Node Error Statistics

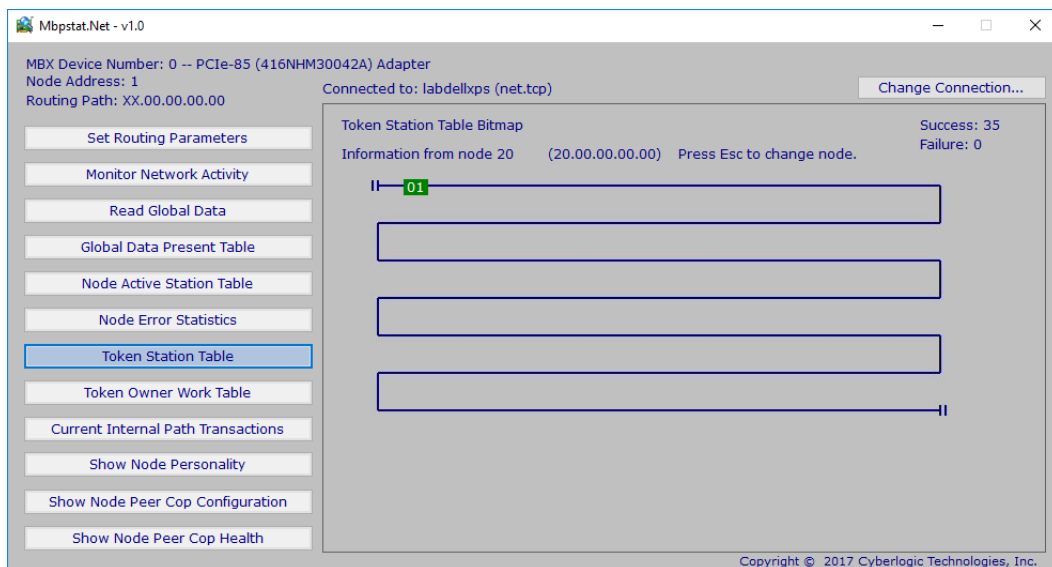
This option continuously queries the specified node for a list of its error statistics and displays the list. As the test runs, the counts in the display can be reset to zeros by pressing the **Space** bar.



If you are running the utility on a single-cable network, one of the 'framing error' counters will be incrementing continuously due to the open connection on the monitoring device's other cable connector. For example, if your single cable is connected to the Cable A connector, the 'Cable B framing error' counter will increment continuously. In such cases, you can disregard the counter.

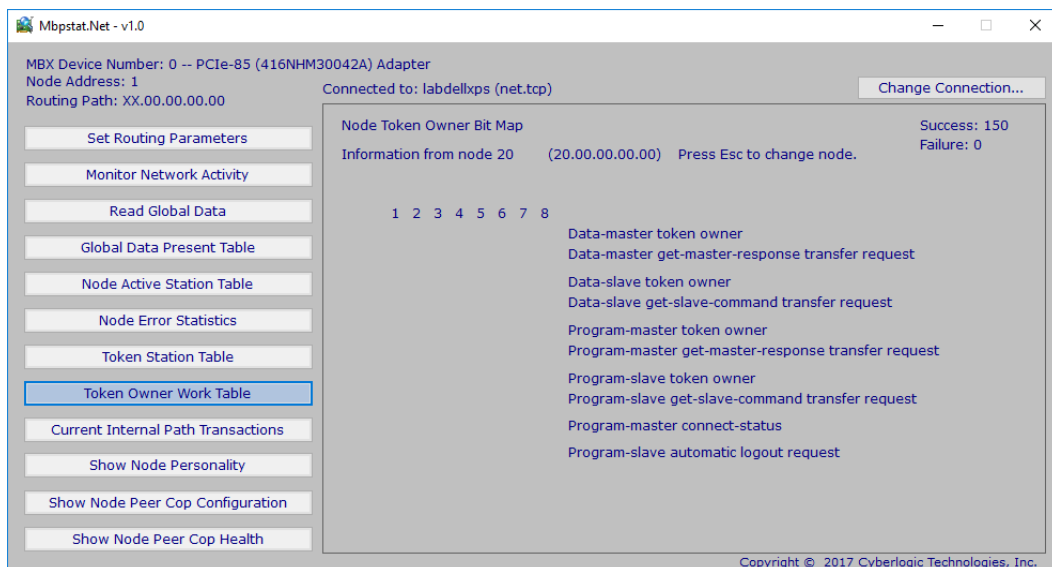
## Token Station Table

This option displays a map table of the nodes that are passing the network token, as seen by the specified node.



## Token Owner Work Table

This option continuously queries the specified node for its list of internal paths that are active. The paths listed are the node's internal data and program paths.



As a path goes active, its number is listed in the columns to the left. Paths shown in normal video are those remaining active during the test. Paths in reverse video are those which were active at least once during the test, but which have subsequently gone inactive.



## Current Internal Path Transactions

This option displays the count of transactions processed by each of the specified node's internal paths. As the test runs, the counts in the display can be reset to zeros by pressing the **Space** bar.

The screenshot shows the Mbpstat.Net v1.0 interface. The left sidebar contains a menu with the following items: Set Routing Parameters, Monitor Network Activity, Read Global Data, Global Data Present Table, Node Active Station Table, Node Error Statistics, Token Station Table, Token Owner Work Table, **Current Internal Path Transactions** (highlighted), Show Node Personality, Show Node Peer Cop Configuration, and Show Node Peer Cop Health. The main window displays the following information:

MBX Device Number: 0 -- PCIe-85 (416NHM30042A) Adapter  
 Node Address: 1  
 Routing Path: XX.00.00.00.00  
 Connected to: labdelxps (net.tcp) [Change Connection...]

Node Transaction Counters    Press SPACE BAR to Clear Error Counters.    Success: 23  
 Information from node 20    (20.00.00.00.00)    Press Esc to change node.    Failure: 0

PATH	1	2	3	4	5	6	7	8
DM Actual	0	0	0	0	0	0	0	0
DM Total	0	0	0	0	0	0	0	0
DS Actual	0	0	0	0	0	0	0	208
DS Total	0	0	0	0	0	0	0	0
PM Actual	0	0	0	0	0	0	0	0
PM Total	0	0	0	0	0	0	0	0
PS Actual	0	0	0	0	0	0	0	0
PS Total	0	0	0	0	0	0	0	0

Station Management Input Commands Actual: 0  
 Station Management Input Commands Total: 0  
 Total Inbound Commands: 0  
 Total Outbound Commands: 0

Copyright © 2017 Cyberlogic Technologies, Inc.

## Show Node Personality

This option displays information about the specified node, such as its node type and software version. It also shows information about the node's current activity on the network.

The screenshot shows the Mbpstat.Net v1.0 interface. The left sidebar contains a menu with the following items: Set Routing Parameters, Monitor Network Activity, Read Global Data, Global Data Present Table, Node Active Station Table, Node Error Statistics, Token Station Table, Token Owner Work Table, Current Internal Path Transactions, **Show Node Personality** (highlighted), Show Node Peer Cop Configuration, and Show Node Peer Cop Health. The main window displays the following information:

MBX Device Number: 0 -- PCIe-85 (416NHM30042A) Adapter  
 Node Address: 1  
 Routing Path: XX.00.00.00.00  
 Connected to: labdelxps (net.tcp) [Change Connection...]

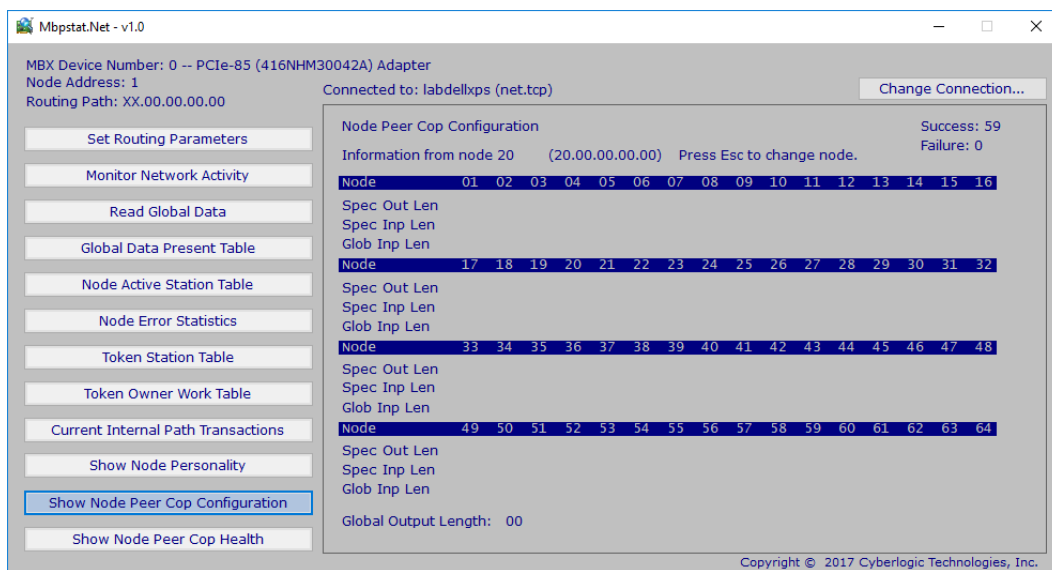
Node Personality    Success: 7  
 Information from node 20    (20.00.00.00.00)    Press Esc to change node.    Failure: 0

Node Number = 20  
 Node Type = Host  
 Software version = 3.01  
 Network address = 20  
 MAC state variable = idle  
 Peer status = normal link operation  
 Token pass counter = 2903  
 Token rotation time = 1 milliseconds  
 Description = None

Copyright © 2017 Cyberlogic Technologies, Inc.

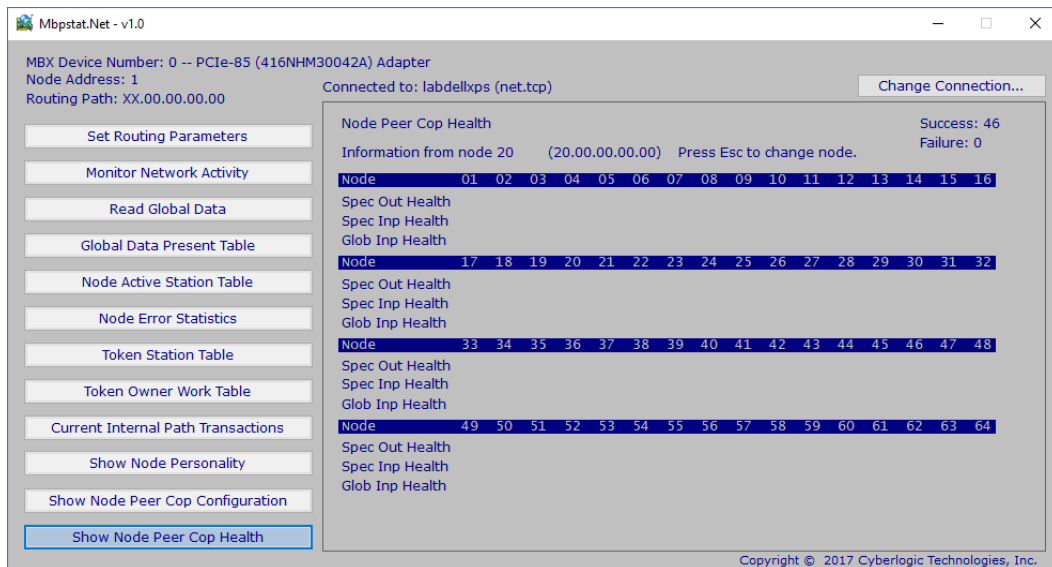
## Show Node Peer Cop Configuration

This option displays information about the Peer Cop Configuration of each active node, as seen by the selected node.



## Show Node Peer Cop Health

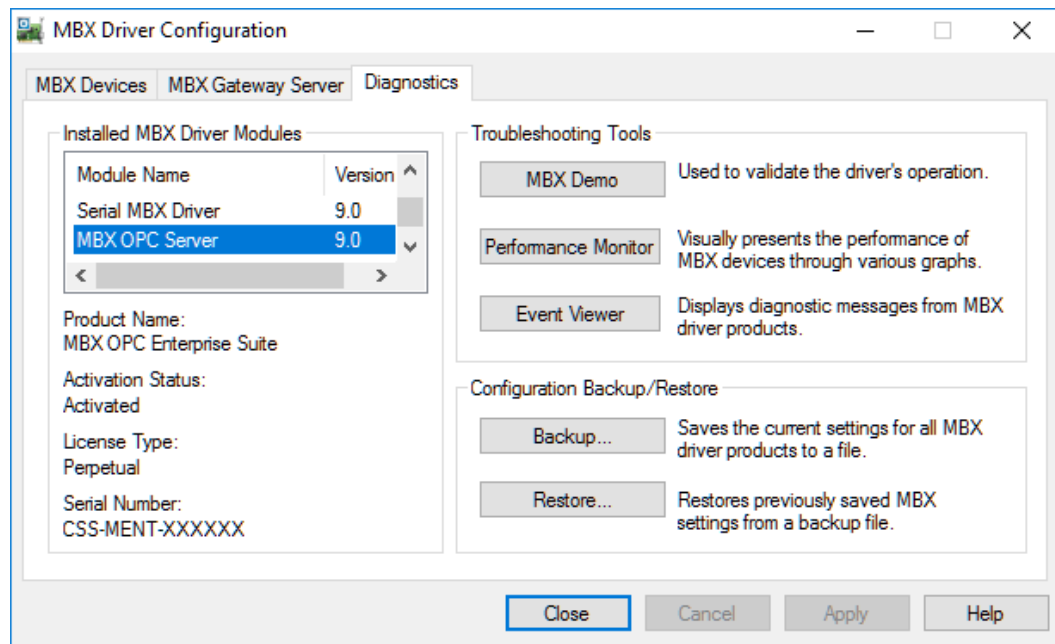
This option displays information about the Peer Cop Health of each active node, as seen by the selected node.



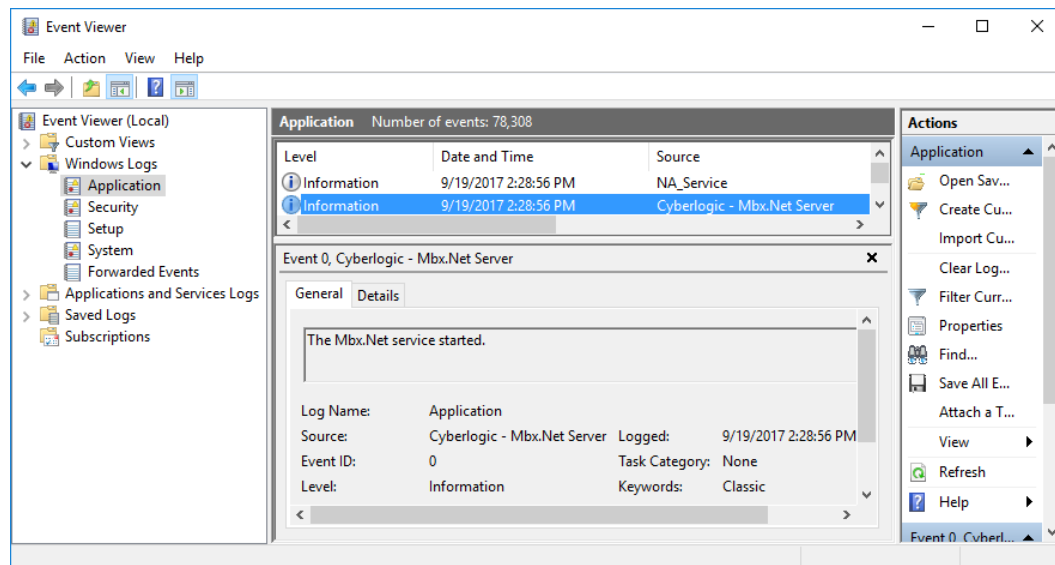
## Event Viewer

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 product suite you have installed. 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.

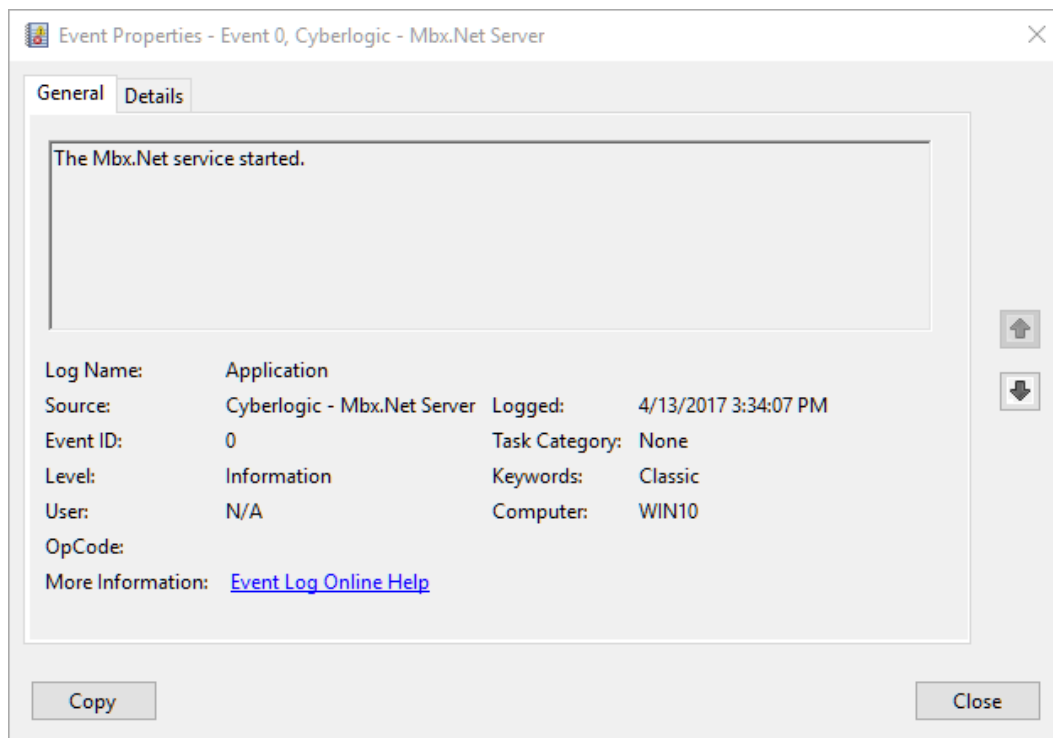


2. If you are looking for events related to the Mbx.Net Server, select the **Application** branch from the Event Viewer tree, and look for entries in the **Source** column named **Cyberlogic – Mbx.Net Server**.

**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.



## Frequently Asked Questions

***My Net.Tcp Server Connection requires authentication and I did not enter it when I configured the connection. When I try to use the Server Connection, I'm not prompted to enter a user name and password.***

In most cases, if you do not enter a user name and password when you configure the Server Connection, you will be prompted to enter it at runtime when the connection is made. However, if an application, such as a service, connects to the Mbx.Net Server, you will not be prompted and the connection will fail. This is because these types of applications do not have a window to present the message and, in many cases, run in the system account. In this case, you will have to enter the user name and password when you configure the Server Connection.

***I have enabled discovery on my Mbx.Net Server. However, when I try to***

***discover servers on my client computer, it is not detected.***

The discovery service will only detect servers on a local subnet and will not go through routers. In this case, you will need to use the [Server Connection Wizard](#) to configure the Server Connection.

***I have configured a Server Connection. However, I cannot connect to the server from my client application.***

This can be caused by a couple of issues. First, check that the [firewall exceptions](#) are enabled on the server system. Also, check that the protocol is enabled and the correct port number is being used. If the Net.Tcp protocol is being used and authentication is enabled, make sure that you have the correct user name and password and that the user is a member of the [Mbx.Net Users group](#).

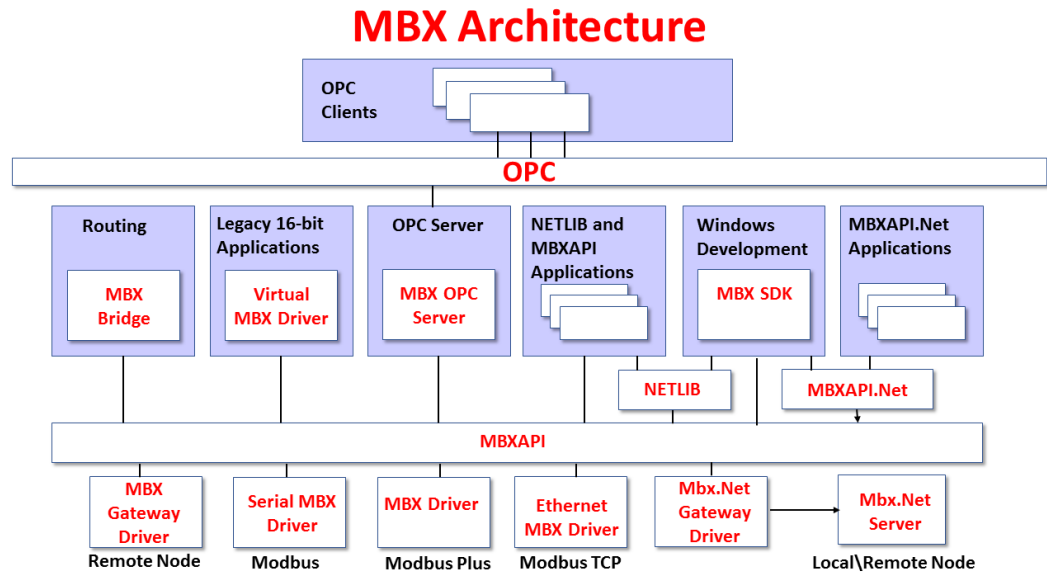
***I have installed the Mbx.Net Server and MbpStat.Net. However, when I run MbpStat.Net and connect to the local system, no MBX devices are found.***

The Mbx.Net Server integrates with the MBX Driver Suite on the system. Make sure that you have a supported version of the MBX Driver Suite installed on the system and at least one MBX device is configured.

## APPENDIX: MBX ARCHITECTURE AND COMPANION PRODUCTS

The Mbx.Net Gateway 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 [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 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.