

KB2010-10: MBX BRIDGE TUTORIAL

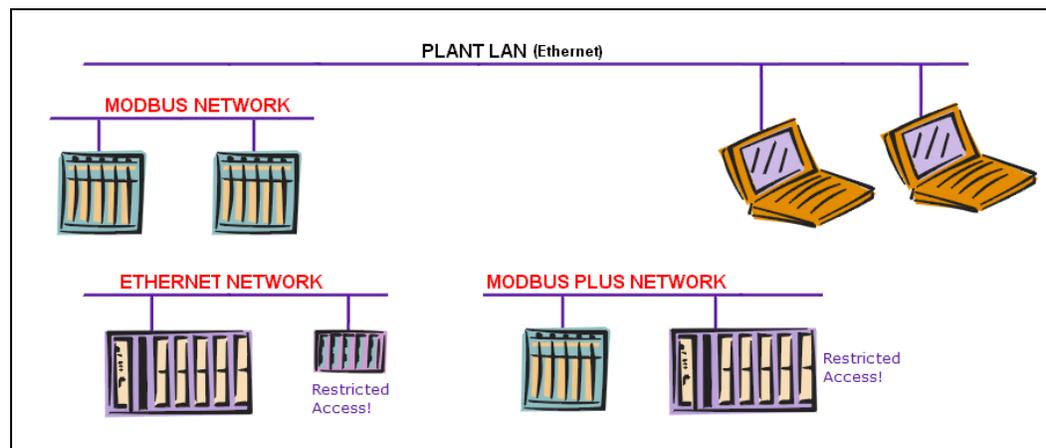
Cyberlogic's MBX Bridge software lets you pass messages among the three Modicon network types: Modbus, Modbus Plus and Ethernet. As the only product that can bridge all three of these networks, it is uniquely able to provide compatibility solutions for Modicon communications.

This tutorial introduces new users to the basic features and capabilities of this powerful software product. For detailed information on configuring and using Cyberlogic's MBX Bridge, refer to the MBX Bridge Help file.

Why would I need an MBX Bridge?

You wouldn't—if your plant had only a single network, and all of your devices could use it to exchange all the information they needed to communicate. But most facilities are not like that. Chances are, your situation is much more complex.

Most likely, you have to deal with many networks in different parts of the plant. These were installed at different times and use different protocols. Where two networks use the same protocol, they may have conflicting or incompatible addressing. Even if you *could* just tie two networks together, you might not want to. You may want to restrict which nodes on one network can exchange data with nodes on the other.



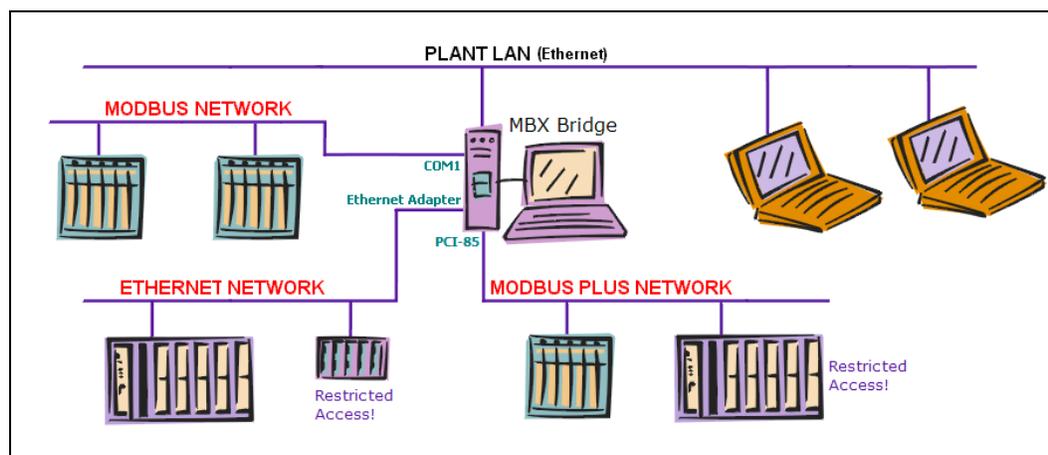
The problem: a disconnected mixture of different networks.

If your networks are a mix of Modbus, Modbus Plus and Ethernet, the MBX Bridge can help you make reliable, secure connections across all of them. You will be able to:

- Pass data across networks that use different protocols
- Pass data across networks that use the same protocol, but with incompatible addressing schemes
- Restrict which nodes are accessible from another network
- Restrict which Ethernet nodes can access another network

How does the MBX Bridge do that?

The MBX Bridge is a software product that runs under Windows 7, Vista, XP, Server 2008 or Server 2003. The system it resides in can connect to Modbus, Modbus Plus and Ethernet through the standard hardware connections these networks use: serial ports, dedicated interface adapters or Ethernet ports. The number of these connections is limited only by your hardware's capability to add expansion cards and ports.



The solution: the MBX Bridge can bring them all together.

When a device on one network wants to send a message to a device on another network, it sends the message to the MBX Bridge. The bridge examines the address associated with the message to decide where to send it. To do this, it uses a table of routing records that you configure to specify the routing you desire. When the bridge sees an incoming message that matches one of the routing records, it sends the message to the network and address specified by that record.

This technique allows the bridge to route messages among all of the network types it can service. You can configure the bridge to route messages between any pair of networks where it is needed, and leave out the routing between networks that should not be able to exchange messages. You can configure routing for each node on a network, designating which of the other networks are permitted to talk to that node and which are not. In the case of an Ethernet node, you can even permit or prevent its access to individual nodes on other networks.

What hardware and software do I need?

You will need a PC running one of these operating systems:

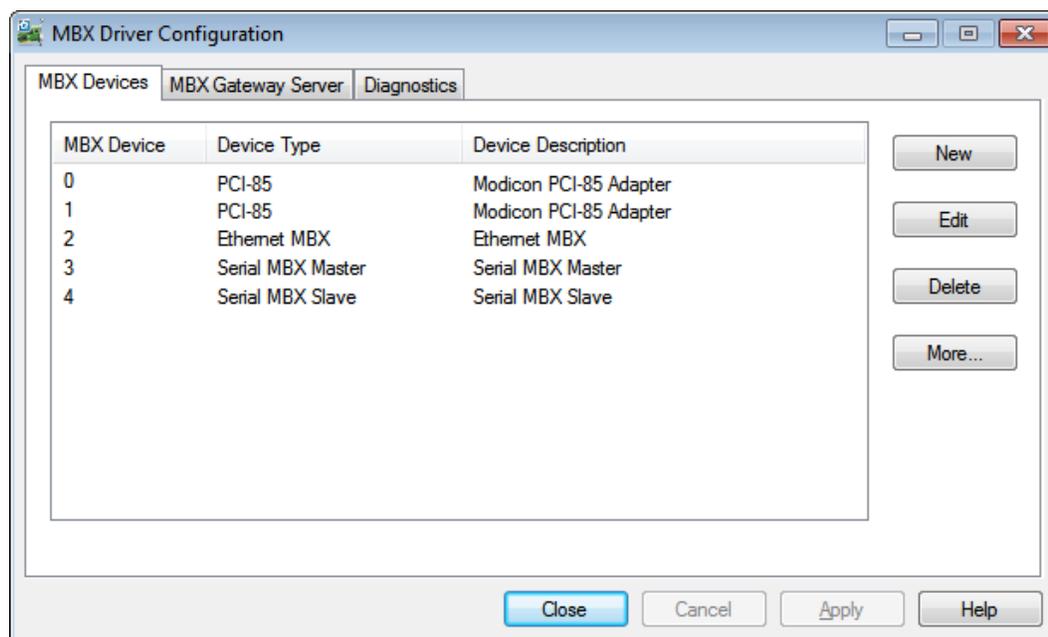
- Windows 7
- Windows Vista
- Windows XP
- Windows Server 2008
- Windows Server 2003

The computer must have the hardware ports or adapter cards needed to make the physical connection to the networks you want to bridge.

The MBX Bridge Suite includes Cyberlogic's MBX Driver Suite, which provides all of the device drivers you need to set up your system. There is no more software that you need to buy. The MBX Drivers are the industry standard, optimized for maximum performance. Because they are included in the MBX Bridge Suite, you are assured that they will be at the same revision level and fully compatible with the bridge software.

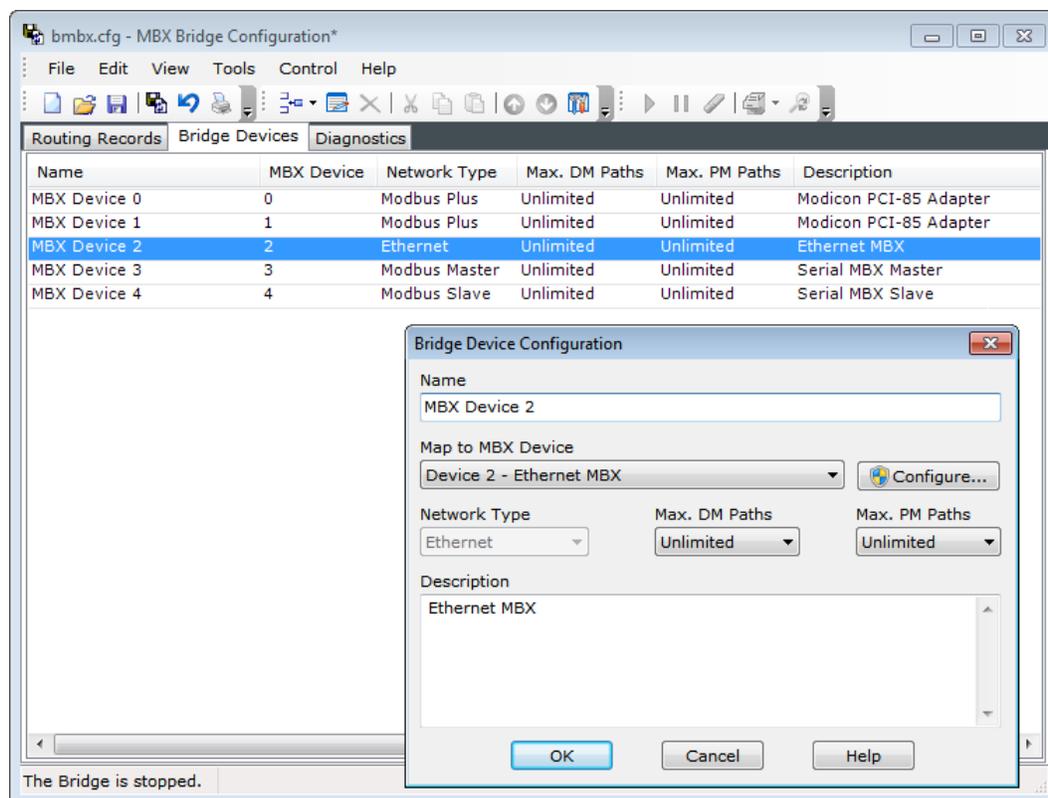
I'M READY TO CONFIGURE THE BRIDGE. WHERE DO I START?

You begin by configuring the device drivers. For each network, you must use the MBX Driver Configuration editor to create an MBX device. These are logical network devices that may use a serial port, Ethernet port or adapter card to connect to a network. Your software applications—including the MBX Bridge—use the MBX devices for their network communications.



MBX Driver Configuration Editor showing five MBX devices.

After you create the MBX devices, you must configure them as bridge devices. This is simply the process of telling the MBX Bridge software which MBX devices you want to use to route messages. The bridge software can automatically configure all of the MBX devices as bridge devices, if you wish. Alternatively, you can configure the bridge devices manually.



*Bridge Device Configuration:
All five MBX devices have been configured for use by the bridge.*

The bridge devices identify the networks that the bridge can use to route messages. The next step is to set up the actual routings.

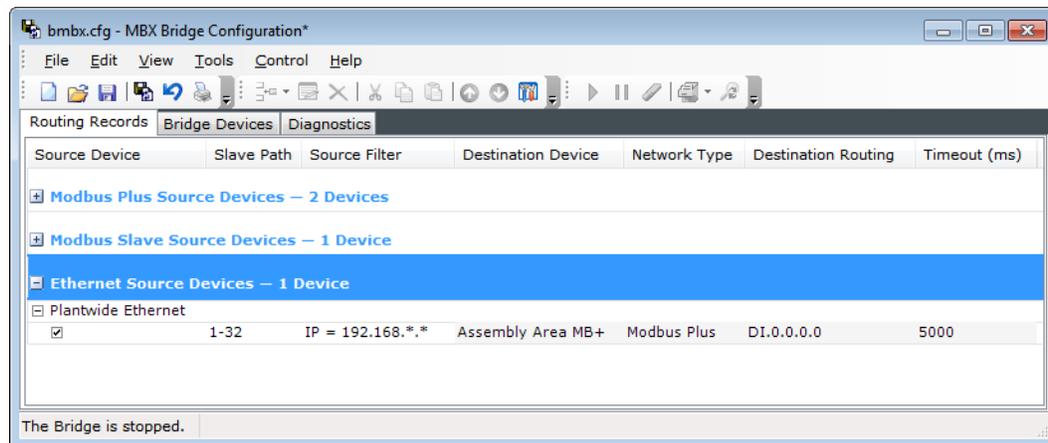
Why do I need routing records?

Routing records serve two crucial functions:

- *They specify which networks will be bridged.* The bridge devices specify only which networks you want to use. The bridge can route messages from any device to any other device, but you may not want or need to permit nodes on every network to send messages to every other network. The routing records allow you to designate, for each device, which of the other devices it can route messages to.
- *They convert one form of address to another.* Each network protocol has its own addressing techniques, and the bridge must be able to convert one format to another. Suppose you want to route a message from a Modbus Plus node to an Ethernet node. The bridge must be able to take the information in the Modbus Plus routing array, and use that to determine the proper Ethernet IP address and destination index. The routing records allow you to control how the bridge does that.

What does a routing record look like?

All routing records have a common structure, but the details will vary for each combination of device types. The routing record in the illustration below is typical. It routes messages from an Ethernet device to a Modbus Plus device.



The first three columns specify the incoming messages that will be routed by this record.

- The Source Device is the bridge device that accepts the message to be routed.
- The Slave Path field defines the path or paths that will use this routing record. Arriving messages can use any of the slave paths available in the bridge, but you may want to route only messages that use certain paths. In this case, the routing record will apply only to messages using slave paths 1 through 32.
- The Source Filter field allows you to limit the scope of the routing record even more. Ethernet messages will report the IP address of the node that sent them, so you can restrict the record to certain ranges of IP addresses. In this case, the field specifies the first two IP address bytes, and allows any value for the last two bytes.

To be routed by this routing record, a message must pass all three criteria. It must be from the correct bridge device, use a slave path in the specified range, and match the requirements of the Source Filter. If a message does not meet all three criteria, it will not be routed by this routing record. It will be compared against each of the other routing records until a match is found and it can be routed. If it doesn't match any of the routing records, the bridge will send an error response to the sending node.

Messages that pass the source criteria are routed according to the specification in the next four fields.

- The Destination Device is the bridge device to which the message will be sent.
- The Network Type displays the type of network that the Destination Device connects to.
- The Destination Routing specifies the address to which the message will be sent. In the example shown, the bridge uses the value of the received

destination index as the first routing byte of the Modbus Plus address, with zeros for the remaining bytes.

- The Timeout field specifies how long the bridge will wait for a response after it sends the message to its destination.

Thus, any node on the "Plantwide Ethernet" network can send messages to the "Assembly Area MB+" Modbus Plus network, provided that the sending node has an IP address beginning with 192.168. It can send the message to Modbus Plus nodes 1-32 by specifying that value as the destination index of the message.

Notice, however, that the Ethernet nodes could not send a message to a node at Modbus Plus address 47. To do so, they would have to use a destination index of 47, which in the default case would use slave path 47. But the routing record's Slave Path criterion accepts only slave paths 1-32. Such a message would not be routed by the bridge. In this way, you can block access to certain nodes, if you wish.

Note

The slave path used is the same as the destination index only for the default Ethernet driver configuration. It is possible to remap this relationship, but the default is the most commonly used setup. In these examples, we assume the default. For details on how to remap the slave paths, refer to the Ethernet MBX Driver Help.

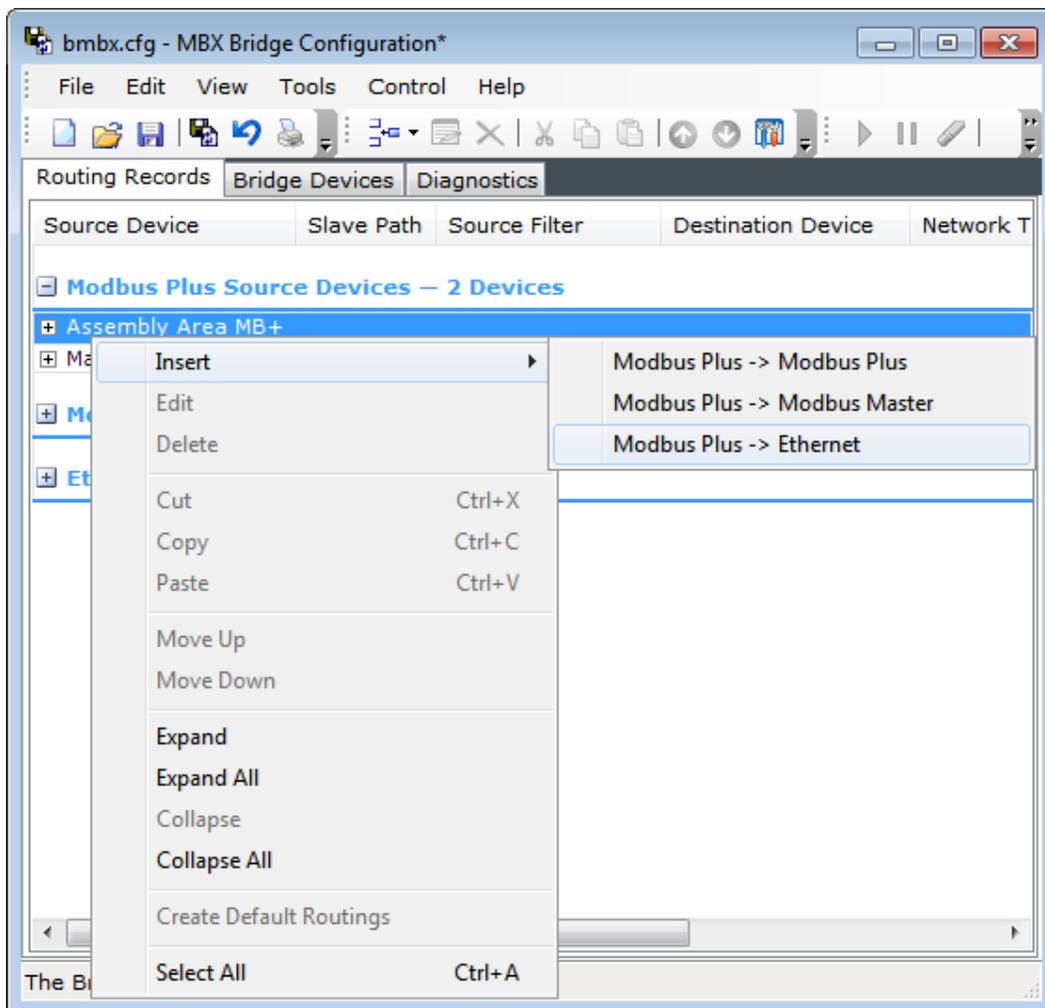
How do I decide what to put in a routing record?

The routing records are the key to your bridge application, because they determine which messages get routed and where each message goes. It is not difficult to design a routing strategy, but careful planning will help you to quickly create a successful system.

Source and Destination Devices

First, you must decide which networks must communicate with each other. If you have more than two networks, you probably won't need to have every network communicating with every other network. In that case, there is no reason to configure routings between the networks that don't need to exchange messages.

Keep in mind that each routing record describes a one-way routing, from a source device that initiates a command message to a destination device whose response is passed back. Communicating both ways requires at least two routing records. On the other hand, if messages will be passed only in one direction, there is no need to create a record to route them in the other direction.



Suppose you want to create a routing record to pass messages from a Modbus Plus network to an Ethernet network. The Modbus Plus device is then the *Source Device*. When you add a routing record under that device, you must choose ***Modbus Plus -> Ethernet***. This will allow you to select the desired Ethernet Destination Device.

| <u>Source</u> | | | <u>Destination</u> | | |
|-------------------|------------|---------------|--------------------|--------------|---------|
| Device | Slave Path | Source Filter | Device | Network Type | Routing |
| Assembly Area MB+ | | | Plantwide Ethernet | Ethernet | |

Once you select the desired Destination Device, your routing record configuration will look like this.

The next step is to decide how to filter the messages so that only those you wish to route will be selected. You do this with the Slave Path and Source Filter fields.

Slave Path Filter

One way to filter messages is to use the Slave Path filter. The source node provides routing information that specifies which slave path the bridge will use to handle the message:

- For Ethernet sources, the destination index specifies the slave path.
- For Modbus sources, the Modbus node address specifies the slave path.
- For Modbus Plus sources, the second byte of the Modbus Plus routing array specifies the slave path.

You can specify that the routing record should accept a single slave path, a range of slave paths, or any slave path.

| <u>Source</u> | | | <u>Destination</u> | | |
|----------------------|-------------------|----------------------|---------------------------|---------------------|----------------|
| Device | Slave Path | Source Filter | Device | Network Type | Routing |
| Assembly Area MB+ | 1-8 | | Plantwide Ethernet | Ethernet | |

For most Modbus Plus configurations, you will want to accept any slave path, and we will do that for this example. Modbus Plus devices can use slave paths 1-8, so that is the value we will place in the Slave Path field.

Source Filter

Most often, all of the filtering is handled by the Source Filter, rather than by the Slave Path filter. It is critically important to understand what you are filtering here:

- For Modbus sources, the Source Filter acts on the Modbus node address.
- For Modbus Plus sources, the Source Filter acts on the Modbus Plus routing array associated with the message.
- However, for Ethernet sources, the Source Filter acts on the IP address of the node that sent the message, and not the IP address to which the message was sent.

As with the Slave Path filter, you can accept a single value, a range of values or any value. You may do this for each of the bytes in the routing.

| <u>Source</u> | | | <u>Destination</u> | | |
|----------------------|-------------------|----------------------|---------------------------|---------------------|----------------|
| Device | Slave Path | Source Filter | Device | Network Type | Routing |
| Assembly Area MB+ | 1-8 | *.*.10-35.0.0 | Plantwide Ethernet | Ethernet | |

This example uses a Modbus Plus source. The first of the five bytes in the Modbus Plus routing array will always be the node address of the bridge, so that is not useful for filtering. We can set that to an asterisk, so that it will accept any value.

The second byte, as you recall, is the slave path. Slave Path filtering was already configured in the Slave Path field, so the only configuration that makes sense for this byte is to accept whatever was passed by the Slave Path filter. Therefore, the bridge automatically sets this value to *, indicating that any value is acceptable.

The last three bytes of the routing array are where the real filtering takes place. These bytes are not needed for routing the message on the Modbus Plus network, so they can be used to specify the target node on the Ethernet. Suppose that we want to permit the Modbus Plus nodes to send messages to Ethernet nodes with IP addresses ranging from 192.168.0.10 through 192.168.0.35. We could simply have the Modbus Plus nodes set the third routing array byte to the desired value for the last IP address byte. We would then expect the third Modbus Plus routing byte to be in the range 10-35, with the last two routing bytes set to 0. This gives us the rest of the Source Filter entry.

With the Source section complete, we have now identified the messages we want to route. The last step is to specify the destination routing.

Destination Routing

The Destination side of the routing record tells the bridge where to send the messages that pass the criteria on the Source side.

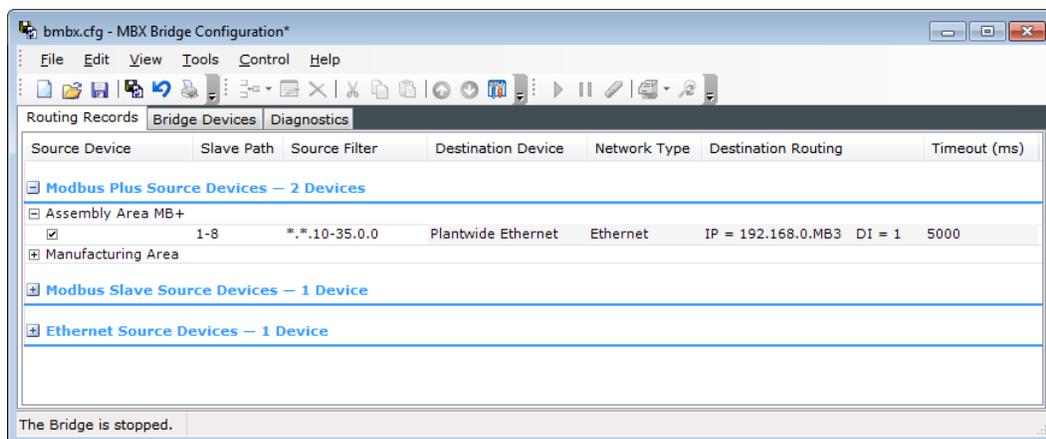
| <u>Source</u> | | | <u>Destination</u> | | |
|----------------------|-------------------|----------------------|---------------------------|---------------------|---------------------------|
| Device | Slave Path | Source Filter | Device | Network Type | Routing |
| Assembly Area MB+ | 1-8 | *.*.10-35.0.0 | Plantwide Ethernet | Ethernet | IP = 192.168.0.MB3 DI = 1 |

We have already chosen the Plantwide Ethernet network to be the Destination Device. The Network Type field is an informational field that is automatically filled by the editor. Only the Routing remains to be specified.

The first three bytes of the IP address in this example are always 192.168.0, so we can specify those values directly.

We know that the third byte of the message's Modbus Plus routing array contains the last byte of the IP address. The bridge allows us to use the operand MB3 to refer to the third byte Modbus Plus routing array byte. Thus, we can specify the IP address as ***192.168.0.MB3***.

We must also specify the destination index. In this case, we might decide that we can simply set it to ***1***. The routing record is now complete. The result is shown in the figure below.



To allow the Ethernet nodes to send messages to the Modbus Plus nodes, we must create another record with **Plantwide Ethernet** as the Source Device and **Assembly Area MB+** as the Destination Device.

Can I use more complex expressions in the Destination Routing?

Yes. Each byte in the Destination Routing can be an expression in one of the following forms:

- operand
- operand + operand
- operand – operand

where the permitted value for *operand* depends on the type of the source device.

For Ethernet source devices, *operand* may be:

- A constant value.
- SP, the slave path used for the received message.
- DI, the destination index value of the received command message.
- IP1, IP2, IP3 or IP4, a byte of the source device's IP address.

For Modbus Plus source devices, *operand* may be:

- A constant value.
- SP, the slave path used for the received message.
- MB1, MB2, MB3, MB4 or MB5, a byte of the routing path as received in the command message.

For Modbus source devices, *operand* may be:

- A constant value.
- SP, the slave path used for the received message.
- MB, the destination Modbus address as received in the command message.

If you use anything other than a constant, be sure that, at run time, the expression will return a valid value for the network. If it is invalid, the bridge will not route the message, but will instead return an error message to the source node.

Can I bridge between two like networks?

Absolutely. You can set up Modbus to Modbus, Ethernet to Ethernet, and Modbus Plus to Modbus Plus routing records just like any other. By doing this, you can configure the MBX Bridge to work like a standard bridge for one of these network types.

This can be particularly useful where you have two like networks that must exchange data, but which have address conflict problems. For example, suppose you have two Modbus Plus networks that you would like to interconnect, but you find that several nodes on one network have the same address as nodes on the other network. If you try to join them into a single network, you must change the node addresses that conflict and then change all of the applications that refer to the nodes that you just reassigned. This could be a lot of work and will likely result in addressing errors. Instead, you can join them with an MBX Bridge. All of the nodes on both networks can retain their current addresses while still being able to access the nodes on the other network.

Your routing records might look like this:

| <u>Source</u> | | | <u>Destination</u> | | |
|------------------------|------------|---------------|------------------------|--------------|-----------------|
| Device | Slave Path | Source Filter | Device | Network Type | Routing |
| Manufacturing Area MB+ | 1-8 | *.*.*.* | Assembly Area MB+ | Modbus Plus | MB3.MB4.MB5.0.0 |
| Assembly Area MB+ | 1-8 | *.*.*.* | Manufacturing Area MB+ | Modbus Plus | MB3.MB4.MB5.0.0 |

When a node on one Modbus Plus network wants to send a message to a node on the other network, it sends the message to the MBX Bridge, placing the desired destination node address in the third Modbus Plus routing byte. The bridge accepts all messages and routes them to the specified node on the other network.

Notice that the Destination Routings also send the fourth and fifth Modbus Plus routing array bytes as part of the destination routing. This allows you to route messages to other bridges on the destination network and, through them, to additional layers of networks.

WHERE CAN I GET MORE INFORMATION?

You can get detailed information on how to install, configure and use Cyberlogic's MBX Bridge by referring to the help files for the MBX Bridge Suite.

Cyberlogic's website, www.cyberlogic.com, has information on related products, news, software downloads and contact information.

Cyberlogic Technologies
5480 Corporate Drive
Suite 220
Troy, Michigan 48098 USA

Sales: 248-631-2200
sales@cyberlogic.com

Technical Support: 248-631-2288
techsupport@cyberlogic.com

Copyright © 2005-2010, 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™ and DevNet™ are trademarks of Cyberlogic Technologies Inc. All other trademarks and registered trademarks belong to their respective owners.