User's Manual

Notice of Alterations

MW100 Communication Command

Please make the following alterations to the Communication Command Manual IM MW100-17E.

■ For the MW100 with Firmware Version R3.03

The following changes have been made to procedures due to changes in functions.

Page 1-19 Note the addition of the underlined portion under "XD Relay Settings"

••

When p2=ALARM, p4=OFF, or p5=OR

`XDp1,p2,p3,p4,p5,p6,p7,p8`<terminator>

- `p1` Relay range (001 to 060)
- `p2` Relay output type
  - `ALARM` Alarm
  - `COM` Communication input
  - `MEDIA` Media free space
  - `FAIL` FAIL output
  - `ERROR` Error output
- `p3` Relay energized/de-energized
  - `ENERG` Energize
  - `DE_EN` De-energize
- `p4` Relay status hold (OFF/ON)
- `p5` Relay operation conditions
  - `AND` Operate when all specified alarms are on alarm status.
  - `OR` Operate when at least one of the specified alarms is alarm status.
- `p6` Reflash function (OFF/ON)
- `p7` Preset value upon error
  - `OFF`
  - `OPEN` Opens (de-energizes) the output relay upon error
  - `CLOSE` Closes (energizes) the output relay upon error
- `p8` Preset value upon stop
  - `OFF`
  - `OPEN` Opens (de-energizes) the output relay upon stop
  - `CLOSE` Closes (energizes) the output relay upon stop

- Reflash can only be selected if the relay status is non-hold (OFF) and the relay operation condition is OR.
- The preset value (OPEN/CLOSE) is only valid when the DO type is alarm DO, and the condition is: "relay operation is non-hold" OR "reflash is OFF."
- When the preset value (OPEN/CLOSE) occurs, priority is given over the alarm relay output, and the preset output is carried out.

••

Page 1-19 Note the changes of the underlined portion to "SO Expression Settings"

`p3` Expression (A001 to A060: max. 120 characters, A061 to A300: max. 10 characters)

Page 1-22 Note the addition of the underlined portion under "AP Preset Operation Settings"

Setting `APp1,p2,p3,p4`<terminator>

- `p1` Channel range (001 to 060)
- `p2` Operation upon startup
  - `LAST` Hold previous value
  - `PRESET` Output preset value
- `p3` Operation upon errors
  - `LAST` Hold previous value
  - `PRESET` Output preset value
- `p4` Operation upon measurement stop, computation stop, or transmission stop
  - `LAST` Hold previous value
  - `PRESET` Output preset value

Page 1-25 Note the addition of the underlined portion under "XG Settings of Operation upon MATH Error" "p4 Sum scale of the TLOG.SUM computation"

HOUR Sum value of the data for each measurement interval/number of computations per hour

DAY Sum value of the data for each measurement interval/number of computations per day

Page 1-41 Note the change of the underlined portion under "FL Output of Logs, Alarm Summaries, and Status" "When Outputting Logs"

RECORD 1 to 250

Page 2-10 Note the change of the underlined portion under "Output of Recording Logs (FL0, RECORD)"

- The data acquisition log is output. Up to 250 past data acquisition logs are retained. If that number is exceeded, old logs are overwritten with new ones.

Page App-2 Note the addition of the underlined portion under "Modbus Protocol Function Code"

Slave/Server Functions

The slave function of the main unit does not support broadcasted commands.

Function codes 1, 2, 5, and 15 have been added to the Modbus Slave function.

Function Code	Function	Action
1	Read coils (0xxxx)	MW100 reads the statuses of coils.
2	Read input relays (1xxxx)	MW100 reads the statuses of input relays.
3	Read hold registers (4XXXX)	MW100 read communication input data 16 written by function code 6 or 16
4	Read Input registers (3XXXX)	MW100 reads the main instrument's measured, computed, and time data.
5	Write single coil (0xxxx)	MW100 turns status of a single coil.
6	Write hold registers (4XXXX)	MW100 writes to the main instrument's communication input data.
8	Loop back test	MW100 performs the loop back test on the main instrument. Main instrument only support message return (diagnostic code (0x00)).
15	Write multiple coils (0xxxx)	MW100 turns status of multiple coils.
16	Write hold registers (4XXXX)	MW100 writes to the main instrument's communication input data.

Page App-3 Note the addition of the underlined portion and the following to "Register Assignments (Modbus Slave)."

The following are the Modbus slave register assignments.

Coils for mapping data on communication input channels (C001 to C300) and data on measurement and computation channels (001 to 060, and A001 to A300) have been added to the Modbus slave function. Input relays and input registers for indicating main unit mode information have also been added.

Coil	Data	Data type
00001	Data on communication input channel C001	Bit
00002	Data on communication input channel C002	Bit
:	:	:
00300	Data on communication input channel C300	Bit

Input relay	Data	Data type
10001	Data on measurement channel 001	Bit
10002	Data on measurement channel 002	Bit
:	:	:
10060	Data on measurement channel 060	Bit
13001	Data on computation channel A001	Bit
13002	Data on computation channel A002	Bit
:	:	:
13300	Data on computation channel A300	Bit
18001	Measuring	Bit
18002	Computing	Bit
18003	Recording	Bit
18004	Alarm occurring	Bit
18005	Waiting to confirm alarm status	Bit

Absolute Address	Relative Address	Allocation	Data Type
38001	8000	Measuring (0, 1)	Int 16
38002	8001	Computing (0, 1)	Int 16
38003	8002	Recording (0, 1)	Int 16
38004	8003	Alarm occurring (0, 1)	Int 16
38005	8004	Waiting to confirm alarm status (0, 1)	Int 16

Page App-4 Note the addition of the following after "Register Assignments (Modbus Slave)."

Notes the values per the Modbus registers for the special values held by the MW100 measurement and computation channels.

Special values held by measurement and computation channels

Value type	Name/description of value	Value per Modbus register		
		Int 32	Float	Bit*1
+Over	+Over Data value greater than the upper limit	2,147,450,879 (0x7fff 7fff)	+Inf(0x7f80 0000)	0
–Over	–Over Data value smaller than the lower limit	–2,147,385,343 (0x80001 8001)	–Inf(0xff80 0000)	0
Skip	Skip Channel disabled	–2,147,319,806 (0x8002 8002)	–NaN(0xff80 0002)	0
No Channel	No channels No channels exist	–2,147,254,269 (0x8003 8003)	–NaN(0xff80 0003)	0
Error	Error Error in computation of data value*2	–2,147,188,732 (0x8004 8004)	–NaN(0xff80 0004)	0
Invalid	Undefined Data value is undefined*3	–2,147,123,195 (0x8005 8005)	–NaN(0xff80 0005)	0
Lack	Computation omitted Computation not processed	–2,147,057,658 (0x8006 8006)	–NaN(0xff80 0006)	0
Valid	Data	Data*4 (mantissa)	Data(physical qty.)	Data(0, 1)

*1 The value of the bit alone is insufficient to determine whether it is a special value. Mode (18000's, 38000's) and other information must also be used to determine whether it is a special value or other value.

*2 If the data to be computed or the computed result is NaN, an error occurs. Furthermore, data resulting in an error is replaced with +Over or –Over according to the "Operation upon Computation Error" setting.

*3 After turning the power to the MW100 ON, data values are undefined (Invalid) until initial measurement and computation are finished (until the first data of measurement and computation channels is created).

*4 Only the mantissa of values can be acquired from Int 32 format data of measurement and computation channels. Decimal place information of some form must be acquired separately from the MW100 in order to convert to physical values.

Note

This is an example of acquiring decimal place information.

- For measurement channels
  - In the Web browser's Input Range Setting screen, check the values in the Decimal columns under Scale (when scaling is not used, the decimal place of the value in the Span column).
  - Acquire channel information using an SR command query of the MW100-specific communication service.
  - Output measurement channels using the FD command of the MW100-specific communication service, then check the "Exponent" value.
- For computation channels
  - In the Web browser's "Expression Settings" screen, check the value in the Decimal column under Span.
  - Acquire channel information using an SO command query of the MW100-specific communication service.
  - Output computation channels using the FD command of the MW100-specific communication service, then check the "Exponent" value.

■ For the MW100 with Firmware Version R3.02

The following changes have been made to procedures due to changes in functions.

Page 1-31 "1.6 Setting Commands (Communication Related)"

YM **Mail Client Settings**

Setting **When p5=OFF**

YMp1,p2,p3,p4,p5<terminator>

When p5=POP3 or AUTH

YMp1,p2,p3,p4,p5,p6,p7<terminator>

p1 SMTP server name (up to 64 characters)
p2 SMTP port number (1 to 65535)
p3 POP3 server name (up to 64 characters)
p4 POP3 port number (1 to 65535)
p5 Use POP before SMTP (OFF/POP3/AUTH)
OFF
POP3 Use POP before SMTP
AUTH Use SMTP Authentication
p6 User name (up to 32 characters)
p7 Password (up to 32 characters)

Query YM?

Example Enable POP before SMTP in the mail client settings.
YMsmtp_server.abc.com,25,pop3_server.abc.com,110,POP3,YOKOGAWA,1234

Explanation
- For the characters that can be used for the SMTP server and POP3 server, see host and domain name strings in section 1.3.
- For the characters that can be used for passwords, see password strings in section 1.3.
- Password is encrypted during a query.

Page 1-40 "1.8 Output Commands"

FL **Output of Logs, Alarm Summaries, and Status**

Setting **When p1=0**

FLp1,p2,p3<terminator>

When p1=1

FLp1,p2,p3,p4<terminator>

When Outputting Logs

Setting FLp1,p2,p3<terminator>

p1 Output type (0)
0 Log output
p2 Data type

MODBUS_T	Modbus server log
CIP	CIP server log
PCCC	PCCC slave log
MATH	Computation status

p3 Max. number of read blocks

MODBUS_T	1 to 50
CIP	1 to 50
PCCC	1 to 50
MATH	1

Page 2-20 “Mail Client Error Messages (cont.)”

Error Code	Description	Corrective Action
EHELO	HELO command com failed	Check whether SMTP server is functioning correctly.
EMAILFROM	MAILFROM command com failed	Check whether SMTP server is correctly.
ERCPTTO	RCPTTO command com failed	Check whether SMTP server is functioning correctly.
EDATA	Data communication failed	Check whether SMTP server is functioning correctly.
ETCPIP	Internal processing error	Servicing required.
EINVAL	Internal processing error	Servicing required.
EPOP3HOSTNAME	SMTP server name not correct	Check whether destination server name is set correctly.
EPOP3UNREACH	POP3 server not found	Check whether destination POP3 server is running.
EPOP3TIMEOUT	Communication with the POP3 server timed out	Check cable connections and power of HUB.
EPOP3AUTH	Authorization on POP3 server failed	Check whether user name and server name are allowed on server.
ESMTPAUTH	SMTP Authentication failed	Check whether user name and server name are allowed on server.
EANOTSUPPORT	Attempted to connect to a server requesting an encoding algorithm that is not supported by the MW100	Please change servers.

Please note the addition of the following page.

Page 2-33 “2.2 ASCII Output”

Output of CIP Server Log (FL0, CIP)

- Output using the FL command.
- The CIP server information is output.

Syntax

```
EACRLF
yy/mo/dd_hh:mm:ss_d:_mmm...mCRLF
ENCRLF
```

yy	Year (00 to 99)
mo	Month (01 to 12)
dd	Day (01 to 31)
hh	Hour (00 to 23)
mm	Minute (00 to 59)
ss	Second (00 to 59)
d	Command/response (C/R)
mmm...m	Message (CIP command/response)
_	Blank

Example

```
EA
99/05/11 12:20:30 C: Register Session
99/05/11 12:20:31 R: Success
EN
```

Output of PCCC Slave Log (FL0, PCCC)

- Output using the FL command.
- The PCCC slave information is output.

Syntax

```
EACRLF
yy/mo/dd_hh:mm:ss_d:_mmm...mCRLF
ENCRLF
```

yy	Year (00 to 99)
mo	Month (01 to 12)
dd	Day (01 to 31)
hh	Hour (00 to 23)
mm	Minute (00 to 59)
ss	Second (00 to 59)
d	Command/response (C/R)
mmm...m	Message (PCCC command/response)
_	Blank

Example

```
EA
99/05/11 12:20:30 C: PLC-5 Typed Read, N10:0
99/05/11 12:20:31 R: Success
EN
```

■ MW100 DAQMaster EtherNet/IP Instruction Manual

Please add the following contents to MW100 User's Manual.

Trademarks

EtherNet/IP is a trademarks of ODVA.

Contents

1. Introduction

EtherNet/IP™
conformance tested

There is a large installed base of industrial automation Programmable Logic Controllers (PLCs) and remote I/O that support EtherNet/IP (EIP) also known as Control and Information Protocol (CIP) over Ethernet. Most notable is the family of PLCs and I/O manufactured by Allen-Bradley (AB) consisting of such controllers: PLC 2, PLC 5, SLC 500, MicroLogix, CompactLogix, and ControlLogix. Yokogawa's MW100 DAQMaster is a multi-protocol data acquisition, recording, and reporting device that supports communications to EIP devices. It is now possible for the MW100 to record data directly from EIP device inputs and registers over an Ethernet network. The MW100 may also act as remote I/O for EIP PLCs and Human Machine Interfaces (HMIs).

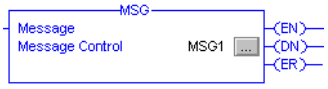
When using EIP, the MW100 is a passive device on the Ethernet network in that it does not initiate read or write requests. In most cases the MW100 will be working in conjunction with a PLC or controller. Read and write requests via EIP are initiated through program logic in controllers via Explicit Messaging and I/O Messaging (also known as Implicit Messaging). With the controller managing communications, it is possible to integrate messaging such that communications only occurs when dictated by the control logic. Management of communications by the controller allows the controller to decide when it is appropriate to write a value to the MW100 (e.g. when a computation is complete).

The MW100 also supports older controllers where EIP was not a standard option. For older controllers that support Programmable Controller Communication Commands (PCCC) also known as DF1 communications via serial ports, an inexpensive gateway can be used to convert communications to EIP. For controllers that support PCCC encapsulated via Ethernet, the MW100 supports EIP with embedded PCCC read and write requests.

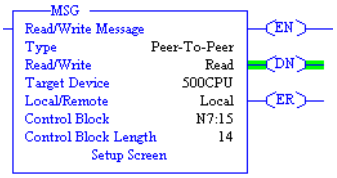
For example purposes RSLogix 5000, RSLogix 500, CompactLogix, and SLC 504 are used as examples of EIP configuration software and hardware – but any EIP software and hardware should work in a similar manner as long as they support Explicit and I/O (Implicit) messaging.

2. Explicit Messaging

Explicit Messaging is a point to point, request/response messaging protocol for unscheduled information transfer. In ladder logic programming explicit messaging is usually denoted by a messaging command that is all inclusive (what is going to be read or written and to what register in what device).



Example message command from RSLogix 5000 as used with CompactLogix PLC



Example message command from RSLogix 500 as used with SLC 500 PLC

The tables below detail the naming convention to use when creating explicit messages within RSLogix. In the left most column is the native MW100 register starting with I/O channels – 001 through 060, Computation Channels A001 through A300 (also known as Math Channels), and finally Communication Channels C001 through C300. I/O Channels and Computation Channels are considered read only while Communication Channels are read/write. To the right of the MW100 register are the naming conventions that are used within RSLogix messages. In an RSLogix 500 message, N10:0 would be used to retrieve an MW100's I/O Channel 001 as a 16-bit integer. In an RSLogix 5000 message, real[3299] would be used for read or write requests of an MW100's Communication Channel C300 as real. If a message was used to read Computation Channel A060 as long integer from an MW100, RSLogix 5000 would use dint[2059] as the Source Element.

File number / Tag names for Expkicit messaging

I/O Channel (001 to 060, max. 60-CH)

Ch.	PLC2	PLC5 / SLC	CIP int	CIP dint	CIP real
001	1000	N, L, F10:0	int [1000]	dint [1000]	real [1000]
:	:	:	:	:	:
060	1059	N, L, F10:59	int [1059]	dint [1059]	real [1059]

Computation Channel (A001 to A300, max. 300-CH)

Ch.	PLC2	PLC5 / SLC	CIP int	CIP dint	CIP real
A001	2000	N, L, F20:0	int [2000]	dint [2000]	real [2000]
:	:	:	:	:	:
A300	2299	N, L, F22:99	int [2299]	dint [2299]	real [2299]

Communication Channel (C001 to C300, max. 300-CH)

Ch.	PLC2	PLC5 / SLC	CIP int	CIP dint	CIP real
C001	3000	N, L, F30:0	int [3000]	dint [3000]	real [3000]
:	:	:	:	:	:
C300	3299	N, L, F32:99	int [3299]	dint [3299]	real [3299]

With N file or CIP int tag, you can access to the data as short integer (word.)
With L file or CIP dint tag, you can access to the data as long integer (double word.)
With F file or CIP real tag, you can access to the data as real (float.)

When using RSLogix 5000 and RSLogix 500 there are different types of messages that correspond to the different type of PLCs. Everything from PLC 2 through an ControlLogix PLC can be communicated with using the MSG block. The following covers all the MSG instructions supported by the MW100 with EIP.

MW100 supports following MSG instructions

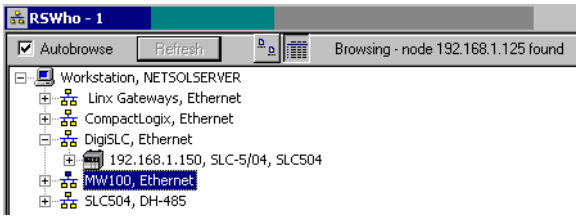
- PLC2 Unprotected Read/Write
- PLC5 Word Range Read/Write
- PLC5 Typed Read/Write
- SLC Typed Read/Write
- CIP Data Table Read/Write
- CIP Generic Read/Write

Step by step examples of explicit messaging within RSLogix 500 and RSLogix 5000 are detailed in section A-1.

3. I/O Messaging (Implicit Messaging)

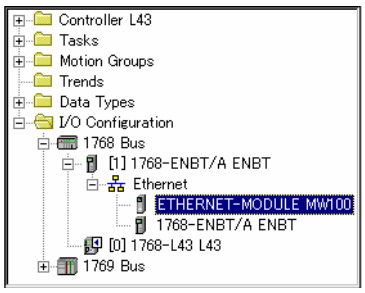
I/O Messaging also known as Implicit Messaging is used for point to point or multicast and are used to transmit application specific I/O data. Implicit messages are exchanged across I/O connections with a Connection Path (predefined path as first defined in RSLinx and then RSLogix). The Connection will define where the MW100 is located (IP Address), the Ethernet port on the PLC through which to communicate, as well as what points are considered inputs or outputs.

The following picture shows RSLinx setup to communicate to the Ethernet device MW100 (this connection points to an MW100 on the network).



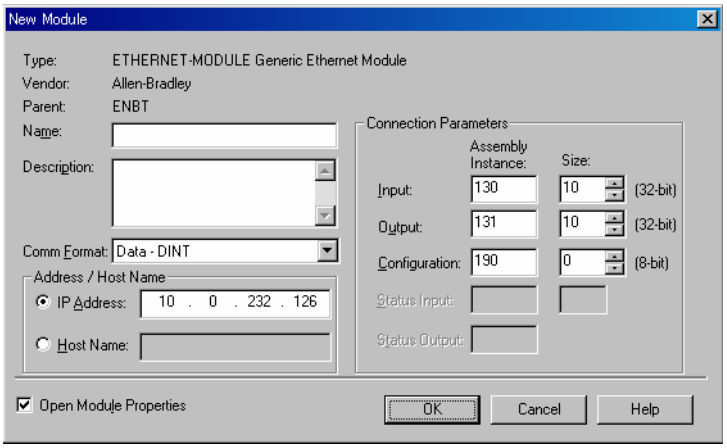
RSLinx Configured with path named MW100

Once a device is configured inside of RSLinx, it can be added to the RSLogix project. In the case below, ETHERNET-MODULE MW100 points to the networked MW100 configured as a Generic Ethernet Module.



Controller Organizer tree showing MW100 Path

By clicking on ETHERNET-MODULE MW100 within the Controller Organizer tree, the connection can be fully configured. Of note is the IP Address which should point to a MW100, Comm Format which defines what data types to use (MW100 supports only DINT type), and Connection Parameters which will layout the inputs and outputs of the MW100. In this case the connection is configured to communicate using double precision integers to an MW100 at IP address 10.0.232.126 with inputs at Assembly Instance 130 (corresponding to Communication Channel C001-C010; 10 Channels due to Size), and outputs at Assembly Instance 131 (corresponding to Communication Channel C101-C110; 10 Channels due to Size). The Size of the Input and Output Assembly Instance can range from 1 to 100 to encompass 100 Communication or Computation Channels in a block and 60 for I/O Channels. When using I/O Messaging, there is a limit of 100 inputs and 100 outputs per MW100 (Explicit Messaging must be used to reach more MW100 channels). Note, a limit of 125 32-bit points per instance is the maximum data size that EIP allows for I/O Messaging.



Connection and Assembly Instance Configuration in RSLogix 5000

The following diagram depicts the available Instance IDs, Sizes, and Data Types that can be configured within a connection.

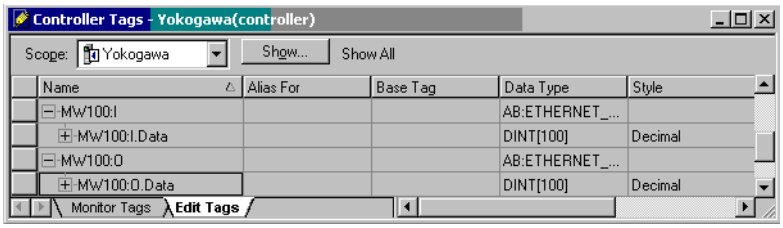
Assembly instances for I/O messaging

Channels in Assembly Object

Ch.	Kind	Instance ID	Size	Type
001 - 060	Producer	110	4 x 60	dint
A001 - A100	Producer	120	4 x 100	dint
A101 - A200		121	4 x 100	
A201 - A300		122	4 x 100	
C001 - C100	Producer/ Consumer	130	4 x 100	dint
C101 - C200		131	4 x 100	
C201 - C300		132	4 x 100	
	Configuration	190	0	
	Consumer	191	0	

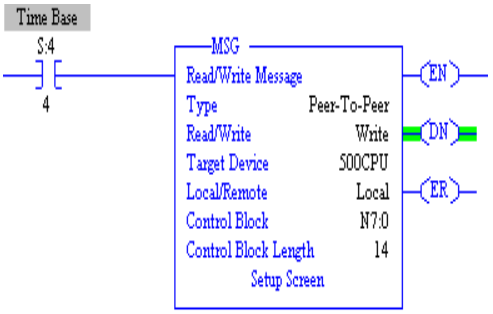
Table of MW100 channels with corresponding Instance ID.

Once an instance has been properly configured, the MW100 inputs and outputs will show up in the Controller Tags window.



MW100 Channels within RSLogix 5000 Controller Tags Screen.

These points can now be assigned as inputs and outputs as well as monitored (when online) within programs as shown in the example below. Note the tags can be used within any logic element (not just MSG blocks as with Explicit Messaging). In this case the logic is reading from the first input instance corresponding to Communication Channel C001 and writing to the output instance corresponding to Communication Channel C101.



Sample logic using I/O Messaging.

Step by step examples of implicit messaging within RSLogix 5000 are detailed in section A-1.

4. Specification

The following table describes how the MW100 conforms to the EIP specification. Note that when interfacing to the MW100 on an EIP network that no more than 10 connections can be active at any given time.

MW100 EtherNet/IP Model Specification

Spec.	Description
Implementation	Level 2 (Message Server + I/O Server)
Connection	Max. 20
Protocol	EIP/PCCC, EIP/native
Messaging	Explicit(UCMM, Clas3) + I/O(Class 1)
Object	Assembly, PCCC, Data Table
Data Exchange	Max. 300-CH(as integer or float data)
I/O	AI/AO, DI/DO(Max. 60-CH)
Sampling	100ms to 60s
Recording	Max. 360-CH(60 I/O + 300 Computation)

Table of MW100 Channels with corresponding Instance ID

5. Summary

The MW100 with EIP support can easily communicate via Explicit or I/O messaging to a variety of PLCs. The MW100 requires that the PLC initiate all communications. Now that the MW100 can communicate with EIP based PLCs, the full capabilities of the DAQMaster can easily be added to a controller network.

A PLC can use the MW100 as remote inputs and outputs within its control logic.

A PLC can write its inputs and register values into the MW100 Communication Channels (C001 through C300) so that the MW100 can record up to 300 PLC data points (Communication Channels must be placed into Computation Channels; only Computation Channels (A001 to A300) and I/O Channels (001 to 060) can be recorded on an MW100).

Full MW100 network service are available – including real-time web-pages for monitoring data values, FTP of data files, e-mail, as well as custom web-pages (layout the data with graphics and save on purchasing a standalone HMI).

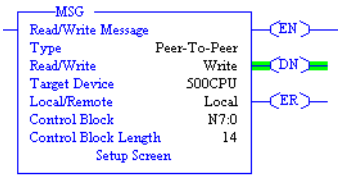
A-1. Detailed Explicit and I/O Messaging using RSLogix

The majority of devices that the MW100 will be connected to using EIP will be AB PLCs. RSLogix 500 or RSLogix 5000 are the programming packages used to configure and program everything from the legacy PLC 5 through the latest ControlLogix.

Explicit Messaging with RSLogix 500

The following assumes basic familiarity with RSLogix 500 and RSLinx and that both are installed and that RSLogix 500 is able to communicate through RSLinx to the designated PLC.

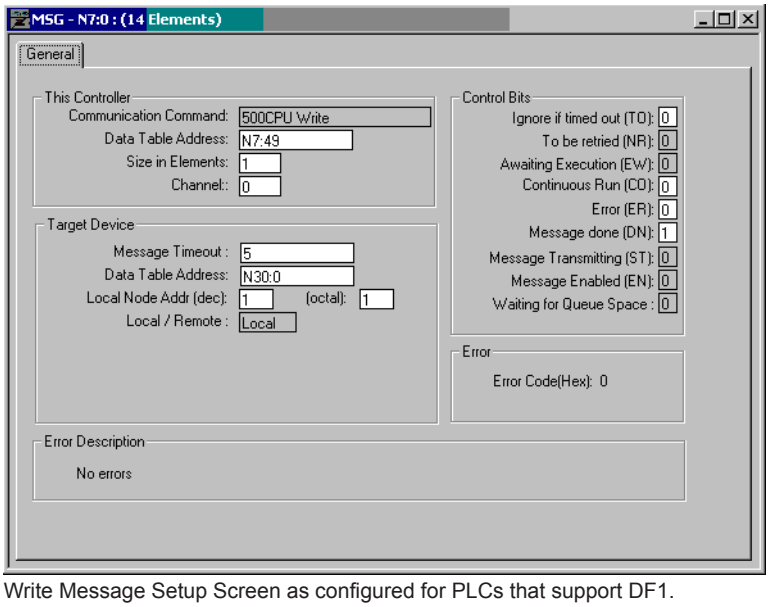
Messages are designated as MSG under the Input/Output tab of the instruction bar and may be inserted as the output of a rung of ladder logic. The MSG command can be used for reads or writes (in this case the example below shows a write message). The target device should be set to 500CPU when talking to SLC 500s and PLC 5 when communicating to older PLC 5s. Control Block is used to set the location in memory for the MSG function to be stored and it should be different from the Data Table Address used on the Setup Screen (what data should be written to the MW100).



RSLogix 500 Write Message

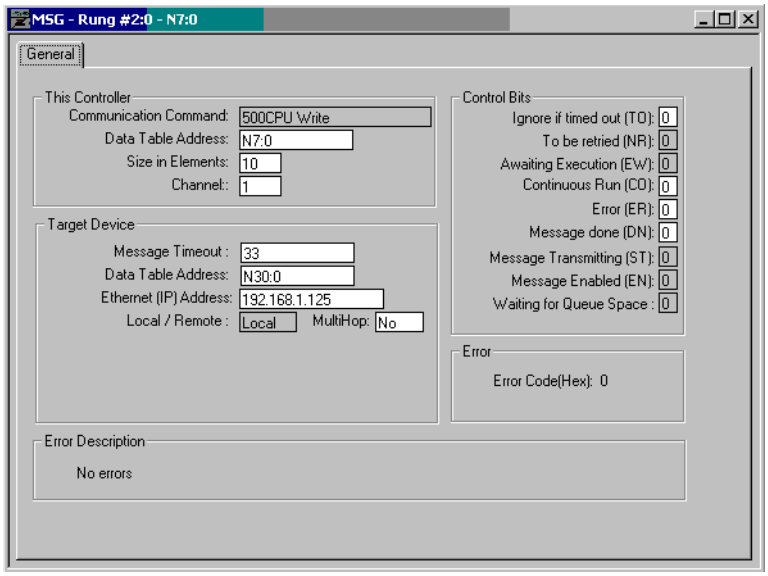
Once the Control Block is designated, the Setup Screen can be configured. The first item to fill in is the location of the data that is to be written from the PLC to the MW100, designated as the Data Table Address. In this case N7:49 is chosen with an element size of 1 (one byte of data - in order to read or write large amounts of data in a single message, increase the elements size to the appropriate value). Channel 0 designates what port to use on the PLC (in this case the serial port for DF1 communications – later routed via a DigiOne IAP (DF1 to EIP gateway).

The next step is to configure where the message will be written. In this case a Message Timeout of 5 seconds is used and Communication Channel C001 is being written to as an integer using the syntax N30:0 for Data Table Address. In this case Local Node Addr is set to 1 so that the gateway device knows to route all commands issued to Node 1 to the IP address of a specific MW100. If multiple MW100s are on a network then using different Node Addresses within the message commands can be used in conjunction with a gateway to route messages to specific MW100s (e.g. Node 1 to MW100 A, Node 2 to MW100 B, etc…).



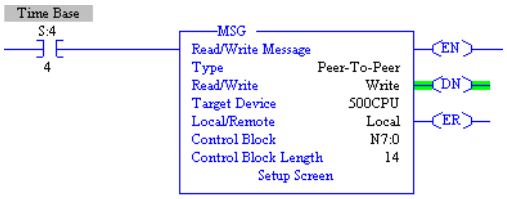
Write Message Setup Screen as configured for PLCs that support DF1.

When using RSLogix 500 with controllers such as the MicroLogix series that have on board Ethernet support, the Setup Screen looks slightly different as instead of a Node Address, direct input of the MW100 IP address is allowed (no gateway or DF1 to EIP routing is required in this case).



Write Message as configured in a MicroLogix or SLC 505 with EIP support

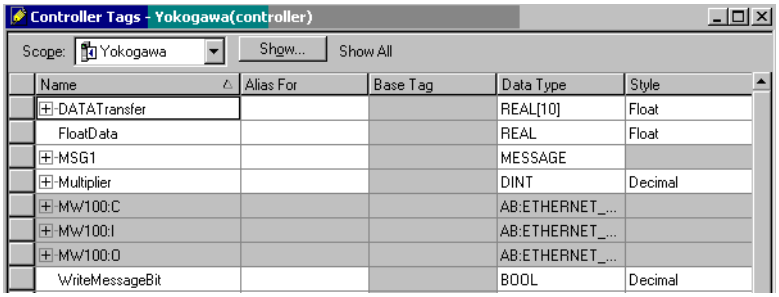
An explicit message should be triggered on/off by some sort of logic; the following image represents using the seconds bit of the PLC's clock to activate the message.



Time Based Message Write

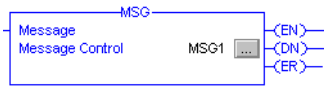
Explicit Messaging with RSLogix 5000

Explicit Messaging within RSLogix 5000 is similar to messaging in RSLogix 500 but there are a few differences; the first is that everything is simplified if tags are predefined. From the Controller Organizer (tree on left) pick Controller Tags and create a tag of Data Type MESSAGE (in this example tag MSG1). Also create a tag that will hold the PLC data that will be written to the MW100 (in this example tag DATATransfer which is a block of 10 floating point numbers). A Boolean bit to activate the message was also created as the tag WriteMessageBit.



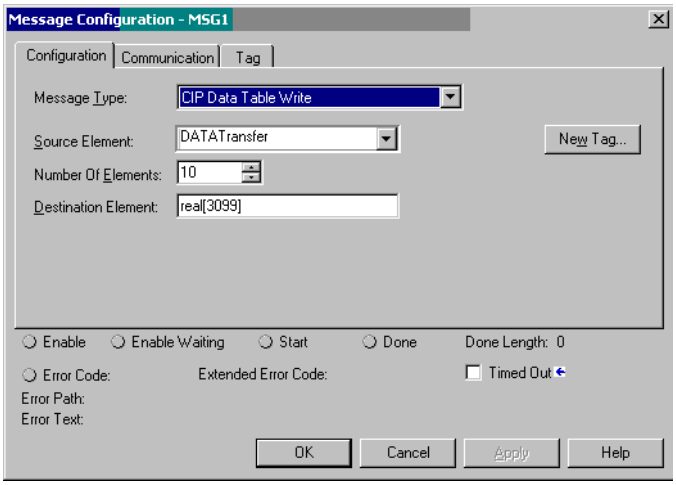
RSLogix 5000 Controller Tags Screen

The next step is to insert the message block from the Language Element Toolbar designated as MSG under the Input/Output tab. The MSG block can be inserted as the output of a rung. A controller tag of data type Message should be assigned to the MSG block, in this case tag MSG1.



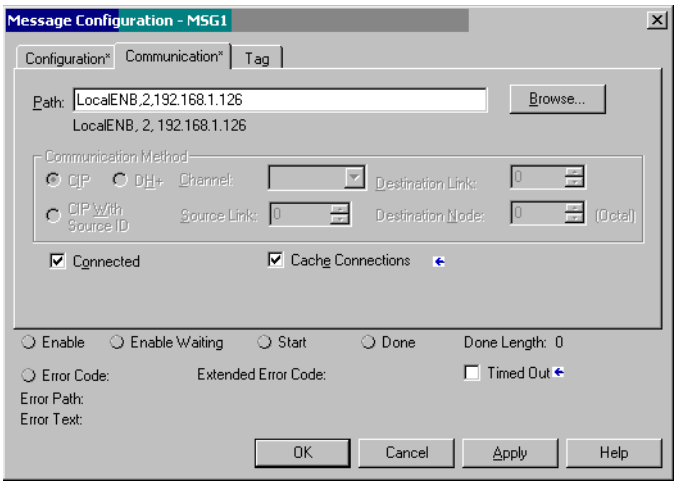
MSG block using tag MSG1

The next step is to configure the MSG block ([…] button). In this case the message block is configured to write data from the PLC to the MW100 and so Message Type is set to CIP Data Table Write. Source Element is set to DATATransfer (tag within PLC) and the Number of Elements is set to 10 (number of bytes of data - in order to read or write large amounts of data in a single message, increase the Number of Elements to the appropriate value). The Destination Element is set to real[3099] which corresponds to MW100 Communication Channel C100.

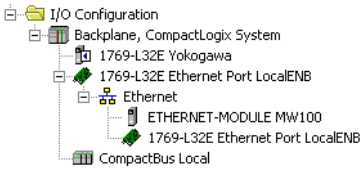


Message Configuration Tab

Next configure the Communication Tab by entering the Path to the MW100. The Path can be designated by the name of the Ethernet port on the PLC (in this case LocalENB – see I/O Configuration below) followed by a comma, with 2 (depth of communications) followed by a comma, and the IP address of the MW100 (e.g. 192.168.1.126). Check Connected and Cache Connections to speed up communications to the MW100.

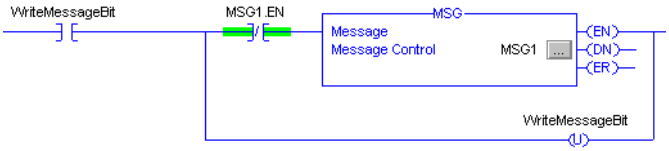


If the path to the MW100 is already configured in RSLinx then RSLogix will automatically replace the explicit path (e.g. LocalENB,2,192.168.1.126) with named path (e.g. MW100 as seen below beside attached ETHERNET-MODULE). If the MW100 has not been configured within RSLinx then ETHERNET-MODULE MW100 would not be present in the tree below and the explicit path on the Communication Tab will not be resolved and replaced with MW100.



I/O Configuration within Controller Organizer tree

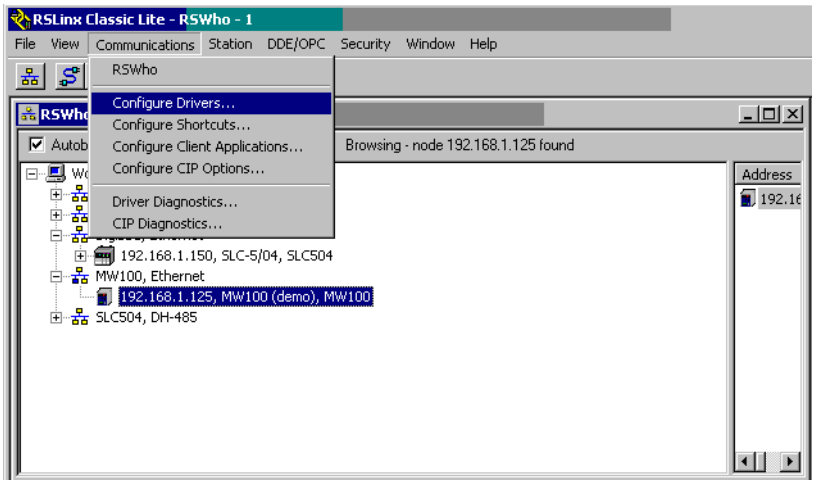
The last thing to do is to place some activation logic around the message to tell it when to write. In this case when the contact WriteMessageBit is toggled on the MSG block activates and writes to the MW100 and the WriteMessageBit is simultaneously toggled off with the output WriteMessageBit unlatch coil.



Contact Based MSG Logic in RSLogix 5000

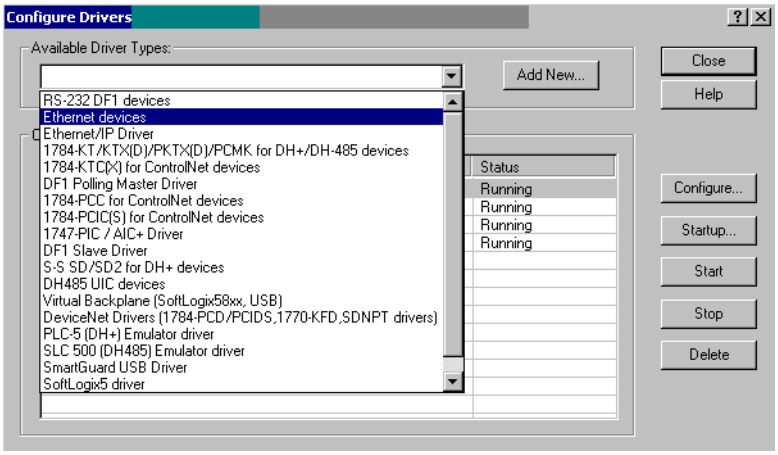
I/O Messaging with RSLogix 5000

The first step in configuring an MW100 to communicate via I/O Messaging is to define a connection within RSLinx. From the top menu under Communications, pick Configure Drivers.



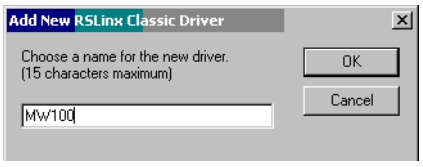
Configure Driver within RSLinx

The next step is to select Ethernet devices (not Ethernet/IP Driver) to support the MW100 and then pick Add New…



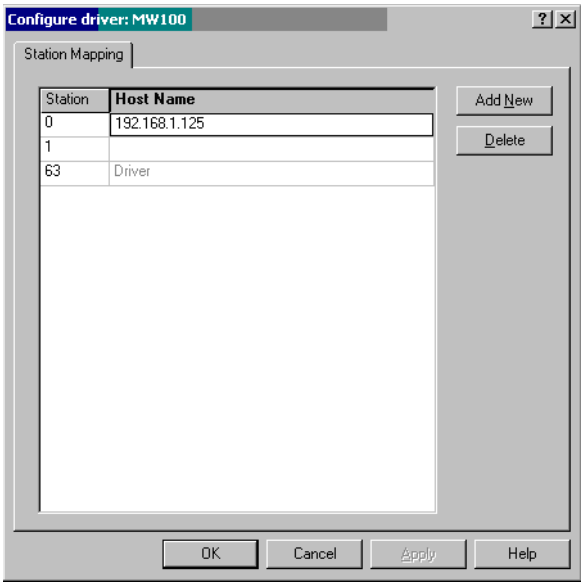
Ethernet devices driver for MW100

When prompted, name the driver – in this case MW100 was used but the name can be changed to suit different naming conventions.



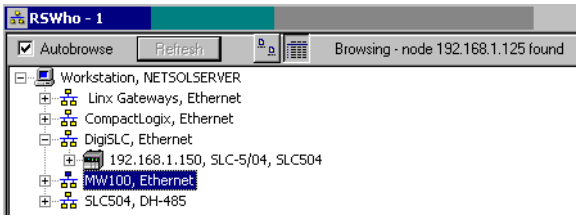
Path name for Ethernet Device

After the driver is named, enter the IP address of an MW100 and click OK to continue.



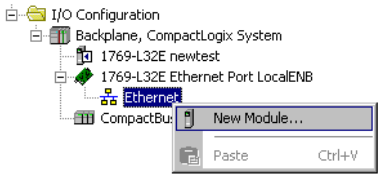
Configuring IP Address of MW100 within RSLinx

When properly configured there should be a new listing in RSLinx for MW100. Note that when browsing the connection, RSLinx indicates the node is found.



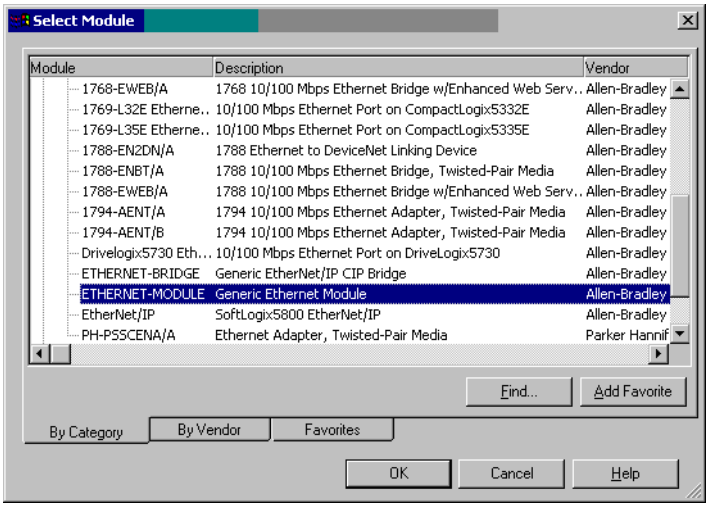
RSLinx with MW100 added

Open RSLogix 5000 and select the PLC that is going to communicate with the MW100. Right click on Ethernet and select New Module...



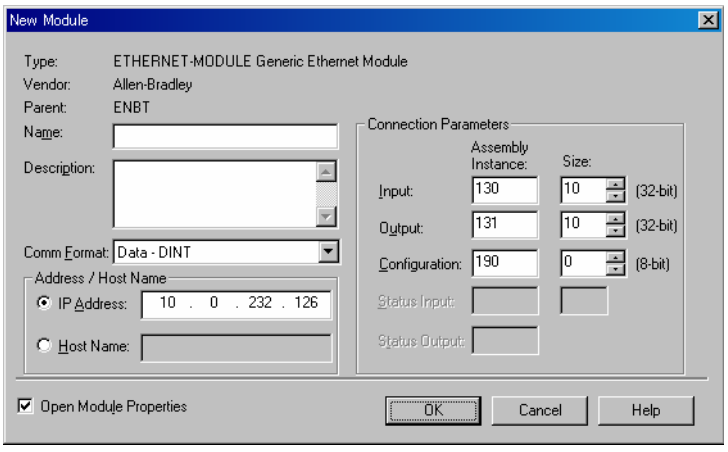
Adding a New Module to an RSLogix 5000 Project

Expand the Communications listing by clicking on the + sign and then scroll down and select ETHERNET-MODULE and click OK.



Selecting Generic Ethernet Module for MW100 Communications

A definition screen should now appear for the ETHERNET-MODULE. In the Name field – type MW100 (or whatever is desired for the connection name). Comm Format can be left at Data – DINT and IP Address should be set to the IP address of the MW100. Connection Parameters are where the inputs and outputs are defined. In the Assembly instances table below there are Instance IDs that correspond to channels in the MW100. All Instance IDs of Kind Producer can be assigned to Input (e.g. Instance ID 130 would point the Input at Communication Channel C001). All Instance IDs of Kind Consumer can be assigned to Output (e.g. Instance ID 131 would point the Output at Communication Channel C101). Size is how many channels are available per Instance ID. In the Assembly instances table, all Instance IDs can have Size 100 except for MW100 measurement channels 001-060 that have a maximum size of 60. A smaller size can be used if less channels are needed (e.g. below only 10 channels per Instance ID are configured). Configuration can be set to Instance 190 with Size 0. In the case that no Outputs are to be used, Output Instance can be set to 191, Size 0 for heartbeat purposes (allows MW100 to stop broadcasting for data if heartbeat goes away).



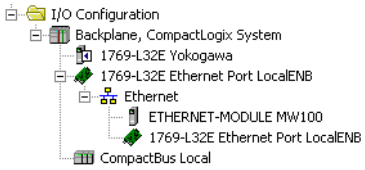
Configuring an MW100 as a Generic Ethernet Module

Assembly instances for I/O messaging

Channels in Assembly Object

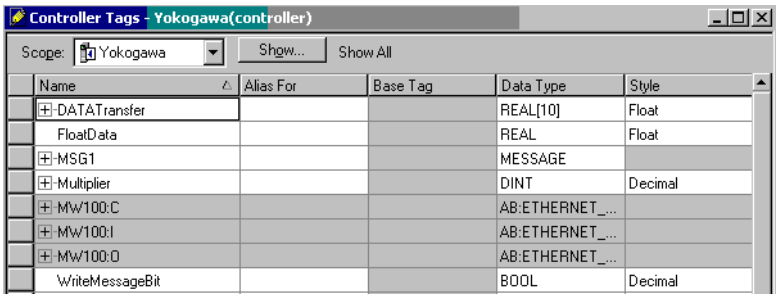
Ch.	Kind	Instance ID	Size	Type
001-060	Producer	110	4 x 60	dint
A001 - A100	Producer	120	4 x 100	dint
A101 - A200		121	4 x 100	
A201 - A300		122	4 x 100	
C001 - C100	Producer/Consumer	130	4 x 100	dint
C101 - C200		131	4 x 100	
C201 - C300		132	4 x 100	
	Configuration	190	0	
	Consumer	191	0	

Now that the MW100 is added, it should appear in RSLogix 5000 as a connection in the Controller Organizer.



I/O Configuration within Controller Organizer tree – ETHERNET-MODULE MW100

Browsing Controller Tags will now show MW100:I and MW100:O as tags that can be used within controller logic. Clicking on the + sign will expand the selection to show all the points up to the Size specified when defining the input and outputs on the module (e.g. Size 10 = 10 Channels/Tags).

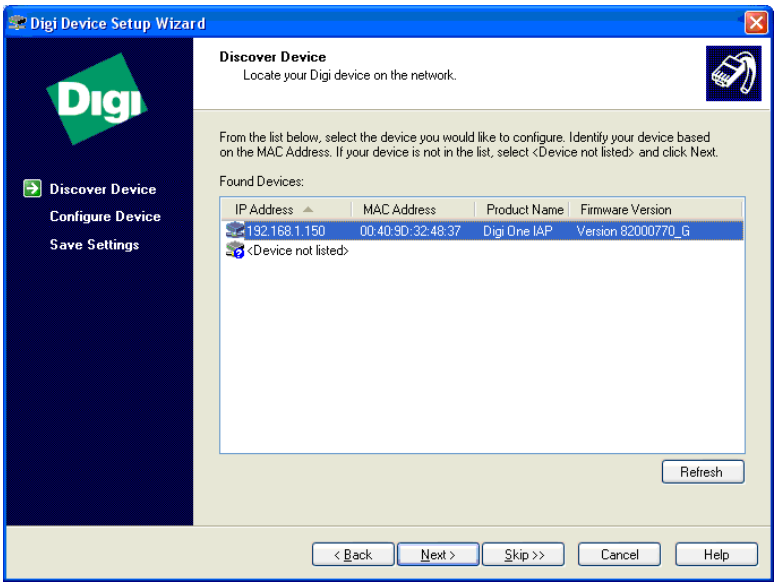


MW100 Channels now available as tags for controller logic

A-2. Detailed Configuration of DigiOne IAP Serial Gateway

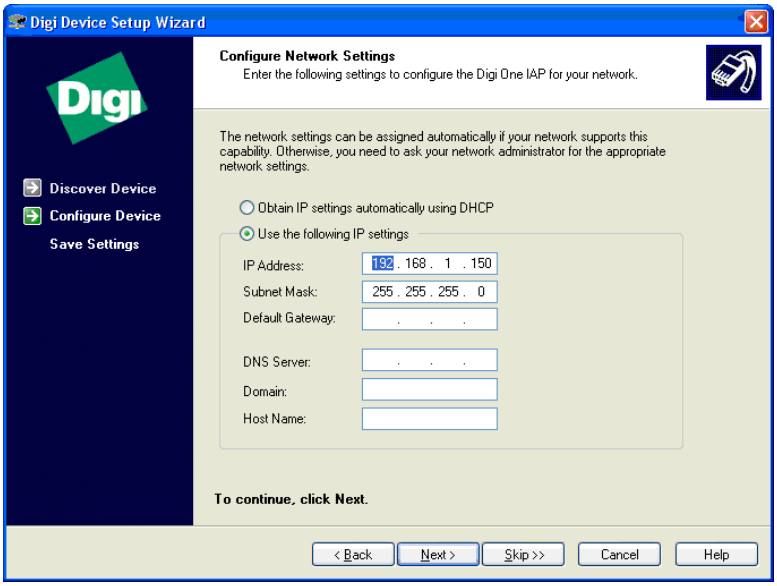
For PLCs that support serial communications via DF1 protocol, there are various gateways that can be used to intercept serial communications and translate to EIP. One such device is the DigiOne IAP. The IAP has two serial ports and a single Ethernet port. Port 2 on the IAP is a 9 pin d-sub connector that can be connected to the 9 pin d-sub connector on PLC CPUs like the SLC 504 (and others). The IAP comes with a serial cable that easily connects the IAP to the SLC 504. A standard Ethernet patch cable can connect the IAP to an Ethernet network. This example will show how to configure an IAP bridge communications between RSLinx/ RSLogix and an MW100 with a SLC 504.

Connect the IAP to the serial port on the front of the SLC 504 via the included serial cable. Connect an appropriate DC power supply to the terminals of the IAP. Power the IAP up. Connect an Ethernet patch cable to the IAP and attach it to the network. The IAP comes with a CD that includes a setup utility. When the CD boots up (assuming auto-run is enabled, follow the instructions on the first page then click next and the setup utility will scan the network for IAPs. For a new IAP it will not have an IP address, but it should be easy to identify as the Product Name will show Digi One IAP. Select the IAP and click Next.



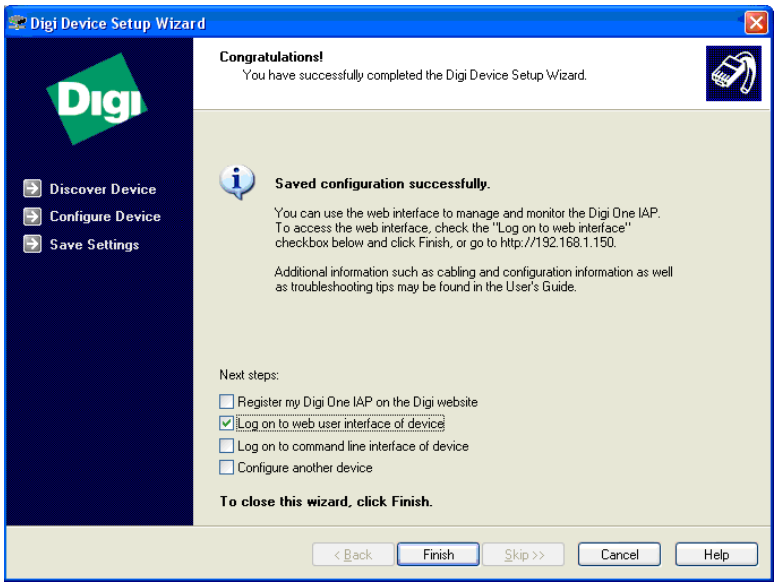
Auto discovery of IAP on network with Digi One Setup Utility

Enter the IP address and subnet mask that is desired.



Configuring IP Address of Digi One IAP

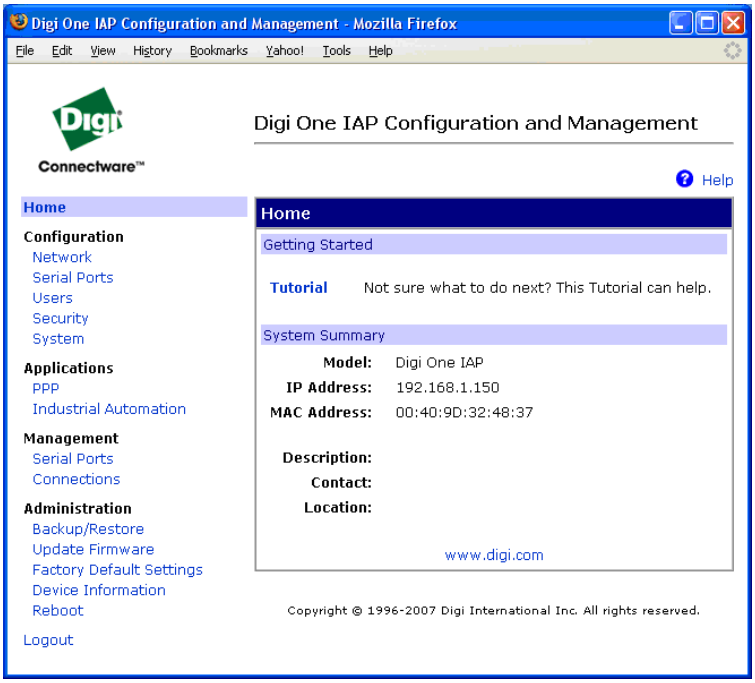
After configuring the network settings choose Skip and Next on the following two screens and the configuration should be saved to the IAP. The web interface has a wizard that can be used to configure the IAP for industrial networks. On the final screen of the setup utility select Log On to the web user interface of device and click Finish.



The default web browser should pop up with a prompt for a user name and password (if it does not automatically launch the web browser, then open a web browser and browse the IP address of the IAP). The default User Name for the web interface is root and the default Password is dbps.

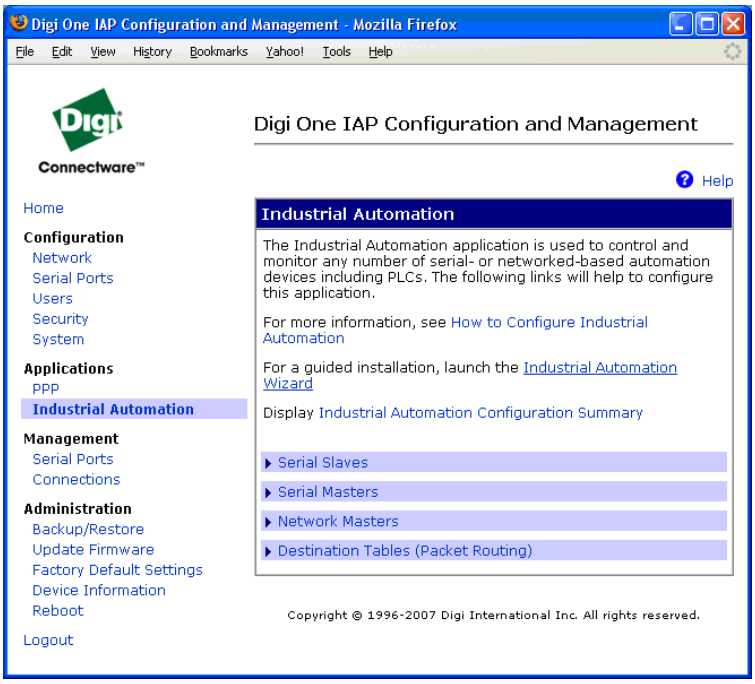
IAP Security Screen

After logging, in the main page of the Digi One IAP web configuration should appear. Select Industrial Automation under Applications to continue.



Home page of IAP configuration

Select Industrial Automation Wizard link in the center of the screen to configure the IAP for an industrial network.



Industrial Automation page

Enter a table name for this configuration and then click Next.

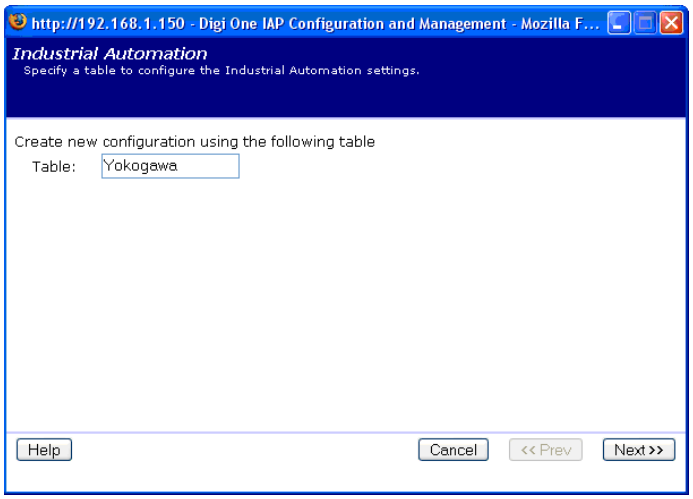
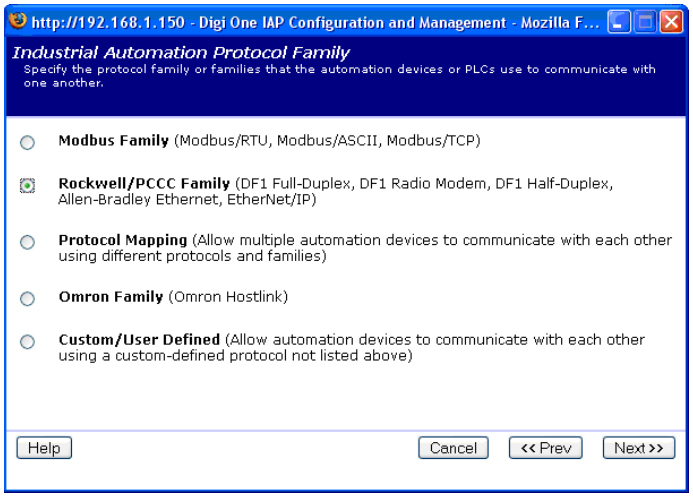


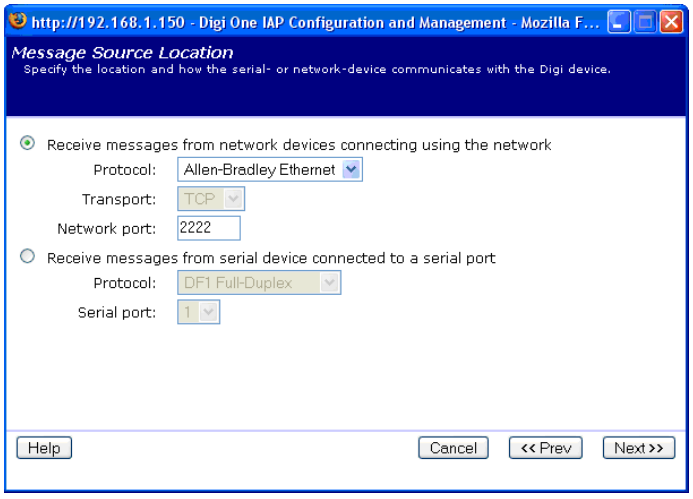
Table Name for Industrial Automation Setup

Select Rockwell/PCCC family and then click Next twice.

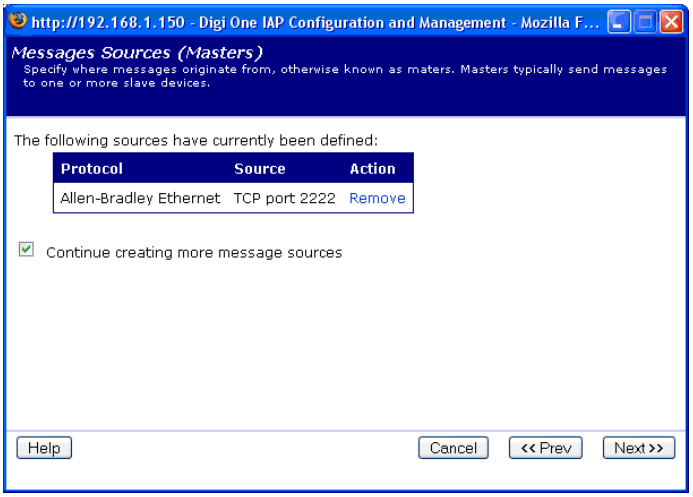


Choosing Industrial Protocol

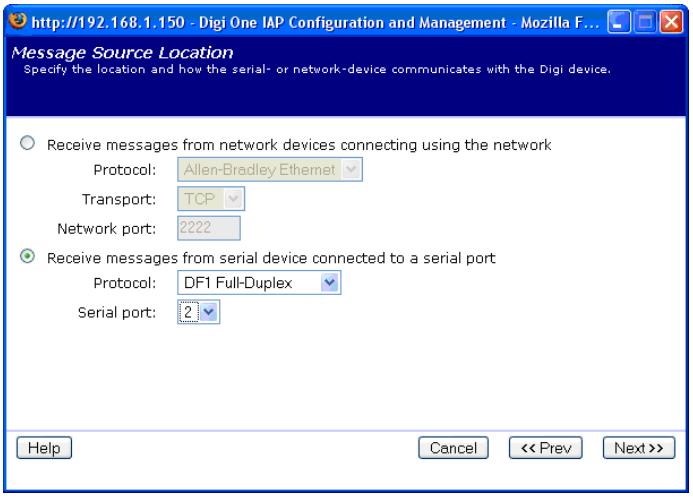
The first source that will be setup is the interface for RSLinx so choose Allen-Bradley Ethernet and click Next.



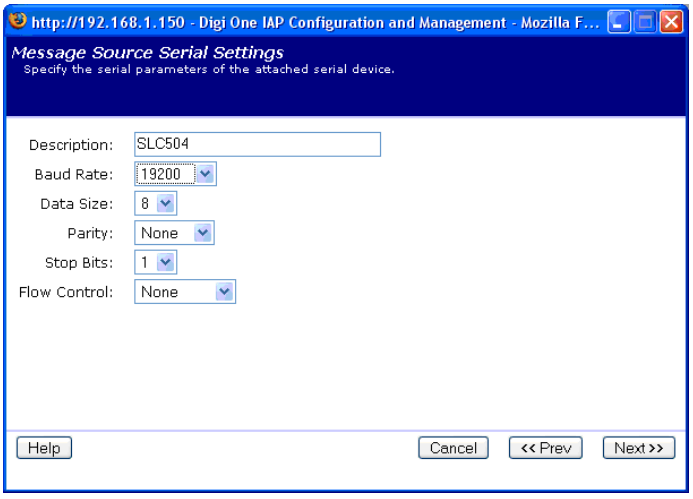
Click Next until the following screen shows up and then check the Continue creating more message sources box and then click Next.



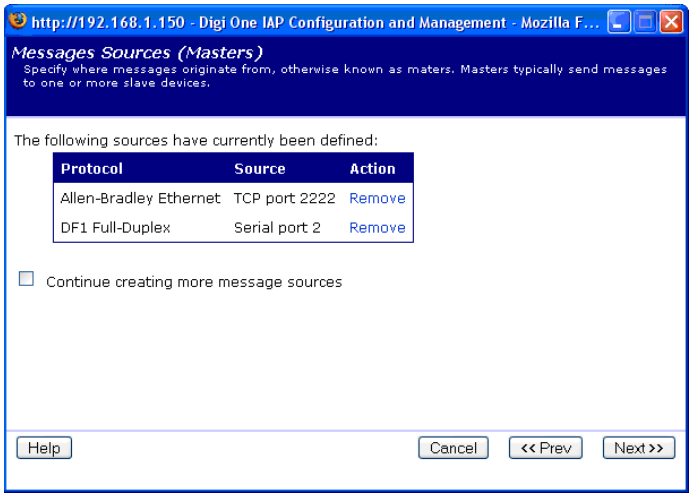
Select Receive messages from serial device connected to a serial port and choose DF1 Full-Duplex for the Protocol and 2 for the Serial port, click Next to continue.



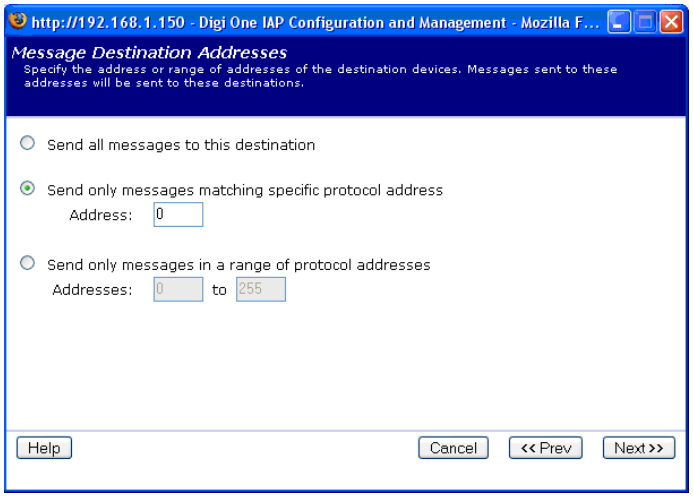
Give the source a Description and configure the serial options to match the configuration of the PLC, then click Next.



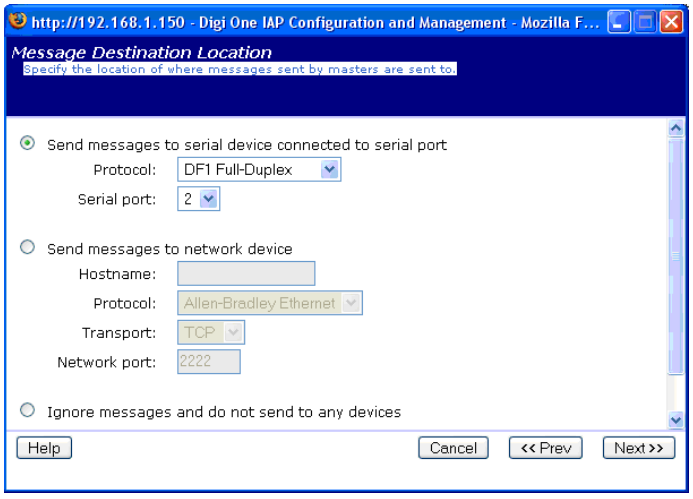
Click Next until the following screen shows up and then uncheck the Continue creating more message sources box and click Next.



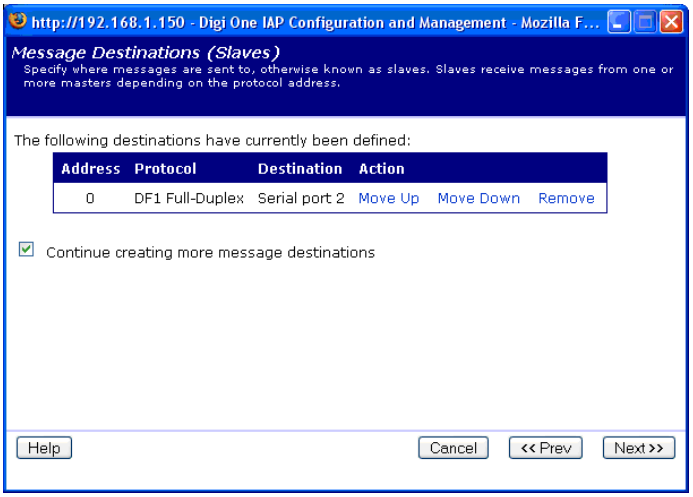
When the IAP receives communications it needs to know where to route the information. To send data to the SLC504 set protocol address to 0.



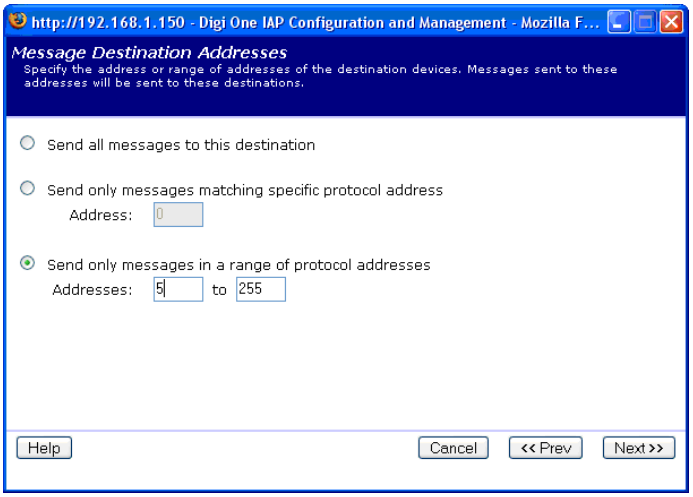
Address 0 communications need to be retransmitted over the serial port connected to the SLC. Set Protocol to DF1 Full-Duplex and Serial Port to 2.



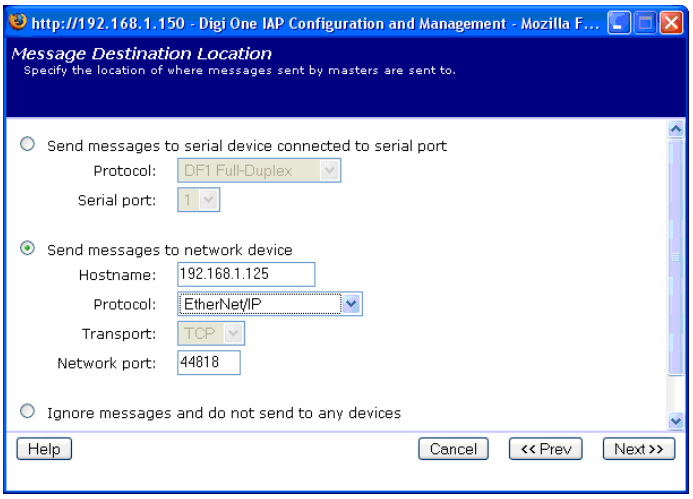
Click Next to accept defaults until returned to this screen and then check Continue creating more message destinations and Next.



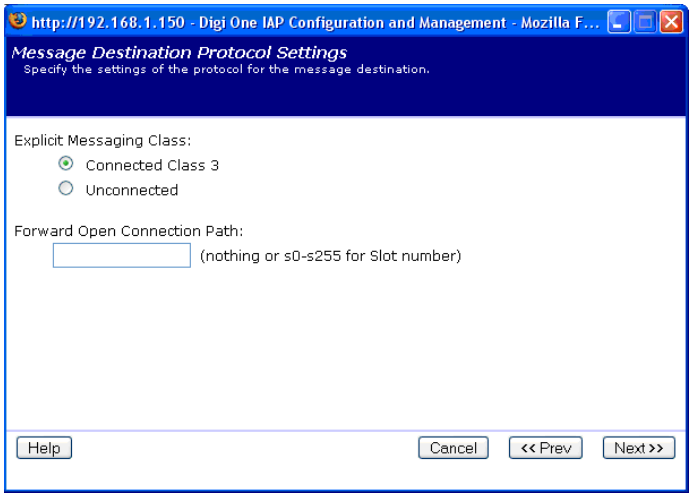
Now it is time to route communications to the MW100. In this case, all write and read commands issued from the SLC504 in this example are sent to Node 5 (it could be set to read or write to any other address). If the SLC is going to write to devices with different addresses, then confine the address to just the address of the specific MW100. In this example the IAP will route any message with an address of 5 to 255 to the MW100. Click Next to continue.



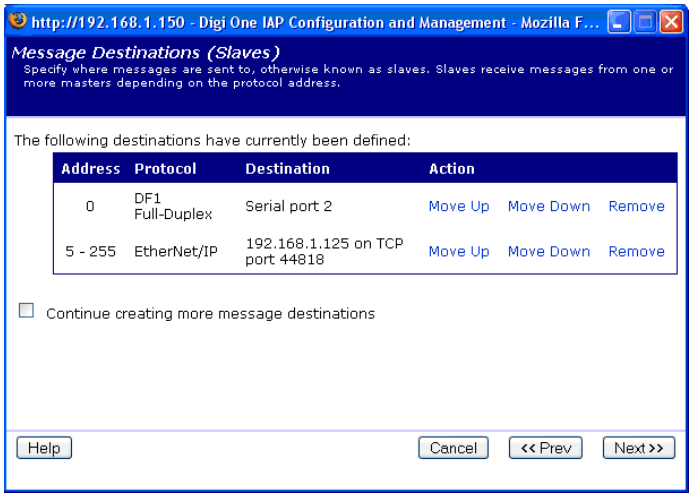
The next step is to tell the IAP where the MW100 is located and how to talk to it. Select Send messages to network device at Hostname – IP Address of MW100. Select EtherNet/IP for the Protocol and then Next.



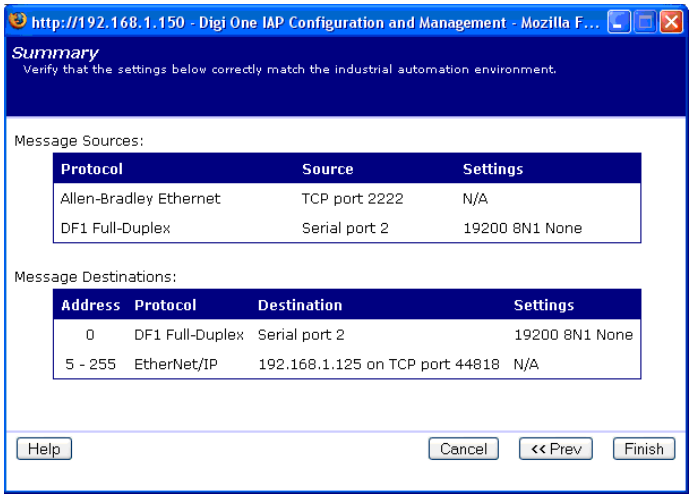
When the Message Destination Protocol Settings screen shows up, ensure that Forward Open Connection Path: is left blank.



Click Next until the following screen shows up and then uncheck Continue creating more message destinations. Click Next to continue.



If all the settings match on the Summary page, click Finish to save the configuration in the IAP.



The IAP will then ask to be rebooted so that all the settings can take effect. Once the IAP is rebooted, the Digi One IAP and attached SLC 504 can now be added as an Ethernet Device (add driver) in RSLinx (use the IP address of the IAP as the IP address for the Ethernet Device).