



**NOTE** Some protocols on the Digi One IAP product family that are described in this document are being moved to legacy status because the protocols described are no longer actively supported. The products themselves continue to be active products.



# **Polling Rockwell PLC and ControlLogix as Modbus slaves with Digi One IAP**

---

March 2020

90000647

## Contents

1	Introduction .....	5
1.1	Sample application.....	5
1.2	Rockwell “PCCC” commands.....	6
1.3	Modbus commands to SLC5 commands.....	6
2	Rockwell PLC notes .....	10
2.1	Understanding file types.....	10
2.2	Overriding default file types.....	11
2.3	ControlLogix notes .....	12
2.4	MicroLogix notes.....	12
2.5	SLC5/0x notes .....	13
2.6	PLC5E notes.....	13
3	Setting up the Digi One IAP .....	13
3.1	Overview.....	13
4	By web wizard, release “E” .....	14
4.1	Select the Industrial Automation Wizard .....	14
4.2	Create a new group and reset all IA settings.....	14
4.3	Limit this group to any protocol that supports “Mapping” .....	15
4.4	Define serial Modbus/RTU master or message source.....	15
4.5	Define AB/Ethernet as master or message source .....	16
4.6	Define DF1 PLC as master or message source.....	17
4.7	Define Modbus/TCP as master or message source.....	17
4.8	Master summary .....	18
4.9	Define master priority .....	18
4.10	Define first message destination—our serial port 2 to MicroLogix .....	19
4.11	Set the protocol and port number .....	19
4.12	Define other destinations .....	20
4.13	Define second message destination—Modbus/TCP PLC .....	20
4.14	Set the Modbus/TCP slave details .....	20
4.15	Define third message destination—AB/PLC5E .....	21

- 4.16 Define fourth message destination – ControlLogix slot 0 .....21
- 4.17 Finished–review the settings .....22
- 4.18 Reboot the Digi One IAP .....23
- 5 Setup by telnet release “E” .....23
- 6 Setting up the ControlLogix for PCCC messaging .....25
  - 6.1 Enable “map PLC/SLC messages ...” .....25
  - 6.2 Usage .....26
- 7 Testing and troubleshooting .....26
  - 7.1 Testing by simulator .....26
  - 7.2 Can I use PLC5 or CIF style commands? .....26
  - 7.3 Why is my MSG block returning a bad status? .....27
  - 7.4 Given Rockwell notation, determine your Modbus .....27
  - 7.5 System documentation .....27

# 1 Introduction

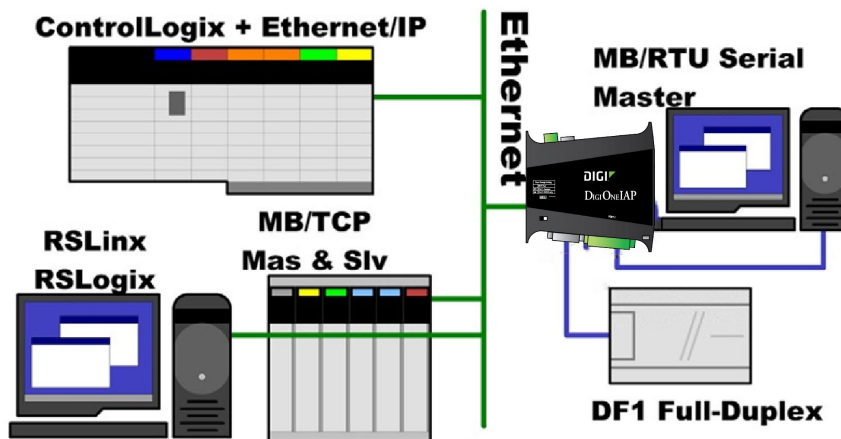
## Abstract

This document describes how to use Modbus polls to read and write information from a ControlLogix PLC. The Digi One IAP makes the ControlLogix appear as a Modbus/TCP slave. The same concepts allow Modbus masters to poll other Allen-Bradley PLC such as MicroLogix, SLC5, and PLC5E.

## 1.1 Sample application

This application note uses a sample design with three masters and three slaves. The table below describes which masters can poll which slaves.

This master	Can poll
Modbus/RTU serial master	ControlLogix using Ethernet/IP
Modbus/RTU serial master	DF1 MicroLogix using DF1 Full-D
Modbus/TCP master	ControlLogix, MicroLogix, or PLC5E
Any Rockwell PLC	Any other Rockwell PLC using normal Rockwell protocols



Here is a summary of the features.

The Digi One IAP allows access to the ControlLogix *concurrently* by Modbus protocols and normal Rockwell protocols. It does not interfere with the normal Ethernet/IP functions of the ControlLogix, RSLinx, or RSLogix 5000.

- Other vendors make Modbus/TCP cards for ControlLogix. However, the cards are expensive and require a second Ethernet cable, hub/switch port, and IP address. The Digi One IAP approach allows the existing ControlLogix Ethernet port to (in effect) concurrently handle Rockwell and Modbus data.

- The Digi One IAP allows the three Rockwell protocols—DF1, AB/Ethernet (CSP), and Ethernet/IP—to bridge with PCCC messaging concurrently to the Modbus access to the Rockwell device.

## 1.2 Rockwell “PCCC” commands

Rockwell refers to the basic commands shared by DF1 and AB/Ethernet (CSPv4) as PCCC. PCCC commands are also encapsulated within Ethernet/IP to integrate ControlLogix PLC to legacy devices.

### 1.2.1 SLC 500 Protected Typed Read/Write (default setting)

You select forming SLC messages by setting the destination in the configuration to a **target** of **slc5**. This setting is the default and it applies to destinations of DF1, AB/Ethernet (CSP), and Ethernet/IP. Details of how Modbus commands convert to Rockwell commands are provided in the next section.

### 1.2.2 PLC2 / CIF style unprotected read/write

You select forming PLC2/CIF messages by setting the destination in the configuration to a **target** of **cif**. This setting applies to destinations of DF1, AB/Ethernet (CSP), and Ethernet/IP. Use this form only with older PLCs that do not support the newer SLC5 Protected Typed Read/Write commands.

Modbus reads to area 3x- or 4x-bridge directly to CIF Unprotected Reads. A Modbus read of 120 registers starting at 4x00001 creates a CIF read of 120 words at offset 0, 4x00101 maps to CIF offset 100, and so on. This off-by-one behavior occurs because Modbus register 4x00001 is actually sent on the wire as address 0, not 1.

Modbus writes to area 4x bridge directly to CIF Unprotected Writes.

**Note** A CIF read is limited to 122 words, which is less than the maximum Modbus read of 125 holding registers. CIF reads to MicroLogix 1000 to access the N7 table, while most other MicroLogix, SLC5, and PLC5 default to N9 unless you change the PLC settings. You are limited to the maximum size of the CIF (or 485CIF or PLC2) file, so SLC5 and MicroLogix are limited to Modbus offsets 4x00001 to 4x00256.

On the ControlLogix, you need to enable PCCC and map PLC2-style messages to a data tag.

## 1.3 Modbus commands to SLC5 commands

### 1.3.1 Function 1 — read multiple coils (Rd/Wr bits)

When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 1 requests to Rockwell Protected Typed Read with three address fields (command 0x0F, sub-function 0xA2).

Although SLC500 commands are word oriented, the Digi One IAP enables reading Modbus style bit ranges without alignment. The Digi One IAP must read the data required, then shift and mask bits to form the expected Modbus response. Therefore, in order to read the four Modbus bits 0x00015 to 0x00018, the Digi One IAP reads 32 bits from the Rockwell device. The four desired bits are B3:0/14, B3:0/15, B3:1/0 and B3:1/1 and thus both elements B3:0 and B3:1 are required to create the desired response.

SLC Protected Typed Reads are limited to 118 words or 1,888 bits, which is less than the maximum Modbus read of 2,000 bits. You may be limited to reading less than the full 1,888 bits at times because of the need to read excess data to isolate the desired bits.

Note that the Digi One IAP assumes all files 9 and higher are type binary unless you explicitly override the type in the configuration. See [Overriding default file types](#) to enable bit access to non-binary files.

Modbus registers		Rockwell files		Data types
0x00001	0x04000	B3:0/0	B3:249/15	Bits, accessed freely in continuous blocks
0x04001	0x08000	B9:0/0	B9:249/15	Bits, accessed freely in continuous blocks
0x08001	0x65536	Bit files B10:0/0 to B24:95/15 per formula		

Download the [Digi Protocol Bridging Calculator](#) (part number 90000653), and use this spreadsheet for easy calculations.

Modbus offsets 0x04001-0x65536 bridge indirectly by formula using integer division and remainder (mod) functions:

- File number = ((offset – 1) divided by 4000) + 8
- Bit offset within File = remainder of (offset – 1) by 4000
- Element = bit offset divided by 16
- Bit number = remainder of bit offset by 16

Example, Reading Modbus 0x04010 returns B9:0/9:

- File:  $((17534-1) / 4000) + 8 = 4 + 8 = 12$
- Bit offset in B-file:  $((17534-1) \text{ mod } 4000) = 1533$
- Element:  $1533 / 16 = 95$
- Bit:  $(1533 \text{ mod } 16) = \text{bit } 13$

Example: Reading Modbus 0x17534 returns B12:95/13:

- File:  $((17534-1) / 4000) + 8 = 4 + 8 = 12$
- Bit offset in B-file:  $((17534-1) \text{ mod } 4000) = 1533$
- Element:  $1533 / 16 = 95$
- Bit:  $(1533 \text{ mod } 16) = \text{bit } 13$

### 1.3.2 Function 2—read multiple input status (read only bits)

When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 2 requests to Protected Typed Read with three address fields (command 0x0F, sub-function 0xA2).

Function 2 requests are treated exactly like function 1 requests.

### 1.3.3 Function 3—read multiple holding registers (Rd/Wr Words)

This is perhaps the most common Modbus command. It allows you to access most of the Rockwell PLC's data files. When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 3 Requests to Protected Typed Read with three address fields (command 0x0F, sub-function 0xA2).

Since the first files from 0 to 8 are normally of fixed type, the Modbus commands will auto-bridge to the following Rockwell standard file types. The complex (structured) data files such as O, I, S, T, C, and R are read-only, since Modbus access to the various bit, byte, and word fields could be unpredictable. Users should refer to Rockwell documentation to interpret the data returned.

**Note** The Digi One IAP will assume all files numbered 9 and higher are type integer unless you explicitly override the type in the configuration. See [Overriding default file types](#) to enable word access to non-integer files.

Modbus Registers		Rockwell Files		Data Types
4x00001	4x00250	N7:0	N7:249	Integer
4x00251	4x00500	I1:0	I1:249	Input Logical by Slot (read only)
4x00501	4x00750	S2:0	S2:249	Status (read only)
4x00751	4x01000	B3:0	B3:249	Binary read as words
4x01001	4x01250	T4:0	T4:83	Timer (3 words each; read only)
4x01251	4x01500	C5:0	C5:83	Counter (3 words each; read only)
4x01501	4x01750	R6:0	R6:83	Control (3 words each; read only)
4x02001	4x02250	F8:0	F8:119	Floats (Modicon Style – Low Word first)
4x02251	4x02500	N9:0	N9:249	Integer
4x02501	4x64000	Integer Files N10:0 to N253:249 per formula		

Use the [Digi Protocol Bridging Calculator](#) (part number 90000653) for easy calculations.

Modbus offsets 0x00251-0x64000 bridge indirectly through a formula using integer division and remainder (mod) functions:

- File number = (offset – 1) divided by 250
- Element within File = remainder of (offset – 1) by 250

Example: Reading Modbus 4x04010 returns word N16:9:

- File:  $(4010-1) / 250 = 16$
- Element:  $(4010-1) \text{ mod } 250 = 9$

Example: Reading Modbus 4x17534 returns word N70:33:

- File:  $(17534-1) / 250 = 70$
- Element:  $(17534-1) \text{ mod } 250 = 33$



You also can map the opposite way with the following formula:

- $\text{Modbus} = (\text{file number} * 250) + \text{element number} + 1$

For example, to read N9:10, read Modbus 4x02261 =  $(9 * 250) + 10 + 1$ .

Limitation: *SLC Typed Reads are limited to 118 words, which is less than the maximum Modbus read of 125 holding registers.* Reading more than 118 words in a single request (or trying to read past the end of a Rockwell file) will cause Modbus exception responses to be returned.

### 1.3.4 Function 4 — read multiple input registers (read-only words)

When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 4 requests to Protected Types Read with three address fields (command 0x0F, sub-function 0xA2).

Modbus function 4 accesses the same files as function 3, except the default range of 3x00001 to 3x00250 accesses the O0 or Output Logical by Slot file.

The N7 file can only be accessed as 3x01751 to 3x02000.

### 1.3.5 Function 5 — force single coil (Rd/Wr Bits)

When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 5 requests to Protected Typed Mask Write with three address fields (command 0x0F, sub-function 0xAB).

Modbus function 5 accesses the same files (using the same calculations) as function 1. This means you can write individual bits in files B3, and B9 to B24. Files 9–24 are assumed type binary unless you've overridden the file type. See [Overriding Default File Types](#) to enable bit writes to non-integer files.

Be aware that the Digi One IAP and Rockwell protocols let you set arbitrary bits within non-binary data elements, such as floating point within F-files, which can lead to invalid or wildly varying floating point values.

### 1.3.6 Function 6 — force single holding register (Rd/Wr Words)

When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 6 requests to Protected Typed Write with three address fields (command 0x0F, sub-function 0xAA).

Modbus function 6 accesses the same files (using the same calculations) as function 3; however, no true read-back is being performed. Some users prefer using function 6 over function 16 for writing 1 register because the data value echoed back within the response acts as a double-check beyond the 16-bit CRC. However, the SLC500 Typed Write does not echo any data back. Consequently, the data echoed within a Modbus function 6 response is just a copy from the request.

Be aware that the Digi One IAP and Rockwell protocols let you set arbitrary words within floating point elements of F-files, which can lead to invalid or wildly varying floating point values.

### 1.3.7 Function 15 — force multiple coils (Rd/Wr Bits)

When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 15 requests to Protected Typed Write with three address fields (command 0x0F, sub-function 0xAA).

**Note** You can only write word-aligned blocks of bits in multiples of 16-bits. Rockwell does not have an equivalent command that writes arbitrary strings of bits without word alignment. Your only options are to write full words of bits or individual bits with Modbus function 5.

Modbus function 15 accesses the same files (using the same calculations) as function 1. This means you can write blocks of bits in files B3, and B9 to B24. Files 9–24 are assumed type binary unless you've overridden the file type. See [Overriding Default File Types](#) to enable bit writes to non-integer files.

SLC Typed Writes are limited to 117 words (1,872 bits), which is less than the maximum Modbus bit write.

### 1.3.8 Function 16 — force multiple holding register (Rd/Wr Words)

When **target** is set to the default **slc5**, the Digi One IAP bridges Modbus function 16 requests to Protected Typed Write with three address fields (command 0x0F, sub-function 0xAA).

Modbus function 16 accesses the same files (using the same calculations) as function 3. Files 9–255 are assumed type integer unless you've overridden the file type. See [Overriding Default File Types](#) to enable bit writes to non-integer files.

SLC Typed Writes are limited to 117 words, which is less than the maximum Modbus register write.

## 2 Rockwell PLC notes

Although this application note covers using Modbus to poll ControlLogix, the Digi One IAP also can poll DF1 or AB/Ethernet (CSP) PLC such as SLC5/05 or PLC5E.

For example, you could attach a MicroLogix 1000 to the second serial pass-through port on the Digi One IAP. This allows the serial Modbus master to query the DF1 PLC as if it were Modbus/RTU. In addition, remote RSLinx and OPC servers can still access the DF1 PLC as DF1.

### 2.1 Understanding file types

Modbus has two sizes of data: 1-bit “coils” and 16-bit “registers.” Additionally, it includes a read-write and read-only area for each. This information is implied by the command selected; Modbus function 1 reads the read-write bits while Modbus function 3 reads the read-write words.

In contrast, Rockwell PLC such as SLC5, PLC5, and MicroLogix use common read-write commands that are word oriented, and the file type is defined as part of the command. So the same command (0x0F 0xA2) is used to read binary (“B”), integer (“N”) and floating point (“F”).

- **Special types supported:** The Digi One IAP offers read-only access to the I/O files 0 and 1, Status file 2, Timer file 4, Counter file 5, and Control file 6. These files are treated as read-only since they contain complex data structures that vary between both PLC models and PLC firmware levels.
- **Special types not supported:** The Digi One IAP does not offer access to undocumented file types such as ST (string), MG (message), or PID.
- **Binary (“B-files”):** AB PLC support reading and writing of binary tables by full words only. This contrasts with the Modbus protocol which allows continuous ranges of bits to be read or

written without regard for word alignment. This limits accessing B-files by Modbus bit commands; some combinations work and some do not.

- **Integer (“N-files”)**: Modbus bridges transparently to N-files.
- **Float (“F-files”)**: Modbus accesses F-files in the same manner as it access ranges of 32-bit IEEE floats within Modicon PLC. You must read pairs of consecutive registers and the low-word of the float is returned in the first register. This is not as one would expect for “big-endian” designs, but is how Modicon implemented 32-bit floating points.
- **Long (“L-files”)**: As with F-files, L-files must be read as pairs of consecutive registers and the low-word of the 32-bit integer is returned in the first register. The data type used for L-files is not published within the DF1 specification, so this file type may not work on some AB PLC.

## 2.2 Overriding default file types

How fussy a Rockwell PLC is relates to file type and varies by product line. Some PLC allow a binary file to be accessed as either type binary or integer, while others reject integer access to binary files with a bad status (STS=0x10).

The first 9 Rockwell PLC files are normally of fixed type and the Digi One IAP uses the correct type for any Modbus access. These files are:

- I/O files #0 and #1
- Status file #2
- Binary file #3
- Timer file #4
- Counter file #5
- Control file #6
- Integer file #7
- Floating Point file #8

Note that files 0, 1, 2, 4, 5, and 6 are always treated as read-only by the Digi One IAP. This is because they contain complex data structures that vary between PLC models and firmware levels.

For files 9–255, Modbus bit commands (1, 2, 5, 15) assume a binary file type, while register commands (3, 4, 6, 16) assume an integer file type. Thus, using Modbus bit commands to read file N9 or Modbus register commands to read file B16 may fail on some PLC.

The Digi One IAP has a configuration option to override these assumed defaults. It is not available by web browser, but can be set by telnet or by editing the “Backup / Restore” file (see Digi application note [Saving and restoring your Digi One IAP configuration](#)). It is an ASCII string consisting of comma separated file names in Rockwell notations. In the example below: N9 marks file 9 as integer; L12 marks file 12 as long 32-bit integers; B13, B19 and B136 mark files 13, 19, and 136 as binary; and F70 marks file 70 as float 32-bit floating points.

```
set ia serial=2 mapping="N9,L12,B13,B19,F70,B136"
```

You only need to use this option to override defaults. Thus, if you have 4 “N” tables from N10 to N13 and only plan to use Modbus function 3 to read them, there is no reason to configure them.

## 2.3 ControlLogix notes

To access the ControlLogix with PCCC messaging, you need to enable support for PCCC messaging (see section 5 on how to set it up). The first response from ControlLogix can take up to five seconds, so you need a five-second timeout on all polls.

When you define the ControlLogix as a destination, consider these destination (or route) options:

- **Class:** When the Digi One IAP acts as client (outgoing or master), it supports either connected or unconnected explicit messaging, also known as UCMM and CM Class 3. Because of the chassis design of the ControlLogix, you must select connected messaging or “**cm3**.” Digi does not currently support implicit or I/O messaging.
- **Connection Path:** Because you don’t want to talk to the ControlLogix Ethernet module (the ENBT), you need to use CM with a connection path to explain to the ENBT which slot number the CPU is in. By default, no connection path is included, and the Modbus poll fails. Add a connection such as “s0” for a CPU in slot 0 or “s5” for a CPU in slot 5.
- **Target:** On the Digi One IAP, you can define the ControlLogix as either a **cif** or **slc5** destination. This defines how the Modbus requests are mapped to PCCC commands. **CIF** (or PLC2) commands limit you to a single data tag in the ControlLogix. **SLC5** Typed Read/Writes allow you to define a large number of file-to-tag mappings within the ControlLogix.
- **Service:** (Optional) When you move PCCC messages that are encapsulated within Ethernet/IP, you can select several optional services. The default of 0x4B (**Execute PCCC**) is fine for a ControlLogix. This form, however, does not preserve the SRC/DST information from DF1, which includes the slave address from Modbus/RTU.
- You must select an alternate service if you want to access a remote Digi One IAP or 1761-NET-ENI that either:
  - Does port routing based on these values.
  - Preserves the values to tunnel DF1 using Ethernet/IP service 0x4C using a header that mimics a DH+ station destination/source header.
  - This setting is available only by telnet (command line); the settings are “service=exec\_pccc” or “service=vdhplus.”

## 2.4 MicroLogix notes

A MicroLogix is best accessed as DF1 full-duplex. For destination options, see the information about ControlLogix in the previous section.

If you define the MicroLogix as **target=cif**, you are limited to a single file in the PLC. For MicroLogix 1000 this file is N7, while for 1200 and 1500 this file is N9. Using **target=slc5** allows you to access many files.

Options for connection path, class, and service don't apply to DF1 destinations.

## 2.5 SLC5/0x notes

The native protocol of the SLC5/05 is the older AB/Ethernet protocol. Rockwell refers to this as Client-Server Protocol (CSP). Newer firmware also supports Rockwell's PCCC-encapsulation in Ethernet/IP, referred to as "Multi-Hop" when you configure outgoing MSG blocks. The Digi One IAP supports either AB/Ethernet or Ethernet/IP to access remote SLC5/05. AB/Ethernet is a little more efficient due to its fixed sized headers, but you can select either protocol. When the destination is configured as AB/Ethernet, only the **target** option is valid, and it is best set to **slc5**. For Ethernet/IP both the **target** and **class** options are valid. You can use either ucmm or cm3, but Rockwell internal buffer allocation schemes favor use of **cm3**.

The serial port of the SLC5/03, SLC5/04, and SLC5/5 can be treated like a MicroLogix with DF1 full-duplex. Newer firmware on these three models can do internal bridging from DF1 to their network ports. Using DF1, AB/Ethernet, or Ethernet/IP with the alternative 0x4C service to preserve the SRC/DST bytes allows network masters to "bridge" to DH485, DH+, or Ethernet/IP using a Digi One IAP connected to the serial DF1 port of the PLC.

## 2.6 PLC5E notes

Most of the SLC5 notes also apply to the PLC5E. These are the only major differences:

- The MicroLogix and SLC5 limit each file to 255 elements
- The PLC5 does not have the same limitation

The instructions in this document are based on the assumption that you understand the basics of setting up and accessing your Digi product by Ethernet and TCP/IP. If you need help with these procedures, see the related documentation on Digi's [support site](#).

# 3 Setting up the Digi One IAP

The instructions in this document are based on the assumption that you understand the basics of setting up and accessing your Digi product by Ethernet and TCP/IP. If you need help with these procedures, see the related documentation on Digi's [support site](#).

## 3.1 Overview

This is a summary of the steps for configuring the Digi One IAP.

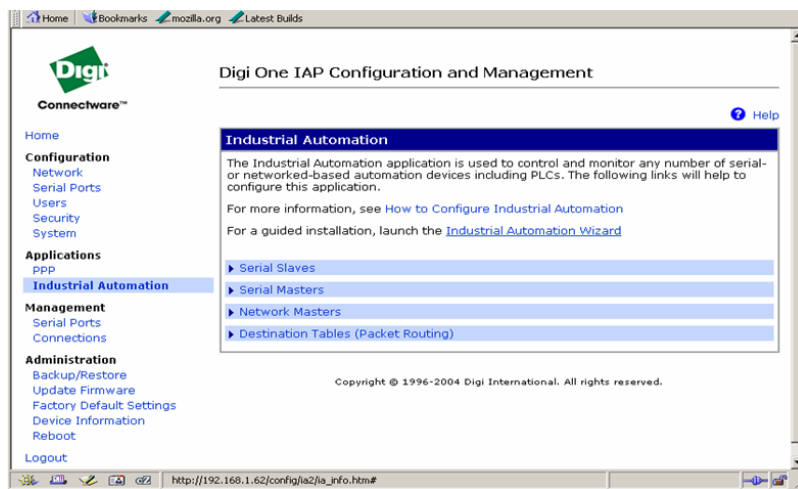
- Configure message sources (incoming masters or peers):
  - Set serial Modbus/RTU master on port #1
  - Set serial AB/DF1 master on port #2 (is peer-to-peer)
  - Set Modbus/TCP masters
  - Set AB/Ethernet masters
- Configure message destination (how messages are answered):
  - Set slave destination #1 as serial port #2 (DF1)

- Set slave destination #2 as ControlLogix at remote IP
- Set slave destination #3 as AB/PLC5E at remote IP
- Set slave destination #4 as Modbus/TCP at remote IP

## 4 By web wizard, release “E”

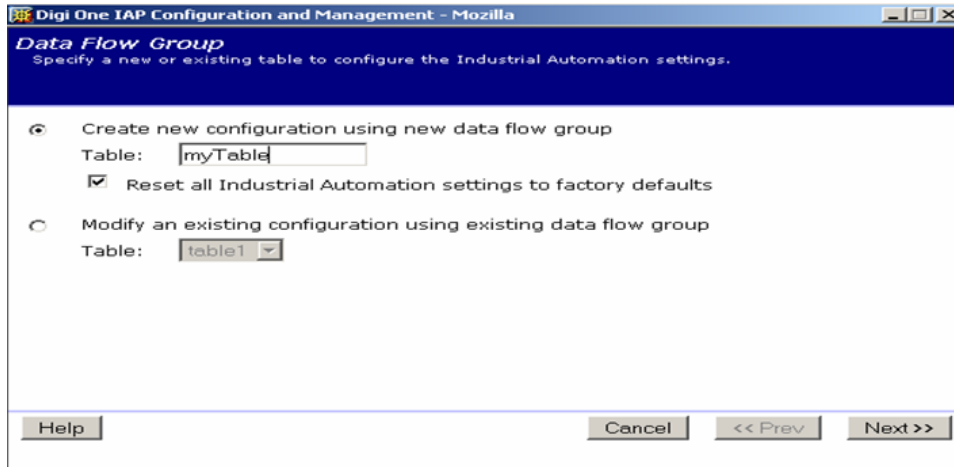
### 4.1 Select the Industrial Automation Wizard

1. In the **Applications** section in the left column, click **Industrial Automation**.
2. Click **Launch the Industrial Automation Wizard**. The wizard helps you define where messages come from and where they go.



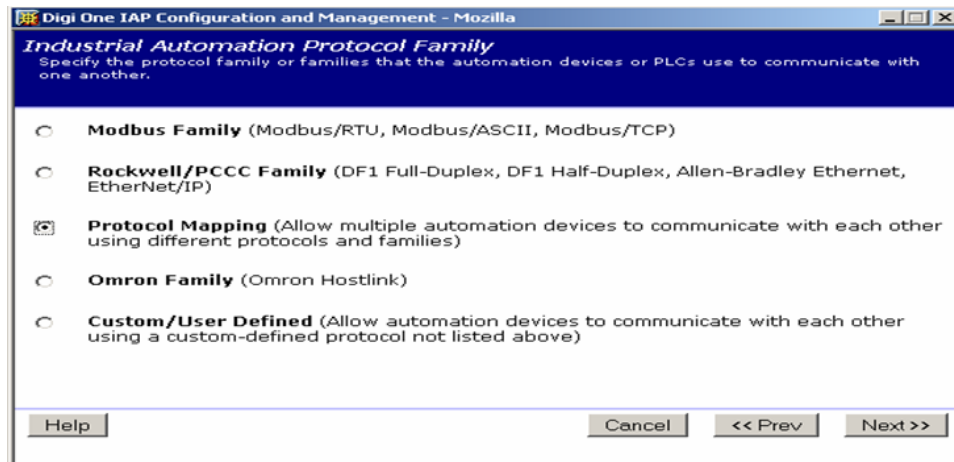
### 4.2 Create a new group and reset all IA settings

1. Click **Create new configuration using new data flow group** to create a new configuration group and clear all existing IA settings.
2. Check **Reset all Industrial Automation settings to factory defaults**.
3. Enter a name in the **Table** input box, such as table1 or myTable.
4. Click **Next**.



### 4.3 Limit this group to any protocol that supports “Mapping”

1. Click **Protocol Mapping** group to map Modbus protocols to Rockwell. This selection allows most protocols to be selected as you configure.
2. Click **Next**.

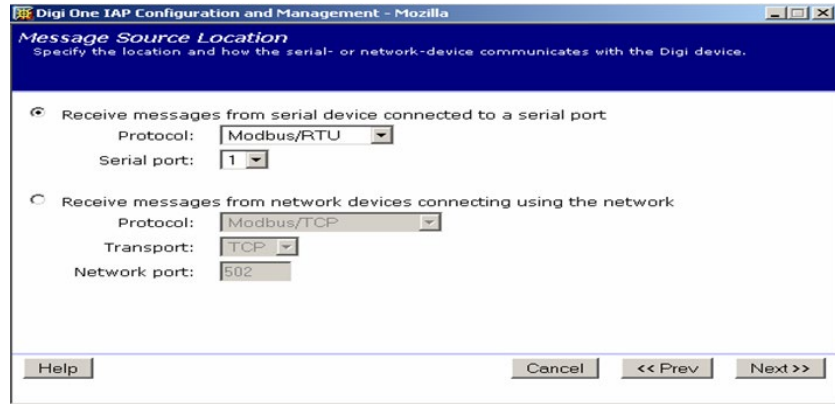


### 4.4 Define serial Modbus/RTU master or message source

You will see some windows about **Message sources** that aren't shown in this document.

1. Click **Next** until you see the window shown next.
2. Click **Receive messages from serial devices connected to a serial port**.
3. From the **Protocol** pulldown menu, select **Modbus/RTU**.
4. Set the **Serial port** to **1**.
5. Click **Next**.

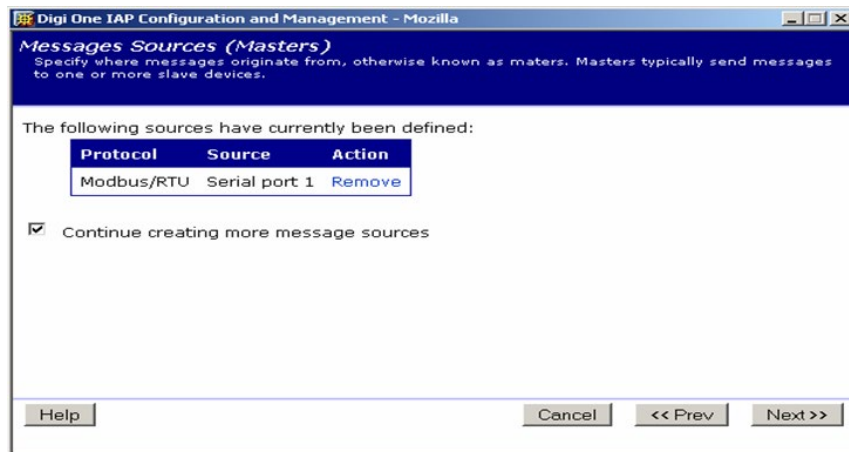
In the next few windows, you'll set port characteristics, such as baud rate and parity.



#### 4.5 Define AB/Ethernet as master or message source

This allows RSLinx to access the MicroLogix on port #2.

1. Click Continue creating more message sources.
2. Click Next.



3. Click Receive messages from network devices connecting using the network.
4. From the Protocol pulldown menu, select Modbus/TCP. The Network port automatically sets to 2222.
5. Click Next and accept the defaults.

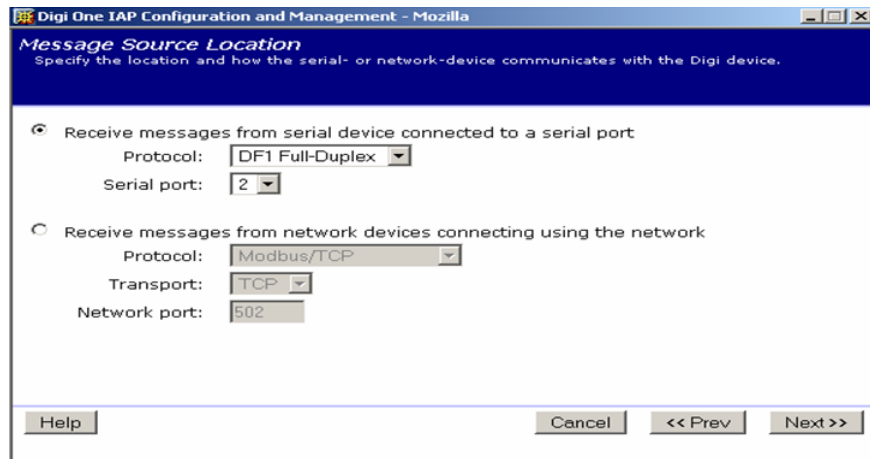




#### 4.6 Define DF1 PLC as master or message source

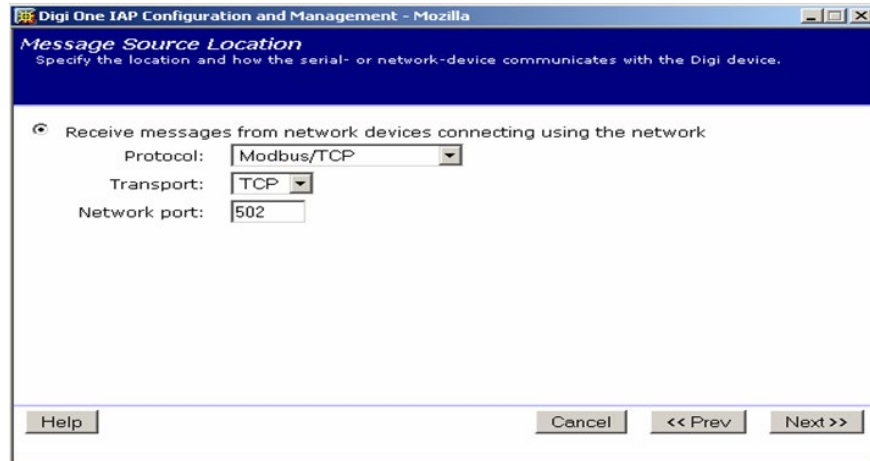
If you want, the MicroLogix on serial port 2 can use MSG blocks to act as master.

1. Define a DF1 full-duplex master on serial port 2.
2. Click **Next** and select the other settings as needed.



#### 4.7 Define Modbus/TCP as master or message source

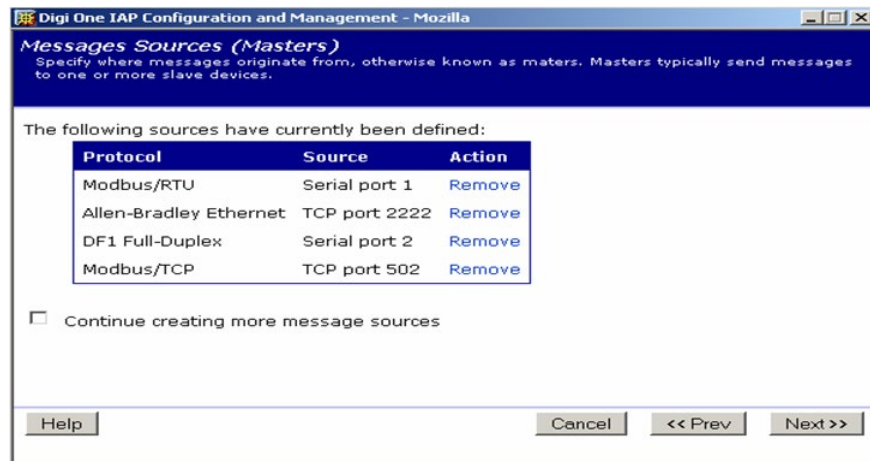
1. To enable remote Modbus/TCP masters, use the Digi One IAP to access Rockwell PLC as Modbus/TCP slaves.
2. Click **Next**.
3. Set the other settings as needed.



## 4.8 Master summary

The four masters or message sources are defined.

1. To stop adding masters, uncheck Continue creating more message sources.
2. Click Next.

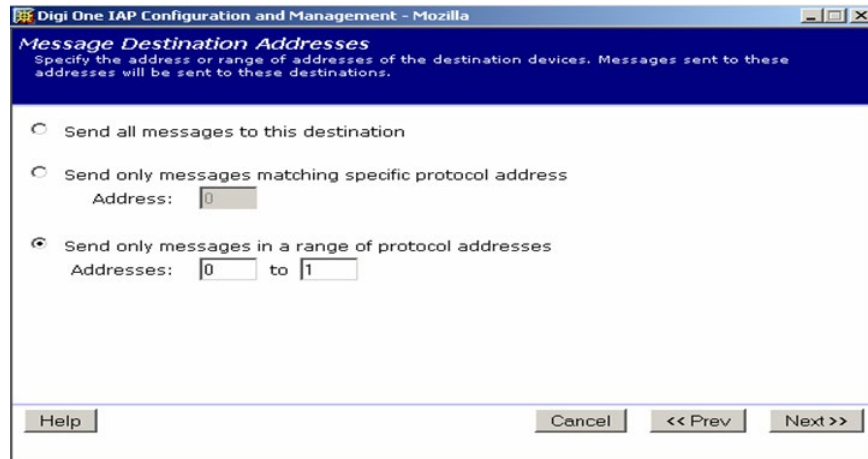


## 4.9 Define master priority

1. You can assign a higher priority to any one master. To leave all masters at the same priority, accept the default.  
When masters have the same priority, each incoming socket is given fair, round-robin access to the serial PLC on port 2. Because over 60 remote network masters can connect, the Modbus/RTU master could be limited to less than 2% of the serial bandwidth to access the MicroLogix. If this limitation becomes a problem, define the serial Modbus/RTU master as highest priority.
2. Click **Next**.

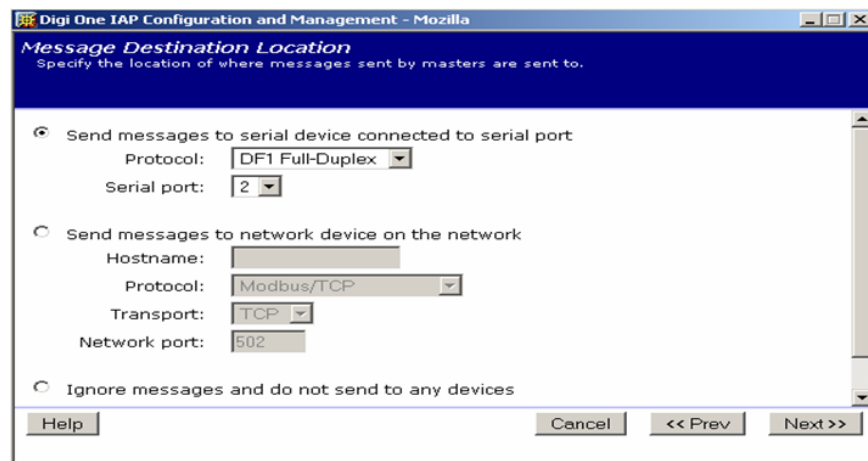
## 4.10 Define first message destination—our serial port 2 to MicroLogix

1. Click **Next >>** until you see the window shown next to pass the window about **Message Destinations**.
2. Click **Send only messages in range of protocol addresses**.
3. In the **Addresses** input boxes, enter 0 to 1. This setting allows the Modbus/RTU serial master to query the MicroLogix as Modbus/RTU slave #1. RSLinx also can access the MicroLogix as DST node 0 or 1.
4. Click **Next**.



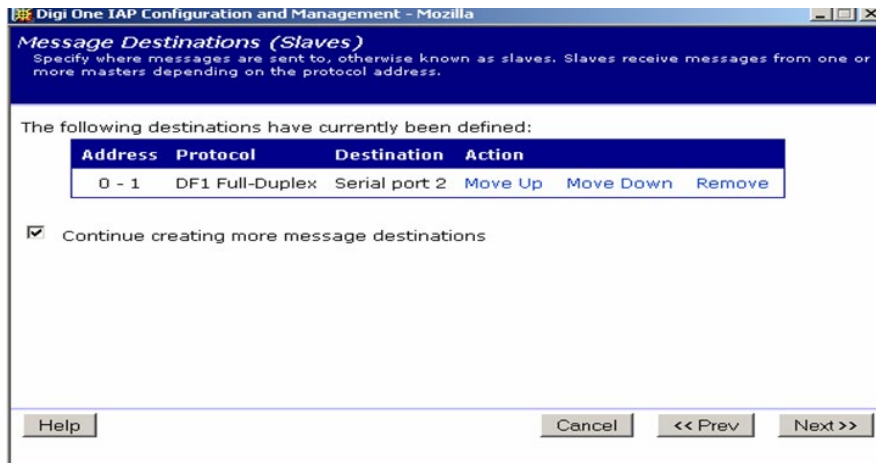
## 4.11 Set the protocol and port number

1. Set up DF1 full-duplex on port 2.  
You'll move through a few windows here; set up the port and protocol as necessary. Generally, the defaults are fine as is.
2. Click **Next**.



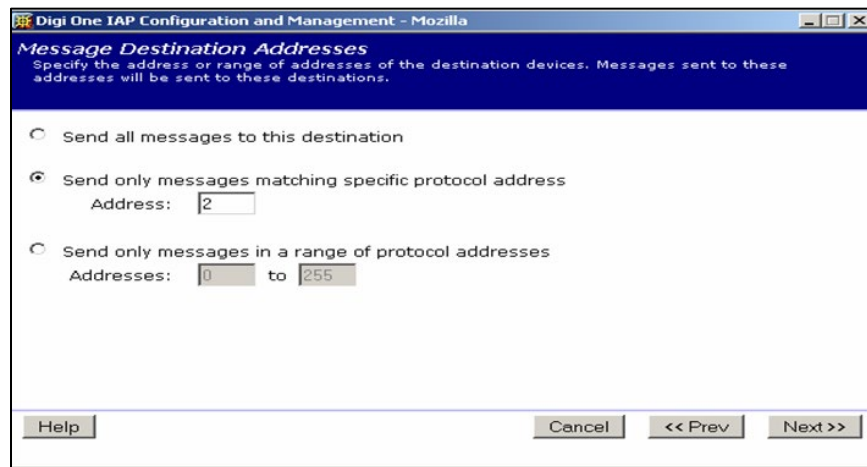
## 4.12 Define other destinations

1. Check **Continue creating more message destinations** when you get to the summary window shown next.
2. Click **Next**.



## 4.13 Define second message destination—Modbus/TCP PLC

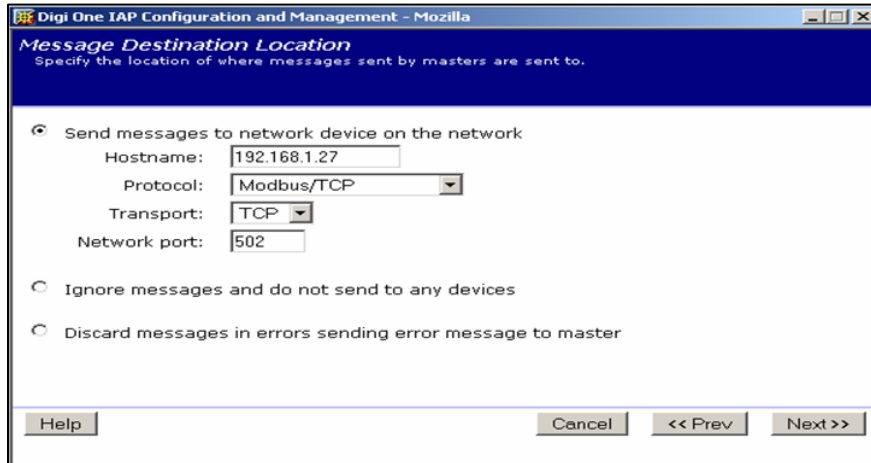
1. Send the single protocol address 2 to the Modbus/TCP PLC at IP 192.168.1.27.
2. Click **Next**.



## 4.14 Set the Modbus/TCP slave details

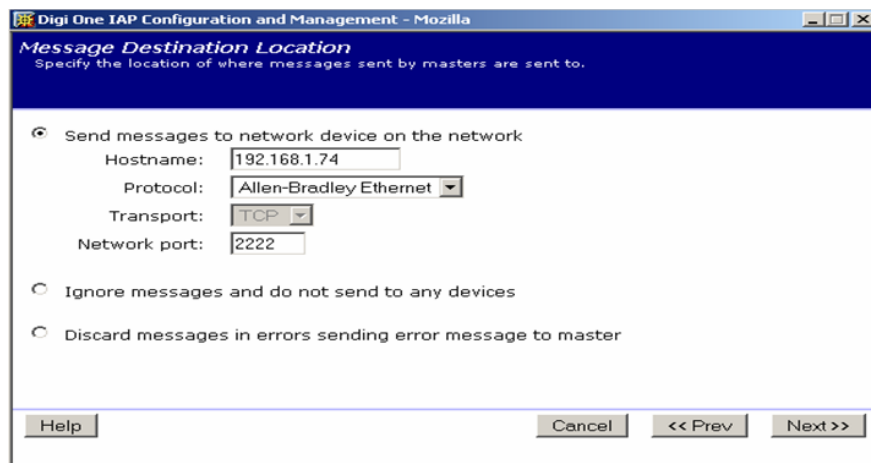
Select to use Modbus/TCP to either an IP address or DNS name.

You'll move through a few more windows; set them up as necessary. The defaults are fine.



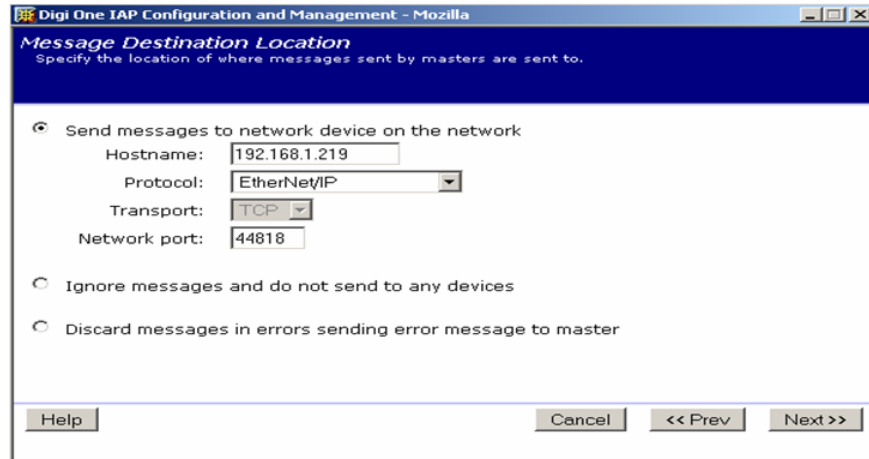
#### 4.15 Define third message destination–AB/PLC5E

1. When you return to the summary, continue and create another destination for protocol address 3 to the AB/PLC5E as slave.
2. Select to use **SLC5 commands**.

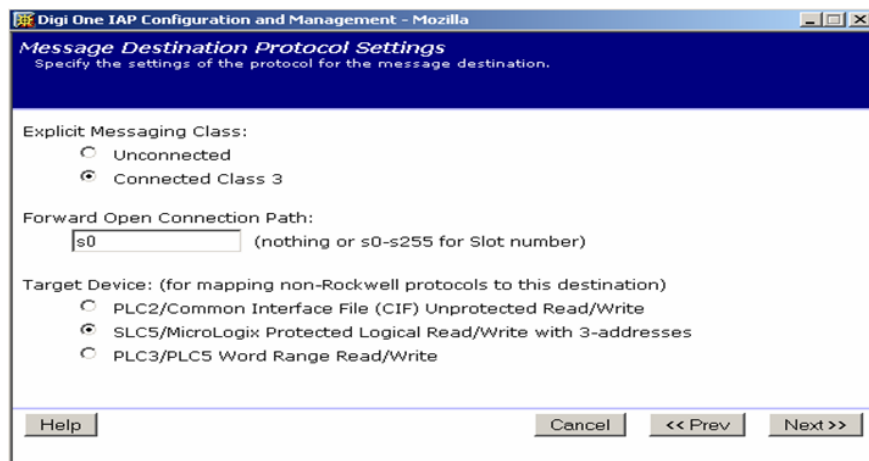


#### 4.16 Define fourth message destination – ControlLogix slot 0

When you return to the summary, continue and create another destination for protocol address 4 to the ControlLogix as slave.



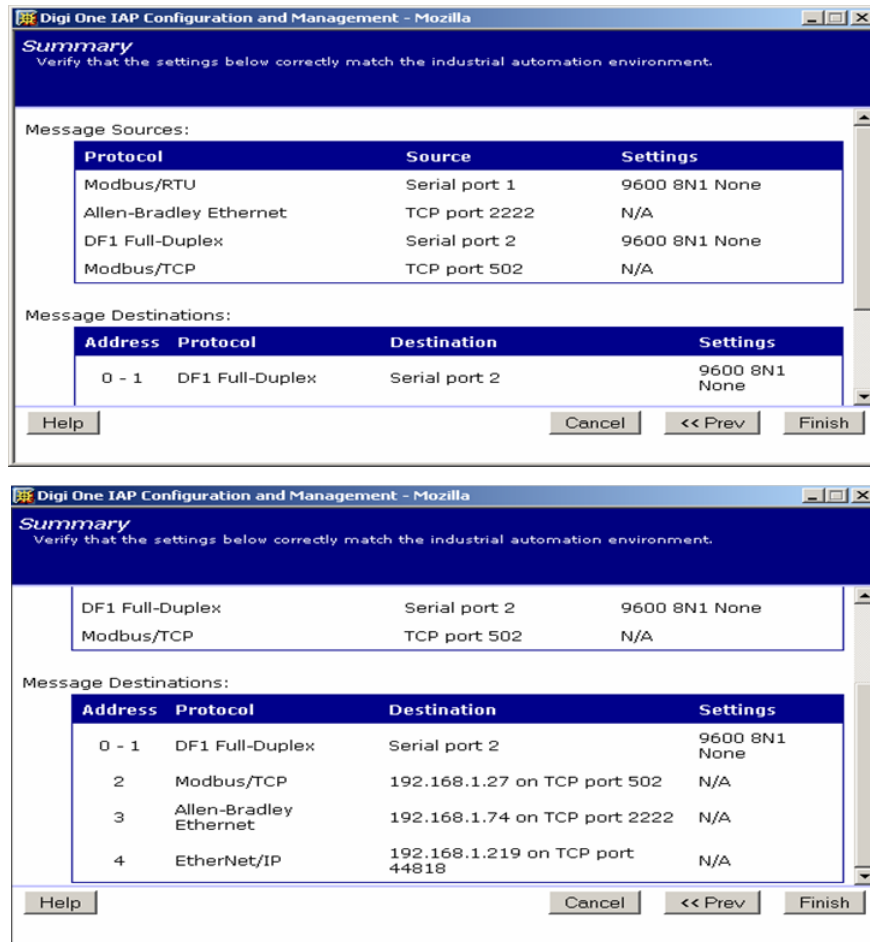
1. Under Explicit Messaging Class, click Connected Class 3 (cm3).
2. Under Forward Open Connection Path, enter s0.
3. Under Target Device, click SLC5 MicroLogix Protected Logical Read/Write with 3-addresses.
4. When you finish the last destination, uncheck Continue ... and click Next.



### 4.17 Finished—review the settings

Here is the final summary. You may need to scroll to see all the settings.

After you review the settings, click **Finish**.



#### 4.18 Reboot the Digi One IAP

You can make minor changes to the Digi One IAP configuration. However, it is safest to reboot when you change the number or type of masters (message sources) or number or type of slaves (message destinations). These changes affect the number and type of tasks running in the Digi One IAP RTOS, and occasionally they fail to take effect without a reboot.

### 5 Setup by telnet release “E”

To log onto your Digi One IAP, use HyperTerminal or any telnet application. Enter the IP address of your DS and the well-known telnet port of 23.

You can use the text script shown next. Cut and paste the script into an editor such as Wordpad. Edit the script as necessary, then cut and paste again into HyperTerminal using **Edit > Paste to Host**.

Here’s what this script does:

1. Clears all IA configurations.
2. Creates Modbus/RTU serial master and AB/DF1 serial PLC on the second port.
3. Enables incoming Modbus/TCP and AB/Ethernet masters.

4. Creates four virtual Modbus/RTU slaves:

- Slave #1 is the DF1 PLC
- Slave #2 is the remote ControlLogix
- Slave #3 is the remote PLC5E
- Slave #4 is the remote Modbus/TCP PLC

Note that protocol mapping is used only when required. So when RSLinx, the PLC5E or ControlLogix issues PCCC messages for the AB/DF1 serial PLC, these PCCC messages are passed through without change.



```
# clear all IA config
revert ia=factory

# setup port 1 as Modbus/RTU master (baud = 9600,8,E,1) set port ra=1 dev=ia
set line ra=1 baud=9600 csize=8 parity=E stopb=1
set ia serial=1 protocol=mbrtu type=master chrtout=50ms table=1

# setup port 2 as AB/DF1 peer (can be master or slave) (baud = 19200,8,N,1)
set port ra=1 dev=ia
set line ra=1 baud=19200 csize=8 parity=N stopb=1
set ia serial=1 protocol=df1fd type=master table=1

# setup network for Modbus/TCP incoming
set ia master=1 active=on protocol=modbustcp transport=tcp ipport=502 table=1

# setup network for AB/Ethernet incoming
set ia master=2 active=on protocol=abethernet transport=tcp ipport=2222 table=1

# setup destination table set ia table=1 name=table1

# setup slave #1 as the MicroLogix on the Digi One IAP's 2nd port
set ia table=1 addroute=1 active=on protocol=df1fd protaddr=0-1 set ia table=1 route=1
type=serial port=2 target=slc5

# setup slave #2 as the Modbus/TCP PLC
set ia table=1 addroute=2 active=on protocol=mbtcp protaddr=2 set ia table=1 route=2
type=ip ipaddress=192.168.1.27 ipport=502

# setup slave #3 as the AB/PLC5E with SLC5 style messaging
set ia table=1 addroute=3 active=on protocol=abethernet protaddr=3
set ia table=1 route=3 type=ip ipaddress=192.168.1.97 ipport=2222 target=slc5

# setup slave #2 as the ControlLogix CPU in Slot1 with SLC5 style messaging
set ia table=1 addroute=4 active=on protocol=ethernetip protaddr=4
set ia table=1 route=4 type=ip ipaddress=192.168.1.123 ipport=44818 set ia table=1
route=4 class=cm3 connpath="s0" target=slc5

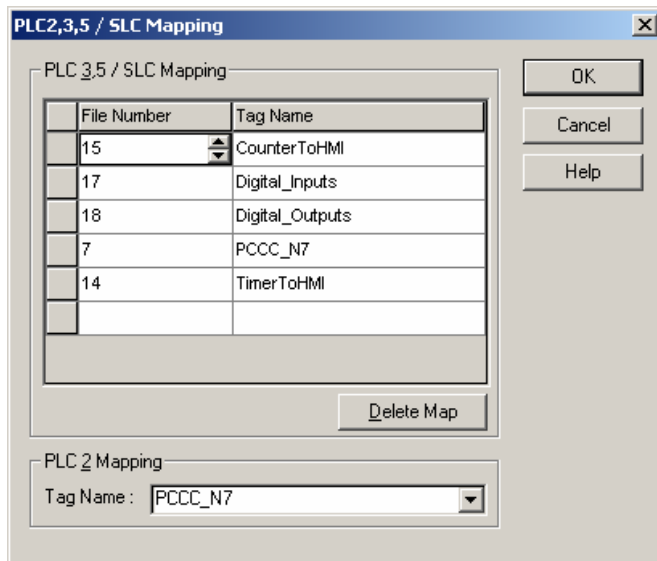
# reboot the Digi One IAP
boot action=reset
```

## 6 Setting up the ControlLogix for PCCC messaging

This document is based on the assumption that you understand the basics of programming and accessing your ControlLogix PLC. By default, the ControlLogix does *not* understand PCCC messages that are aimed at it by DF1 or as encapsulated in Ethernet/IP. You must turn on this function and map existing ControlLogix data tags to the PCCC notion of files.

### 6.1 Enable “map PLC/SLC messages ...”

1. Run RSLogix5000.
2. Select **Logic > Map PLC/SLC Messages**. The dialog box shown next opens and is empty by default. (See figure below.)
3. Map tags to tables.



In this example, a data tag named PCCC\_N7 of 150 words is mapped to both PLC2/CIF messages and as file number 7.

There is nothing special about this name. It could have been “words” or “donuts”. Because most PCCC messages include a byte count, you don’t define the file type. ControlLogix does its best to map the tag data to the file type in the PCCC.

## 6.2 Usage

You might need to integrate any new tags into your application. In the previous example, the tag **Digital\_Inputs** is system-defined. Because we have 32-inputs, it creates a B- or N-like table of two words only.

The PCCC\_N7 array of 150 INT data types can be accessed within the ControlLogix as PCCC\_N7[0] to PCCC\_N7[149]. It is more efficient to explicitly have the ControlLogix copy data to/from an array tag like PCCC\_N7 than to have a dozen small two or three-word files like files 14, 15, 17, and 18 above. Modbus can read 125 words in a single message.

## 7 Testing and troubleshooting

### 7.1 Testing by simulator

Protocol mapping is not as easy to test because it introduces many details that can affect the testing. The easiest way to test is to use a Modbus slave simulator that allows you to see the actual polls. One simulator is ModSim32 from [www.win-tech.com](http://www.win-tech.com).

### 7.2 Can I use PLC5 or CIF style commands?

Commands other than CIF/PLC2 Unprotected Reads/Writes and SLC500 style Typed Reads/Writes are processed by the Digi One IAP doing protocol mapping. The results, however, may not be what you want.

The ControlLogix MSG block doesn’t have a preference for PCCC commands. Use the SLC Typed Read/Write if you want predictable results as documented in this application note.

### 7.3 Why is my MSG block returning a bad status?

If your Ethernet is working, and the ControlLogix is able to create an Ethernet/IP connection to the Digi One IAP, two conditions that can cause an error response include:

- The MSG block is sending something other than SLC Typed Reads/Writes.
- The MSG block is requesting file types other than “N” or “B”.

Although your Modbus master is issuing Modbus polls, the Digi One IAP must turn these into actual PCCC commands for Rockwell slaves to be able to answer. By default, these will be in SLC500 format, although you can manually configure other target sets such as CIF (also called 485CIF or PLC2).

### 7.4 Given Rockwell notation, determine your Modbus

Download the [Digi Protocol Bridging Calculator](#) (part number 90000653), and use this Excel spreadsheet to convert your existing Rockwell notations into the appropriate Modbus polls. With the Digi One IAP, it is important to determine the correct Modbus address to poll the Rockwell register.

### 7.5 System documentation

Because the Digi One IAP converts Modbus requests to Rockwell by a fixed formula, you can create simple documentation. The example below shows 6 words stored in a Rockwell N10 table. Because fixed Modbus polls are required, you can document and have a permanent record.

AB	Modbus	Item (parameter)	Set (monitor) value
N10:0	4x00251	Process value	<ul style="list-style-type: none"> <li>• Temperature: Follow the specified range of the sensor.</li> <li>• Analog: Scaling lower limit -5%FS to scaling upper limit +5%FS</li> </ul>
N10:1	4x00252	Status	Refer to Status on page XXX
N10:2	4x00253	Internal set point	SP lower limit to SP upper limit
N10:3	4x00254	Heater current monitor	0000 to 0226 (0.0 to 55.0)
N10:4	4x00255	MV monitor (heat)	<ul style="list-style-type: none"> <li>• Standard: FFCE to 041A (-5.0 to 105.0)</li> <li>• Heating and cooling: 0000 to 041A (0.0 to 105.0)</li> </ul>
N10:5	4x00256	MV monitor (cool)	0000 to 041A (0.0 to 105.0)