

**Wonderware®**

**Modicon MODBUS Ethernet I/O Server  
User's Guide**

**Revision E**

**Last Revision: September 2003**

**Invensys Systems, Inc.**

All rights reserved. No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Invensys Systems, Inc. No copyright or patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this documentation, the publisher and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The information in this documentation is subject to change without notice and does not represent a commitment on the part of Invensys Systems, Inc. The software described in this documentation is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of these agreements.

**© 2003 Invensys Systems, Inc. All Rights Reserved.**

Invensys Systems, Inc.  
26561 Rancho Parkway South  
Lake Forest, CA 92630 U.S.A.  
(949) 727-3200  
<http://www.wonderware.com>

### **Trademarks**

All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized. Invensys Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArcestrA, Avantis, DBDump, DBLoad, DT Analyst, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A<sup>2</sup>, InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, InTrack, MaintenanceSuite, MuniSuite, QI Analyst, SCADAAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware, and Wonderware Logger are trademarks of Invensys plc, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

# Contents

**Wonderware Modicon MODBUS Ethernet I/O Server ..... 5**

- Introduction ..... 5
  - Communication Protocols ..... 5
  - Accessing Remote Items via the I/O Server ..... 7
- Getting Started ..... 7
- Configuring the I/O Server ..... 8
- Configuring a Topic Definition ..... 9
  - Topic Definition ..... 9
  - MBENET Topic Definition ..... 10
- Saving the I/O Server’s Configuration File ..... 13
  - Save Configuration ..... 14
  - Saving Multiple Configuration Files ..... 14
- Configuring the I/O Server Settings ..... 14
  - Server Settings ..... 15
- Accessing I/O Server Help ..... 16
  - Contents ..... 17
  - How to Use Help ..... 17
  - About Modicon MODBUS Ethernet ..... 17
- Item Names ..... 18
  - Special Item/Point Naming Conventions ..... 20
- Monitoring the Status of Communications with a PLC ..... 25
  - Using the Status Item in Excel ..... 25
- Monitoring the Status of Communications with InTouch ..... 25
  - Using DDEStatus and IOStatus in Excel ..... 25
  - Reading Values from the I/O Server into Excel ..... 26
- Writing Values to the I/O Server from Excel ..... 26
- Troubleshooting I/O Server Communication Problems ..... 28
  - Debugging Communication Between InTouch and an I/O Server ... 28
  - .Debugging Communication Between SuiteLink and an I/O Server 31
  - Debugging Communication Between an I/O Server and a PLC ..... 31



# Wonderware Modicon MODBUS Ethernet I/O Server



## Introduction

The Wonderware® Modicon MODBUS Ethernet I/O Server (referred to as the server through the remainder of this user's guide) is a Microsoft® Windows® application program that allows access to data in Modicon PLCs (also referred to as devices) over an Ethernet network. The server requires only a standard 10BaseT Ethernet network card (or a Ethernet card with fiber optic output) to access the Ethernet network. Each Ethernet Network Card provides an interface to as many PLC's as the Ethernet topology allows. The server supports Modicon TSX Quantum Automation Series controllers equipped with Quantum Ethernet TCP/IP NOE modules (P/N 140 NOE 211 00 for 10BaseT twisted pair or P/N 140 NOE 251 00 for fiber optic).

The current release of the server has added support for Modicon's P/N 174 CEV 200 30 MODBUS Plus to Ethernet Bridge and Modicon's P/N 174 CEV 300 10 MODBUS to Ethernet Bridge.

While the server is primarily intended for use with Wonderware InTouch® (version 3.01 and later), it may be used by any Microsoft Windows program which is capable of acting as a DDE, FastDDE or SuiteLinkä client.

## Communication Protocols

Dynamic Data Exchange (DDE) is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as InTouch and Microsoft Excel can simultaneously be both a client and a server.

FastDDE provides a means of packing many proprietary Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server. Although Wonderware's FastDDE has extended the usefulness of DDE for our industry, this extension is being pushed to its performance constraints in distributed environments.

NetDDE™ extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM® compatible computers connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS™ and UNIX®.

SuiteLink uses a TCP/IP based protocol and is designed specifically to meet industrial needs such as data integrity, high-throughput, and easier diagnostics. This protocol standard is only supported on Microsoft Windows NT 4.0 and Windows 2000.

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink was designed to be the industrial data network distribution standard and provides the following features:

- Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.
- Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT and Windows 2000 operating systems Performance Monitor. This feature is critical for the scheme and maintenance of distributed industrial networks.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

## Accessing Remote Items via the I/O Server

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the application name, topic name and item name. The following briefly describes each portion of this naming convention:

<i>application name</i>	The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to Modicon equipment via this server, the application portion of the address is <b>MBENET</b> .
<i>topic name</i>	Meaningful names are configured in the server to identify specific devices. These names are then used as the topic name in all conversations to that device. For example, <b>Node209</b> .  <b>Note</b> You can define multiple topic names for the same device (PLC) to poll different points at different rates.
<i>item name</i>	A specific data element within the specified topic. For example, when using this server, an item can be a relay, timer, counter, register, etc., in the PLC.  <b>Note</b> The item/point names are predefined by the server. The term "point" is used interchangeably with the term "item" in this user's guide.

☞ For more information on item/point names, see the "Item Names" section later in this user's guide.

## Getting Started

Wonderware supplies a version of the server for Microsoft Windows NT and Windows 2000. No special Kernel Device Driver is needed. It uses the standard TCP/IP protocol stack that comes with Windows NT and Windows 2000 operating system. If you have your Ethernet adapter card configured for accessing the Ethernet network, it will work without modification for communicating with the Modicon Quantum Series PLCs.

1. Configure the Modicon MODBUS Ethernet module to have a unique valid IP Address on the network which it is connected. You may need to see your Network Administrator to obtain valid IP Addresses.
2. Configure TCP/IP on your PC with a unique IP Address.
3. Make sure that the PC and the PLC or Bridge are either on the same subnet or are connected by routers and/or bridges.
4. To make sure that your PC can communicate with the PLC or Bridge, try pinging the IP Address of the Modicon MODBUS Ethernet module that you are trying to access.

## Configuring the I/O Server

Once the server has been installed, a small amount of configuration is required. Configuring the server automatically creates a configuration file named, MBENET.CFG. This file stores the configuration information for the adapter cards or communication ports and all of the topic definitions (described in detail later).

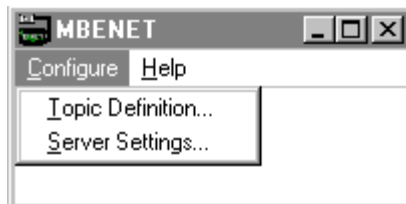
The configuration file is automatically saved to the directory in which the server is installed unless a different directory is specified.

To perform the required configurations, start up the server by double-clicking on its icon. If the server starts up as an icon, double-click on the icon to open the server's window.



To perform the required configurations, start up the server by double-clicking on its icon. If the server starts up as an icon, double-click on the icon to open the server's window.

To access the options used for the various configurations, open the **Configure** menu:



---

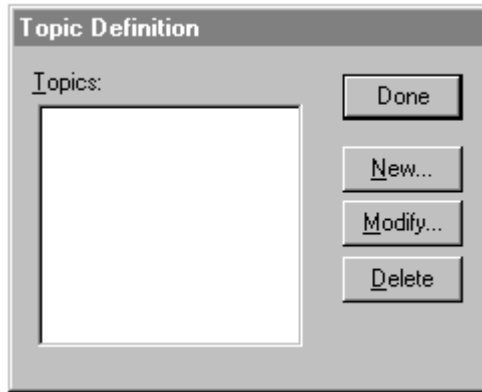
**Note** If any of the options appear grayed, these options are not available with this software version.

---



## Configuring a Topic Definition

Use the Topic Definition option from the Configure menu to create new, modify, or delete topic definitions. One or more topic definitions must exist for each PLC that the server will communicate with. Each topic definition must contain a unique name for the PLC associated with it. When this option is selected, the **Topic Definition** dialog box will appear:



### Topic Definition

Note: Once topics have been defined, their names will be listed in the Topics section of this dialog box.



Click this button to close the dialog box and accept any new definitions, modifications or deletions made.



To modify or view an existing topic definition, select its name in the list and click on this button. The **MBENET Topic Definition** dialog box (described below) will appear displaying the selected topic definition.



To delete an existing topic definition, select its name in the list and click on this button. (A message box will appear prompting you to confirm the deletion.)



To add a new topic definition, click on this button. The **MBENET Topic Definition** dialog box will appear.

The number of new topics per PLC (IP Address) is limited to 16 topics per PLC.

## MBENET Topic Definition

Enter a unique name (up to 32-characters long) for the PLC in this field, for example **Node209**.

**Note** When communicating with InTouch, this **exact** name is used as the topic name in the Access Name definition.

Enter the unique **IP Address** that was configured in the Quantum PLC. If your PC has access to a Domain Name Server or a HOST file, then enter the name of the Quantum host PLC. The HOST file is an ASCII text file that maps user friendly host names to IP Addresses. An example of a HOST file entry would be:

**10.32.200.209                  Wonder0                  wonder0**

In this case, enter **Wonder0** or **wonder0** in the edit box for the IP Address.

If the PC is located on a remote network, then a fully qualified domain name may be required in the HOST file. For example:


**10.32.200.209                  mbe2.wonderware.com**

In this case, enter mbe2.wonderware.com in the edit box. It is recommended to enter the IP Address, if available. This is faster because the server does not have to resolve the name to an IP Address.

The number of new topics per PLC (IP Address) is limited to 16 topics per PLC.

Dest\_Index or Unit\_ID:

Enter the **Dest\_Index** if you are using the MODBUS Plus to Ethernet bridge. Enter the **Unit\_ID** if you are using the MODBUS to Ethernet bridge. This value may contain any integer from 0 to 255.

Slave Device Type:  

Use Concept Data Structures

Communication Channels

Unsolicited Messages

484 PLC  
584/984 PLC  
Quantum (6 digit address)  
Micro84 PLC  
Daniel Flow Computer  
Elliot Flow Computer  
Micromotion Flow Computer

From the drop down list box, select the **Slave Device Type**. The **Block I/O Sizes** will automatically change when a new slave type is selected.

Use Concept Data Structures

Select this option to configure the server to be compatible with the Modicon Concept software data structures. Only active for six digit devices.

Communication Channels

Unsolicited Messages

Check the **Unsolicited Messages** box to allow the server to accept unsolicited messages.

The Quantum PLC must be programmed using a MSTR block to send messages back to the PC that is running the server. Use the IP Address of the PC in the MSTR block. If the Update Interval value is non-zero, the server will poll for the data in this topic in addition to accepting unsolicited messages. For unsolicited messages to work, your InTouch application must have the items on advise. Program the MSTR block to send messages that contain data for these items.

String Variable Style

Full length

C style

Pascal style

Select the option for the style the server uses to store ASCII strings into the PLC registers.

---

**Note** For more information on the **String Variable Styles**, see the "Item Names" section later in this user's guide.

---

A dialog box titled "Register Type" with two radio button options: "Binary" (selected) and "BCD".

Select the option for the **Register Type** being used.

The following **Block I/O Size** examples are for the Quantum Series Slave Type.

A text input field labeled "Coil Read:" containing the value "1600".

Enter the maximum number of consecutive coils to be read at one time in this field. In this example, the valid **Coil Read** values can be between 8 to 1600 and must be an even multiple by 8.

A text input field labeled "Coil Write:" containing the value "800".

This field is used to enter the maximum number of consecutive coils that can be written to at one time. In this example, the valid **Coil Write** values can be between 8 to 800 and must be an even multiple by 8.

A text input field labeled "Register Read:" containing the value "100".

Enter the maximum number of consecutive registers to be read at one time in this field. In this example, the valid **Register Read** values can be between 1 to 100.

A text input field labeled "Register Write:" containing the value "100".

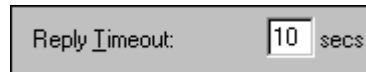
Enter the maximum number of consecutive registers that can be written to at one time in this field. In this example, the valid **Register Write** can be between 1 to 100.

A text input field labeled "Update Interval:" containing the value "800" followed by the unit "msecs".

Setting an **Update Interval** of 0 is equivalent to setting an infinite poll interval. This is useful for **Unsolicited Messages** (usually alarm messages). When an item is put on advice for a topic with 0 **Update Interval**, the initial data for that item will be polled only once by the server. That item will never be polled again. It then waits for the PLC to send data for the item as an **Unsolicited Message**. This is usually done by programming the MSTR Block in the PLC. (Refer to Modicon's documentation for details.) An example would have the PLC monitoring a thermocouple input of a furnace's temperature. The PLC is programmed to read the thermocouple data and convert this reading to Engineering Units as an IEEE float. This data is then stored as a float in register pair 400100 and 400101.

Set up a topic called **Furnace Temperature** with an **Update Interval** of 0. To read the data as a float for this topic, put the following item on advice: 400100 F.

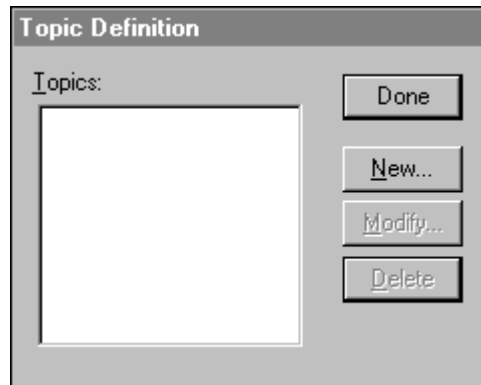
Program the PLC to monitor the data in registers 400100 and 400101. If an overheat (alarm) condition occurs, program the PLC to execute the MSTR Block. This could be programmed to send the data in register block 400100 to 400101 to the target PC workstation where the server is running. By setting the **IP Address** of the PC workstation, the client program in the PC workstation could be set up to display the data.



Enter the amount of time (in seconds) that the PLCs for this topic will be given to reply to commands from the server.

Click **Cancel** to close the dialog box without saving changes.

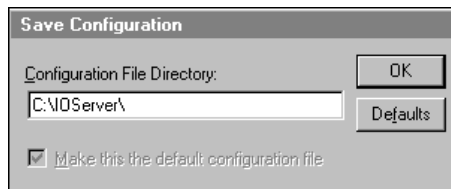
Click **OK** to accept the topic definition and return to the **Topic Definition** dialog box.



Click **Done** to close this dialog box.

## Saving the I/O Server's Configuration File

If a configuration file does not currently exist in the configuration file directory, the server will automatically display the **Save Configuration** dialog box:



## Save Configuration

Configuration File Directory:  
C:\IOServer\

This field displays the drive\directory into which the server will save the current configuration file. To save the configuration file to a different directory, enter the path for that directory in this field.

Make this the default configuration file

This option is selected and disabled on initial entry to the **Save Configuration** dialog box. This field becomes active if the Configuration File Directory is changed. Once enabled, selecting this option will record a new Configuration File path in the **WIN.INI** file. This option allows the server to find its configuration file automatically each time it is started.

---

**Note** When the server initially starts up, it attempts to locate its default configuration file by first checking the **WIN.INI** file for a previously specified path. If a path is not present in the **WIN.INI** file, the server will assume that the current working directory is to be used.

---

Click **Defaults** to reset the settings to their default values without saving changes.

Click **OK** to save the configuration file to the specified directory.

## Saving Multiple Configuration Files

There is no limit to the number of configuration files that you can create as long as they are saved in separate directories. This allows you to have multiple configuration files that can be accessed by using a special switch (/d:). For example, to start the server using a configuration file located in a different directory, click the **Start** button on the Taskbar, then choose **Run** and enter the following in the **Open** combo box:

```
MBENET /d:c:\directoryname
```

## Configuring the I/O Server Settings

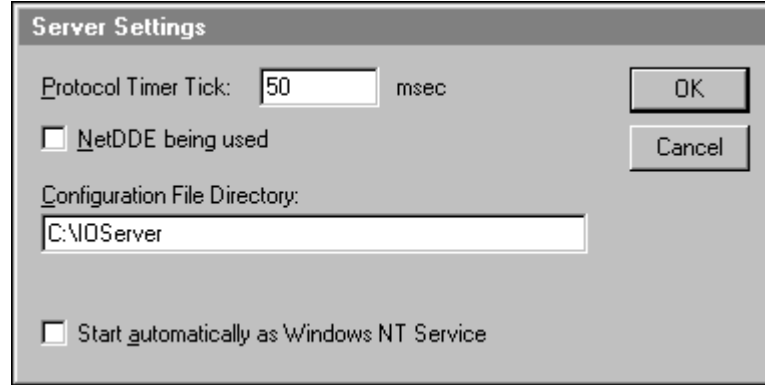
Use the **Server Settings** option from the Configure menu to change the protocol timer, network using Wonderware NetDDE, change the default configuration file path, or to enable the server to start automatically as a Windows NT or Windows 2000 service.

---

**Note** When configuring the server on Windows NT or Windows 2000, the user must be logged on with system administrator privileges. This will ensure that updates to the system registry may be performed.

---

When the **Server Settings** option is selected, the **Server Settings** dialog box will appear:



## Server Settings

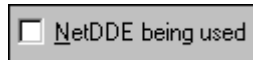


Enter the frequency (in milliseconds) that the server is to check for data to process. This should be approximately two to four times faster than the fastest rate desired to update data from the equipment.

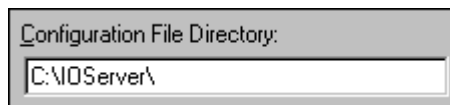
---

**Note** The default protocol timer tick value will vary between servers.

---



Select this option if you are networking using Wonderware NetDDE.

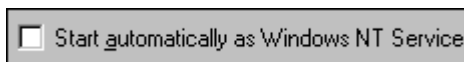


To create a new default configuration file, enter the complete path for the directory in which the file is to be saved in this field. This new path will automatically be written to the **WIN.INI** file and the server will use this path to load its configuration file the next time it is started.

---

**Note** There is no limit to the number of configuration files created. However, each must be saved in a different directory. When using the server with InTouch, we recommend that you save the configuration file in your application directory. For more information on the Configuration File, see "Saving the I/O Server's Configuration File" in this user's guide.

---



Enabling this option will cause the server to start as a Windows NT and Windows 2000 operating system service.

Windows NT offers the capability of running applications even when a user is not logged on to the system. This is valuable when systems must operate in an unattended mode. Enabling this option and rebooting the system will cause the server to run as a Windows NT service. However, to view configuration information or to reconfigure the server, the user must log on to the system. Any server related problems that may arise such as missing adapter cards, licensing failures or device drivers not loading will not be visible to the user until a log on is performed. Disabling this option and rebooting the system will cause the server to run as a Windows NT application program once again.

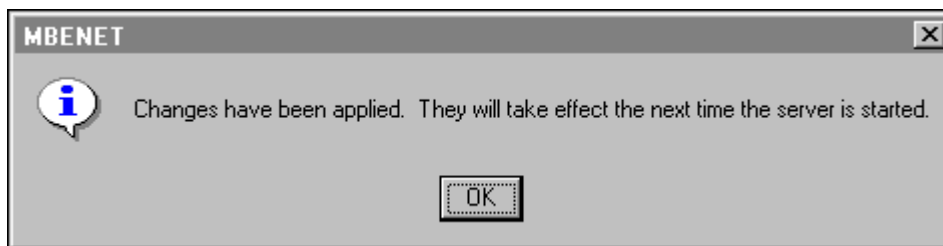
---

**Note** It is highly recommended that the server is configured and communicating successfully prior to running it as a Windows NT and Windows 2000 operating system service.

---

Click **Cancel** to close the dialog box without saving changes.

Click **OK** to accept the server settings. The following message box will appear:



Click **OK** to close the dialog box.

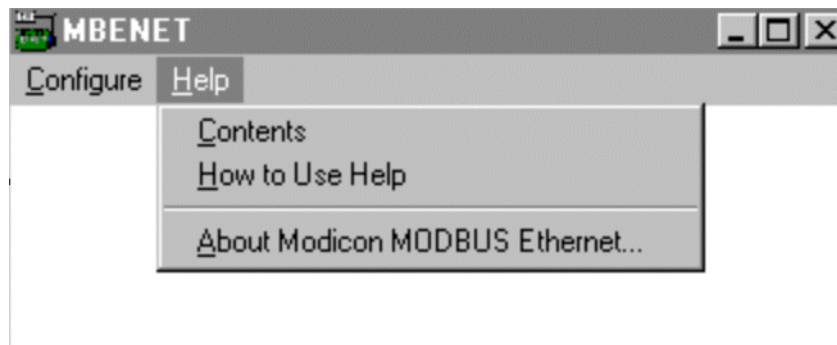
---

**Note** You must restart the server for the changes to take effect.

---

## Accessing I/O Server Help

The Help menu contains three options that are used to access help for the server.



The following briefly describes the Help menu options.



## Contents

This option is used to display the table of contents for the Help file.

## How to Use Help

This option is used to access a list of basic instructions for using the Help file

## About Modicon MODBUS Ethernet

This option is used to access miscellaneous information regarding the server such as the software version, the copyright information, license information, etc.

Your FactorySuite system license information can be viewed through the license viewing utility that is launched from the About dialog box.

- For more information on the license viewing utility, see your online *FactorySuite System Administrator's Guide*.

## Item Names

The server supports item/point names that are consistent with the point naming convention used by Modicon PLCs. The server allows you to select a Slave Type when you configure the topic definition for the PLC. The PLC Address Ranges supported are:

PLC Type	Point Type	Range	Tag Type	Access
484	Coil	1-999	Discrete	Read/Write
	Contact	1001-1999	Discrete	Read Only
	Input Register	3001-3999	Analog	Read Only
	Holding Register	4001-4999	Analog	Read/Write

:

PLC Type	Point Type	Range	Tag Type	Access
584/984, Micro84	Coil	1-9999	Discrete	Read/Write
	Contact	10001-19999	Discrete	Read Only
	Input Register	30001-39999	Analog	Read Only
	Holding Register	40001-49999	Analog	Read/Write
	*Ext Memory Register	600001-69xxxx	Analog	Read/Write

:

PLC Type	Point Type	Range	Tag Type	Access
Quantum (6-Digit)	Coil	1-65536	Discrete	Read/Write
	Contact	100001-165536	Discrete	Read Only
	Input Register	300001-365536	Analog	Read Only
	Holding Register	400001-465536	Analog	Read/Write
	*Ext. Memory Register	600001-69xxxx	Analog	Read/Write

\*Extended Memory Registers are supported on the Quantum Automation Series PLCs.

Addresses are assigned as follows:

File#	Range
1	600001-610000
2	610001-620000
3	620001-630000
4	630001-640000
5	640001-650000
6	650001-660000
7	660001-670000
8	670001-680000
9	680001-690000
10	690001-69xxxx

---

**Note** Analog tagnames can be either Integer or Real.

---

The supported Item Names for **Daniel FC**, **Elliott FC**, **Omni FC**, and **Micromotion** slave device types are consistent with the Item Names used by Modicon PLCs. However, the Item Name and Read/Write Access may vary depending upon the application being used. For more information, see your user's guide for the device being used. The Flow Computer Address Ranges supported are:

FC Type	Point Type	Range	Tag Type
Daniel	Coil	1001 - 2999	Discrete
	Holding Register	3001 - 4999	Integer
	Holding Register	5001 - 6999	Integer
	Holding Register	7001 - 65535	Real

FC Type	Point Type	Range	Tag Type
Elliot	Coil	1001 - 2999	Discrete
	Holding Register	3001 - 4999	Integer
	Holding Register	5001 - 6999	Integer
	Holding Register	7001 - 65535	Real

FC Type	Point Type	Range	Tag Type
Omni	Coil	1001-2999	Discrete
	Holding Register	3001 - 4999	Integer
	Holding Register	4001-4999	String
	Holding Register	5001-6999	Integer
	Holding Register	7001-7999	Real

FC Type	Point Type	Range	Tag Type
	Holding Register	13001-13999	Integer
	Holding Register	14001-14999	String
	Holding Register	15001-16999	Integer
	Holding Register	17001-17999	Real

FC Type	Point Type	Range	Tag Type
Micromotion	**Coil	00002 - 00020	Discrete
	**Coil	00113 - 00160	Discrete
	**Read-Only Discrete	10021 - 10038	Discrete
	Input Register	30001 - 30011	Integer
	Input Register	30120 - 30126	Integer
	Holding Register	40012 - 40051	Integer
	*Strings	40052 - 40119	String
	Holding Register	40124 - 40140	Integer
	***Floating Point Register Pairs	40141 - 40278	Real

\*Registers 50052 - 50119 map to 40052 - 40119. To access the first ASCII string located at 50052 - 50055, use item name 40052 - 40055 M. The other ASCII strings are accessed in the same way.

\*\*Caution! Advise only items within the ranges shown.

\*\*\*To access a float located at register pair 20141 - 20142, use the item name 40141 F. Other floats are accessed in the same way.

Floating Point Registers must be accessed as pairs only.

## Special Item/Point Naming Conventions

Special handling of data from Modicon equipment can be done by using the following conventions:

### Signed Registers

The server normally allows register values in the range of 0 to 65535. Registers may be treated as 16-bit signed quantities having values between -32,768 and 32,767.

To specify that a register is to be treated as a signed quantity, append a space and the letter **S** to the item name. For example, to indicate that the first Holding Register is signed, enter "**400001 S**" for the item name.

## Long Integers

Pairs of registers can be treated as 32-bit signed integers. This is done by appending a space and the letter **L** to the item name. For example: **400001 L**

## Floating Point

Pairs of registers can be treated as IEEE 32-bit floating point numbers. This is done by appending a space and the letter **F** to the item name. For example: **400001 F**

## Bits In Registers

Individual bits in registers can be read as discrete tags by using the notation **rrrr:b**.

Where **rrrr** specifies a valid input register or holding register and **b** specifies a bit position between **1** and **16** (**1** specifies the most significant bit of the register).

Examples:

**400001:1** - Most significant bit of first holding register

**300008:16** - Least significant bit of an input register

---

**Note** Bits in registers are read-only. The Ethernet protocol has no commands to write individual register bits.

---

## Explicit Conversions

Registers are normally treated as integers. This can be altered for specific registers by adding a blank space (using the spacebar) and one of the following to the point name:

**L** - Long

**F** - Float

**I** - Integer

**M** - Message (ASCII Strings)

## Modulo-10000 Points

Two or three consecutive registers may be interpreted as a single numeric quantity. In doing so, each of the component registers must be in the range of 0-9999. For example two registers "**400001-400002**" can represent numbers between 0 and 99999999.

Overflow becomes a possibility when grouping three consecutive registers for interpretation as a single numeric quantity. The largest number that may be represented in the PLC with three consecutive Modulo-10000 registers is 999,999,999,999. Unfortunately, the largest number which can be contained in an 'integer' type variable is 2,147,483,647. The latter number is used by the server to represent an overflow condition. Therefore, the maximum usable value represented in three Modulo-10000 registers is 2,147,483,646 ( or  $< 21 \times 4748 \times 3646 >$  ). Any numbers larger than this will be clamped at 2,147,483,647.

## ASCII Strings

Multiple consecutive registers (1-100) can be treated as a string of ASCII characters. The first character of the string is in the high-order 8-bits of the lowest numbered register. Three options are provided for the representation of variable length strings:

- A. The string may be padded with ASCII spaces (hex 20).
- B. The string may be terminated with a zero byte.
- C. The length may be stored in the first byte.

Strings of ASCII characters can be stored in consecutive registers in a PLC. The number of registers that are allocated to the storage of an individual string must be fixed. However, the string contained in those registers can have a variable length up to twice the number of registers allocated. Strings are stored in the PLC registers from lowest numbered register to the highest and within each register, first the most significant byte, then the least significant byte. The server supports three methods of storing strings of ASCII characters in registers:

## Full Length

If strings are stored in this manner, the string always uses all of the registers allocated. If the string is shorter than the allocation of registers, it is padded with ASCII space characters (hex 20) to the full length. For example, string item "400001-400010 M" containing the ASCII string "Wonderware":

	MSB	LSB	
400001	57	6F	"Wo"
400002	6E	64	"nd"
400003	65	72	"er"
400004	77	61	"wa"
400005	72	65	"re"
400006	20	20	" "
400007	20	20	" "
400008	20	20	" " padded to the end
400009	20	20	" " with ASCII spaces
400010	20	20	" "

## C Style

If strings are stored in this manner, the end of the string is marked by a byte of zero immediately following the last character in the string. This option is so named because this is the way strings are stored in the "C" programming language. For example, string item "40001-40010 M" containing the ASCII string "Wonderware":

	MSB	LSB	
400001	57	6F	"Wo"
400002	6E	64	"nd"
400003	65	72	"er"
400004	77	61	"wa"
400005	72	65	"re"
400006	0	x	end marked by zero
400007	x	x	the remaining
400008	x	x	bytes
400009	x	x	are
400010	x	x	unused

## Pascal Style

If strings are stored in this manner, the length of the string is denoted by a length byte which occupies the first byte of the string (MSB of the first register). This option is so named because this is the way strings are stored in the "Pascal" programming language (for most compilers). For example: string item "**400001-400010 M**" containing the ASCII string "Wonderware":

	MSB	LSB	
400001	0A	57	length = 0A (hex) "W"
400002	6F	6E	"on"
400003	64	65	"de"
400004	72	77	"rw"
400005	61	72	"ar"
400006	65	x	"e"
400007	x	x	the remaining
400008	x	x	bytes
400009	x	x	are
400010	x	x	unused

## Absolute Notation

Absolute item naming conventions are available that are independent of PLC model numbers. The absolute notation allows accessing of the four Ethernet data types, each with an address from 0 to 65535. The data types are indicated by the item name suffix characters as follows:

nnnnn <b>DO</b>	(Discrete Output) Refers to the same data Ethernet calls "coils" (valid range is 0 <b>DO</b> through 65535 <b>DO</b> .)
nnnnn <b>DI</b>	(Discrete Input) Refers to the same data Ethernet calls "contacts" (valid range is 0 <b>DI</b> through 65535 <b>DI</b> .)
nnnnn <b>IR</b>	(Input Registers) Refers to the same data Ethernet calls "input registers" (valid range is 0 <b>IR</b> through 65535 <b>IR</b> .)
nnnnn <b>HR</b>	(Holding Registers) Refers to the same data Ethernet calls "holding registers" (valid range is 0 <b>HR</b> through 65535 <b>HR</b> .)
nnnnn <b>PV</b>	(Process Variable) Refers to holding registers but treats them as floating points and assumes 2 registers per floating point number. The valid range is 0 <b>PV</b> through 32767 <b>PV</b> .

The **IR** and **HR** absolute notations can also be combined with the following conversions: **L** (long), **F** (floating point) or **S** (signed.) For example: :

219 <b>HRS</b>	16 bit signed integer
0 <b>HRL</b>	32 bit signed integer
100 <b>HRF</b>	32 bit floating point

## Unsolicited Data

Unsolicited Data has no special point naming conventions. However, due to the architecture of the Modicon PLC's, only 4xxxxx tags may be specified to be read with unsolicited messages.

## Global Data

The server does not support Global Data.



## Monitoring the Status of Communications with a PLC

For each topic name (PLC), there is a built-in discrete item that can be used to monitor the status of communications with the PLC. The discrete item, **Status**, is set to **0** when communication with the PLC fails and is set to **1** when communication is successful.

### Using the Status Item in Excel

The status of the PLC communications can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet:

```
=MBENET|Node209!Status
```

where:

<b>MBENET</b>	Is the name of the server application.
<b>Node209</b>	Is the exact topic name defined in the server for the PLC.
<b>Status</b>	Built-in discrete item used to monitor the status of communications with the PLC.

## Monitoring the Status of Communications with InTouch

InTouch supports built-in topic names called DDEStatus and IOStatus that are used to monitor the status of communications between the server and InTouch. For more information on the built-in topic names DDEStatus and IOStatus, see your online "InTouch User's Guide."

### Using DDEStatus and IOStatus in Excel

The status of communication between the server and InTouch can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet:

```
=view|DDEStatus!Node209
```

or

```
=view|IOStatus!Node209
```

where:

<b>view</b>	Is the name of the InTouch application.
<b>[DDE][IO]Status</b>	Built-in topic name used to monitor the status of communications between the server and InTouch.
<b>Node209</b>	The exact topic name defined in the server for the PLC.

## Reading Values from the I/O Server into Excel

Values may be read directly into Excel spreadsheets from the server by entering a DDE formula into a cell using the following format:

**=applicationname|topicname!'itemname'**

Example formula:

**=MBENET|Node209!'400001'**

Where:

<b>MBENET</b>	Is the name of the server application.
<b>Node209</b>	Is the exact topic name defined in the server for the PLC.
<b>400001</b>	Is the actual location in the PLC that contains the data value. This is the item name.

In this example, each time the value of **400001** changes in the PLC, the server will automatically send the new value to the cell containing the formula in Excel.

---

**Note** Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

---

## Writing Values to the I/O Server from Excel

Values may be written to the server from Microsoft Excel by creating an Excel macro that uses the **POKE** command. The proper command is entered in Excel as follows:

```
channel=INITIATE("applicationname","topicname")
=POKE(channel,"itemname", Data_Reference)
=TERMINATE (channel)
=RETURN()
```

The following describes each of the above **POKE** macro statements:

```
channel=INITIATE("applicationname","topicname")
```

Opens a channel to a specific topic name (defined in the server) in a particular application name (the executable name less the .EXE) and assigns the number of that opened channel to **channel**.

---

**Note** By using the **channel=INITIATE** statement the word **channel** must be used in the **=POKE** statement instead of the actual cell reference. The **"applicationname"** and **"topicname"** portions of the formula must be enclosed in quotation marks.

---

```
=POKE(channel,"itemname", Data_Reference)
```

**POKEs** the value contained in the **Data\_Reference** to the specified item name (actual location in the PLC) via the **channel** number returned by the previously executed **INITIATE** function. **Data\_Reference** is the row/column ID of the cell containing the data value.

**=TERMINATE(channel)**

Closes the channel at the end of the macro. Some applications have a limited number of channels therefore, they should be closed when finished. **Channel** is the channel number returned by the previously executed **INITIATE** function.

**=RETURN()**

Marks the end of the macro.

---

**Note** Refer to the **.XLM** sample Excel poke macro provided in the installed server's directory. Also refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

---

## Troubleshooting I/O Server Communication Problems

This section provides you with some simple steps that can be taken to ascertain and correct communication problems. The problems described here represent the most probable causes of communication failure.

---

**Note** This is a general troubleshooting guide and for the sake of brevity we cannot cover every possible source of communication problems.

---

### Debugging Communication Between InTouch and an I/O Server

This section explains the most common error situations that can occur when attempting to establish communication between InTouch and a server.

Servers are Window applications that communicate with I/O, PLCs, and/or other data sources. If a server supports either the Microsoft Dynamic Data Exchange (DDE) or the Wonderware SuiteLink protocol, it is capable of communicating with the Wonderware InTouch program.

---

**Note** All Wonderware version 7.0 or later servers support both DDE and SuiteLink. However, the SuiteLink protocol is only supported on the Windows NT (version 4.0 or later) and Windows 2000 operating system.

---

Servers respond to data requests made by other applications. Requesting applications are called clients. When WindowViewer acts as a client and requires the value of an item, it contacts the server and requests the item's value. The server will report the value and update WindowViewer only if a change occurs. All WindowViewer data requests provide information relating an item to a register, coil number, or I/O data point understood by the server. The server uses the information to automatically handle all messages to and from I/O, hardware devices (PLC), and/or other data sources.

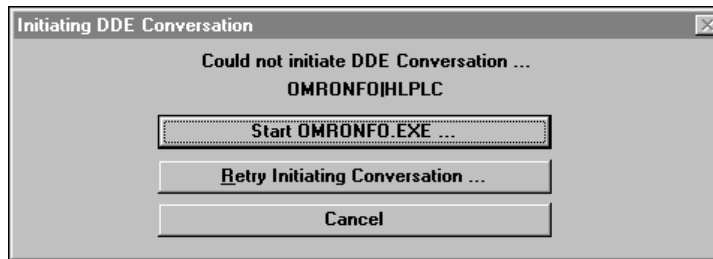
---

**Note** We highly recommend starting all the servers required by the InTouch application before starting WindowViewer. InTouch (versions prior to 7.0) will display the **Initiating DDE Conversation** message box for each uninitiated conversation.

---

For example:

If you start up WindowViewer and cannot successfully establish a conversation with a server, the following Initiating DDE Conversation dialog box will appear:



The information in the second line indicates that you have at least one I/O type tagname defined in your Tagname Dictionary that is associated with an Access Name that defines OMRONFO as the Application Name, and HLPLC as the Topic Name. Make note of exactly how the application and topic names are spelled.

- ☞ This example only applies when using a version of InTouch prior to InTouch 7.0.

To troubleshoot communication problems between WindowViewer and the server, perform the following steps as listed below.

#### Verify the I/O Server is running.

1. Start the server program.
2. Verify the server is running by checking to see if it is in the Windows Task List.

On Windows NT and Windows 2000, click the right mouse button on the Windows taskbar and select Task Manager from the menu. Click the Applications tab to view all currently running applications. Or press the CTRL+SHIFT+ESC keys.

On Windows 95, press the ALT+TAB keys while holding down the ALT key.

On Windows 3.1 or Windows for Workgroups, press the CTRL+ESC keys.

#### If the I/O Server is running, verify that the I/O Server's program name is correct in all WindowMaker Access Name definitions.

1. Switch to (or start) WindowMaker. Select Access Names from the Special Menu, the Access Name Definitions dialog box appears listing all Access Names defined in the WindowMaker.
2. In the Access Names list, select the Access Name referencing the server and click Modify. The Modify Access Name dialog box will appear.
3. Verify the server's program name in the Application Name box is correct. If it is wrong then correct it and click OK, else click Cancel.

- ☞ The server's exact "executable name" must be typed in the Application Name box in all Access Name definitions. The ".exe" extension is not used.
- ☞ If you are debugging a remote tagname reference, also verify that the node name for the remote computer in the Node Name box is correct.

4. Repeat steps 2 & 3 and verify the server program name is correct in all Access Names that use it.

**If you still cannot establish a conversation, verify that the exact topic name used in the WindowMaker Access Name definitions are defined in the I/O Server program.**

1. Close WindowViewer if it is running. The server cannot be configured if WindowViewer is running.
2. Start the server program.
3. From the server's Configure menu select Topic Definition. The Topic Definition dialog box appears listing all topic names defined in the server.
4. Verify that the topic name exists and is spelled exactly the same (including spaces) as the topic name referenced in the WindowMaker Access Name definition.

☞ Blank spaces cannot follow the topic name in either the server's Topic Definition or the Access Name definition.

5. If the topic name is different, either correct it in the server or switch to WindowMaker and correct it in the Access Name definition.
6. Once you performed the above procedure, restart WindowViewer and switch to the server program. Data should now appear in the server's program window to indicate that WindowViewer and the server are communicating.

☞ The data in the server's program window indicates the read and write messages the server is sending to and receiving from the PLC. These are not error messages; only status messages are written to the server's program window.

7. If no data appears in the server's program window, switch to the Wonderware Logger to check for error messages. For example, a common error message is:

"Error for DDE: OMRONFO|HLPLC!<null>("item") Advise failed"

This message appears when the item defined in one or more tagnames is invalid for the server.

☞ InTouch tagnames use specific naming conventions when accessing data from a server. The valid item names for all Wonderware servers are documented in their respective user's guides. Typically, the item naming conventions used by each server are consistent with the names used by the equipment manufacturer.

☞ For more information on the Wonderware Logger, see your online *FactorySuite System Administrator's Guide*.

**If you are still experiencing problems, continue with the following troubleshooting section.**

## Debugging Communication Between SuiteLink and an I/O Server

If you have successfully applied the debug techniques listed in the previous section and are still experiencing communication problems to a server that is attempting to communicate using the SuiteLink protocol, perform the following steps as listed below:

**Verify the I/O Server supports the Wonderware SuiteLink protocol, that is, the I/O Server is version 7.0 or above.**

**Try communicating to the I/O Server using the DDE protocol. If this is not possible, then proceed to the next troubleshooting section otherwise continue with the following steps:**

1. Verify Microsoft's TCP/IP stack is installed and configured properly.
  - ☞ SuiteLink uses the Microsoft TCP/IP stack for its communications even if the client application and the server reside on the same node.
2. If you do not have an Ethernet card to bind to the TCP/IP stack, install the Microsoft Loop Back Adapter.
3. Install the Microsoft TCP/IP stack.

## Debugging Communication Between an I/O Server and a PLC

This section provides you with simple steps to diagnose and correct server to PLC communication problems. The debug techniques listed below address both serial and board servers. Disregard any information that is not applicable to the server type that you are using.

When attempting to establish communication between a server and a PLC, if no data appears in the server's program window and the data items are not updating in WindowViewer, switch to the Wonderware Logger and check for error messages.

☞ For more information on the Wonderware Logger, see your online *FactorySuite System Administrator's Guide*.

For example, some of the most common errors that may appear in the Wonderware Logger for serial servers are:

Response Timeout  
WCRET = -2  
WakeUp = -2  
Receive Overrun  
Framing Errors

---

**Note** Unless specified otherwise, most serial communication based servers are full duplex. If you require a server for half duplex (one that monitors the CTS and RTS lines) or if you are not sure whether the PLC's protocol is full or half duplex, call your PLC supplier.

---

**Check your cabling to the PLC.**

Is it wired correctly? Check for shorts, loose wires, broken wires, crossed wires, and so on.

☞ A continuity tester can be helpful here.

**Verify the I/O Server's serial configuration settings (Parity, Stop Bits, Baud Rate, Handshaking and so on) against the settings in the hardware device.****Verify the communication port is working properly in Windows.**

1. Close the server program.

☞ If you are using a server that requires a TSR, you will not be able to verify that the port is functioning properly while the TSR is running. Stop all TSRs then continue with this procedure. If you confirm that the port functions properly without the TSR running, change your software interrupt (IRQ) to another number, for example, change 60 to 62.

☞ Also, if you are using an AT type computer, two devices cannot share interrupts. Verify that the communication port you are using has a unique interrupt setting.

2. On Windows 3.1 or Windows for Workgroups, start the Terminal program. On Windows 95 or Windows NT and Windows 2000, start the HyperTerminal program.

3. Configure the Terminal (or HyperTerminal) program to use the same communication port with the same settings (baud rate, parity, stop bits and so on) as the hardware device.

4. Connect a null modem cable to a second computer's port.

5. On the second computer, start and configure the Terminal (or HyperTerminal) program with the same settings as the first computer.

6. Verify that you can send data between the two computers.

☞ If you do not have two computers and the computer you are using has another port, start two instances of the Terminal (or HyperTerminal) program with each configured to their own port. Then try communicating between them.

☞ If you have an external modem, connect the modem to the communication port that you are testing to see if you can dial out.

7. If the communication port does not appear to be functioning properly, check your environment files (**AUTOEXE.BAT**, **CONFIG.SYS**, **SYSTEM.INI**, and **WIN.INI**). Look for suspicious programs or drivers that might be taking control of the port or its interrupt before the server is loaded. Always keep your environment files as clean as possible. If you are in doubt about an entry, comment it out.

8. If the previous step was unsuccessful, try another communication port or another computer.



---

**Note** A common misconception of connecting to a PLC with a DOS program and the same communication port will work in Windows is not the case! Windows is an entirely different environment than DOS.

---

### Does your computer lock up?

Verify the COM port's IRQ's do not conflict with each other or with other communication boards in the computer.

### If the PLC or field device has more than one COM port, verify the connection to the correct port.

The COM port on your computer uses the RS-232 hardware communication standard and connects the cable from the COM port to an RS-232 compliant device.

---

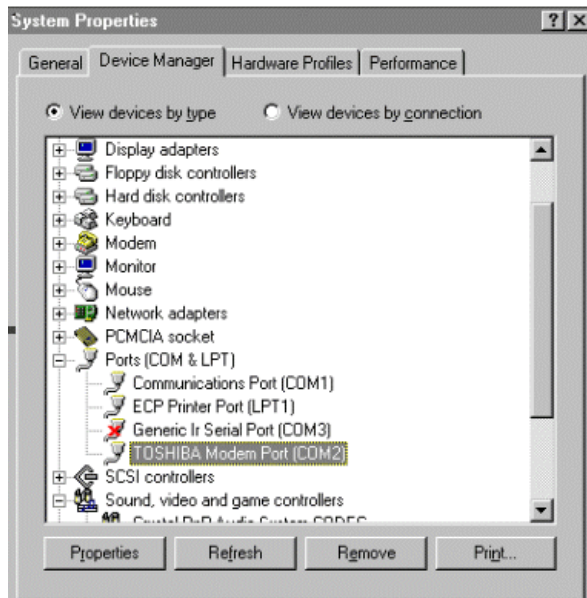
**Note** To connect to an RS-422 or RS-485 port on the PLC, you need an RS-232 to RS-422/485 conversion device.

If possible, use an external converter instead of a board-based converter that plugs into a slot in the computer. A board-based converter is difficult to get working for inexperienced users. If a board-based converter is not set up properly, it can conflict with other communication boards in the computer such as, internal modems.

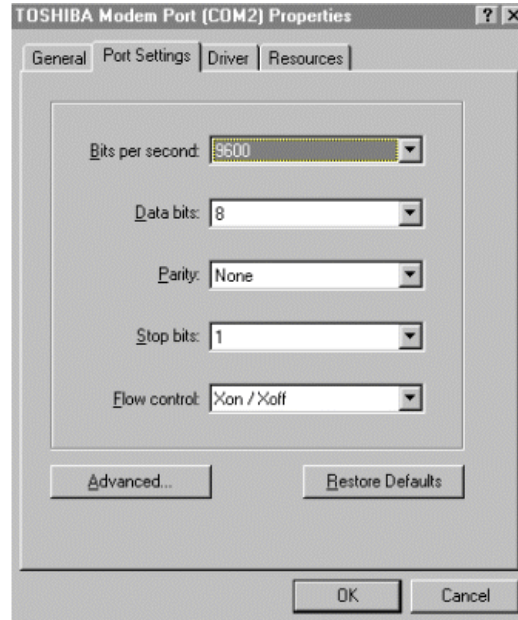
---

### If you are using the Windows 95 operating system, verify the following:

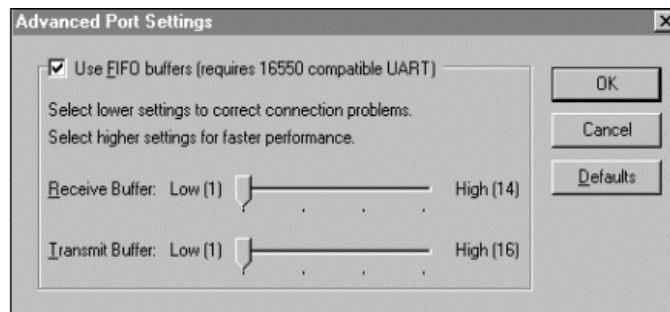
1. Click Start on the Windows taskbar. Point to Settings, then click Control Panel in the menu. The Control Panel dialog box will appear.
2. Double-click the System icon. The System Properties dialog box will appear. Click the Device Manager tab and select the COM port that you are using for the server. For example:



- Click Properties. The Properties dialog box will appear. Click the Port Settings tab:



- Click Advanced. The Advanced Port Settings dialog box appears:

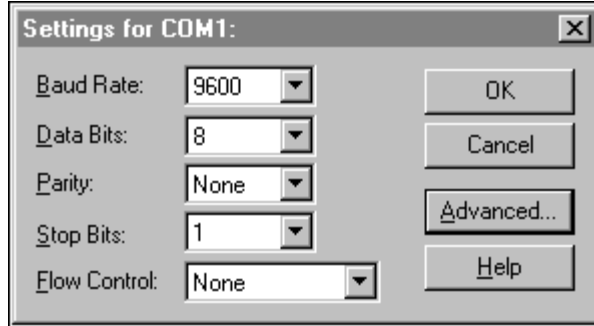


- Lowering the default Receive Buffer and Transmit Buffer settings to their minimum may solve I/O communication problems for portable computers (notebook or laptops) and framing errors for standard computers.
- If using a 16550 UART chip, select the Use FIFO buffers (requires 16550 compatible UART) option. If you are not using a UART chip, make sure this option is not selected.

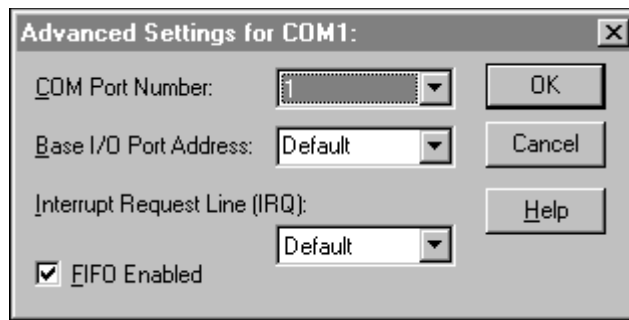
**If you are using the Windows NT or Windows 2000 operating system, verify the following:**

- Click Start on the Windows task bar. Point to Settings, then click Control Panel in the menu. The Control Panel dialog box will appear.
- Double-click the Ports icon, the Ports dialog box will appear.

3. Select a port and click the Settings button. The Settings for COMx dialog box appears:



4. Click Advanced. The Advanced Settings for COMx dialog box appears:



5. Lowering the setting for the Interrupt Request Line (IRQ) value to the minimum may solve I/O communication problems for portable computers (notebook or laptops) and framing errors for standard computers.

### How long is your RS-232 cable?

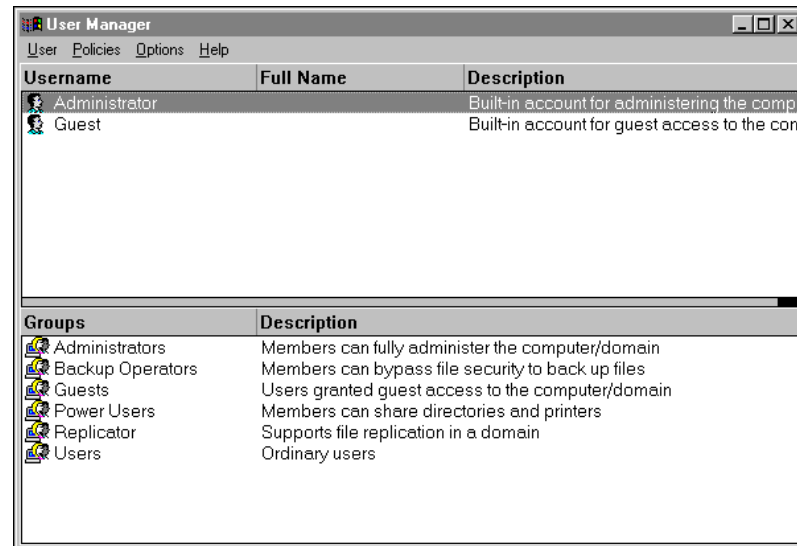
Fifteen meters (fifty feet) is the maximum practical length for the RS-232 standard.

### Try using a different COM port for the I/O Server.

### If you are installing an I/O Server or configuring a board-based I/O Server on a computer running on the Windows NT or Windows 2000 operating system, log on with Administrator privileges.

- Without Administrator privileges, the server and Server Install program cannot make the necessary edits to the Windows NT or Windows 2000 Registry during installation or board configuration of the server.

1. Click Start on the Windows taskbar. Point to Programs, then to Administrative Tools (Common), and click User Manager in the menu. The User Manager dialog box will appear:



2. Double-click the Username you typed in during log on.
3. If the User Properties dialog box does not appear, you do not have Administrator privileges.
4. If the User Properties dialog box does appear, click on the Groups button and verify "Administrators" is in the "Member of" list.

**If you experience occasional or random communication errors in the Wonderware Logger, such as "Response Timeouts," check for noise.**

Do the physical cables run past any known noise sources such as photocopiers, fluorescent lamps, fans, pumps, motors or generators? Are the cables properly shielded from its environment? With radio modems and satellite link ups, occasional communications errors in the Wonderware Logger are normal and to be expected as long as they do not adversely impact the flow of data.

**Increase the Reply Timeout setting in the I/O Server to a value between 5 and 10 seconds.**

Not allowing the PLC or field device enough time to respond to the server's request for data may result in communication errors.

**Verify the PLC is properly configured and the cable is good by using the programming software for the PLC.**

1. Connect to the PLC with the programming software. The connection must be through the same port and cable. Go into the programming software configuration and write down what the communications parameters are (baud rates, routes, node number, error checking, etc.).
2. Close the programming software. Open the I/O Server and verify the communications settings are the same.
3. Poke data into the PLC with InTouch or WWClient.
4. Shut down the server and use the programming software to verify that the values were correctly poked.

☞ Performance of this test depends upon the type of PLC you are using.

**Reinstall the I/O Server and verify that you are using the latest version.**

Wonderware is continually improving our servers and using the latest version will guarantee the best results.

☞ New versions of the Wonderware I/O Servers are released regularly on the Knowledge Base CD. These are available to Comprehensive Support customers on the the Wonderware Bulletin Board System (949-727-0726) or from the Wonderware WEB site at: <http://www.wonderware.com>.

**Move the I/O Server's configuration file to another location on the computer's hard drive. This will clear all configuration for the I/O Server, then reconfigure the I/O Server.**

☞ Wonderware server configuration files are typically the exact same name as the server's executable name with the .CFG extension. For example, **OMRONFO.CFG**. Refer to the Configuration File section of the specific server user's guide for the exact name of the configuration file.

**If possible, reinstall the Windows operating system.**

Files installed earlier on your computer or the NT registry may have been corrupted or accidentally modified.

**If these troubleshooting suggestions do not solve your problem, there may be a problem with your computer. There are many subtle differences between the various computer hardware brands. Using a computer that is a different brand and meets the following criteria:**

1. Select a different PC manufacturer and if this is not possible, try a different PC model from the same manufacturer.
2. The computer can not use an OEM (Original Equipment Manufacturer) version of Microsoft Windows. We highly recommend using only a Microsoft Windows product. Contact your vendor to determine if installing an off-the-shelf copy of Microsoft Windows will cause any problems.

**If you feel you have tested all possible situations that may be causing your failed I/O communications, contact your local Wonderware distributor for technical support.**

↪ For more information on obtaining technical support, see your online *FactorySuite System Administrator's Guide*.