

**Wonderware®**

**MBTCP DAServer User's Guide**

**Version C**

**Last Revision: 3/1/07**

**Wonderware**

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.

**© 2003-2004, 2007 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

<b>Before You Begin .....</b>	<b>7</b>
About This Book .....	7
<b>Introduction .....</b>	<b>9</b>
Overview .....	9
Generic Modbus Controllers .....	10
Communications Protocols .....	10
Application Communications Protocols.....	10
Network Communications Protocols.....	12
Modbus TCP/IP Ethernet Protocol .....	12
Accessing Items via the DAServer.....	13
Features .....	14
Demo Mode.....	15
<b>Configuration.....</b>	<b>17</b>
Getting Started Quickly with the DAServer .....	17
Configuring the DAServer .....	19
MBTCP Hierarchy in the DAServer Manager .....	22
TCPIP_PORT Object .....	22
ModbusBridge Object.....	24
TSXQuantum Object.....	25
TSXMomentum Object.....	29
TSXPremium Object.....	33
ModbusPLC Object.....	37
Adding the Compact984, ModiconMicro, TSXMomentumRS, or ModbusPLCRS Objects to the MBTCP Hierarchy.....	41
String-Data Handling.....	57
Full Length Sytle.....	57
C Style.....	58
Pascal Style .....	58
Message Optimization .....	59
Configuring Device Group and Device Item Definitions .....	59
Device Group Definitions.....	59
Device Item Definitions.....	62
Scan-Based Message Handling .....	67
Unsolicited Message Handling.....	68
Archiving Configuration Sets.....	68
Hot Configuration .....	69

**Item Names ..... 71**

Data and Register Types .....	71
Modbus Item Naming .....	72
Register-Number Item Names .....	73
Item Names Using the Modicon PLC Register Addresses.....	74
Absolute Notation Item Names .....	75
Modulo-10000 Point Item Names .....	76
Modulo-10000 Items, BCD Register Type, and Concept Data Structures .....	77
DAServer Standard System Items.....	78
DAServer Global System Item.....	78
DAServer Device-Specific System Items .....	79
DAServer Device-Group-Specific System Items.....	81
Generic OPC Syntax .....	83

**Troubleshooting ..... 85**

Monitoring Connectivity Status with the PLC.....	85
Monitoring the Status of DAS Conversations.....	86
Using DDEStatus and IOStatus in Excel.....	86
Reading Values from the DAServer into Excel.....	87
Writing Values to the DAServer from Excel.....	87
Error Messages and Codes.....	88
DAServer Error Messages.....	89
Server-Specific Error Codes.....	98

**Reference ..... 99**

DAServer Architecture .....	99
DAServers .....	99
Plug-ins .....	100
DAS Engine .....	101
PLC Protocol Layer .....	101
Component Environments.....	101

**Supported DASMBTCP Hardware and  
Firmware ..... 103****The Modbus Protocol ..... 105**

Controller Function Codes.....	105
Modbus Exception Codes .....	106
TCP Port.....	107
Data Types.....	108

**Index ..... 109**



---

# Before You Begin

## About This Book

This book describes how the Wonderware® MBTCP DAServer™ is configured and used, after it has been installed. The book is organized in the following fashion:

- **Contents**
- **Introduction:** contains overview information about the MBTCP DAServer and the environment in which it works.
- **Configuration:** contains a detailed description of the user-interface elements of this DAServer in addition to its functionality.
- **Item Names:** describes the item-naming conventions for targeted devices.
- **Troubleshooting:** provides information about error messages and codes displayed by this MBTCP DAServer.
- **Reference:** describes the DAServer architecture in general.
- **Index**

You can view this document online or you can print it, in part or whole, by using the Adobe Acrobat Reader's print facility. To view this document properly, you must use version 4.0 of the Acrobat Reader.





## CHAPTER 1

# Introduction

This chapter provides you with an overview of the Wonderware® MBTCP DAServer™, including the application- and bus-level communications protocols, item naming conventions, and DAServer features.

## Contents

- Overview
- Generic Modbus Controllers
- Communications Protocols
- Accessing Items via the DAServer
- Features
- Demo Mode

## Overview

The Wonderware® MBTCP DAServer (referred to as the DAServer throughout the remainder of this user's guide) is a Microsoft® Windows® application program that acts as a communications protocol server. It allows other Windows application programs access to data in the (Schneider's) Modicon-family of controllers (also referred to as devices), including the TSX Premium, TSX Quantum, and TSX Momentum that are connected to the DAServer through the computers' Ethernet ports and the Modbus TCP/IP protocol.

The MBTCP DAServer also provides connectivity to the Modicon Micro, Compact 984, and TSX Momentum controllers through a Modbus Bridge (174CEV30010 or 174CEV30020) and/or an NR&D Pen-T Bridge that bridges the communications between the Ethernet and the Serial Modbus protocol.

The MBTCP DAServer also supports Modbus-compatible controllers. A supported Modbus controller can be connected directly through the TCP/IP protocol, or serially through the RS232/RS485 connection via a Modbus Bridge. This supported controller can either be a 4-Digit, 5-Digit, or 6-Digit controller, and it must conform to and comply with the Modbus specifications as listed in Appendix B, The Modbus Protocol.

While the DAServer is primarily intended for use with Wonderware InTouch® (Version 7.11 Patch 02 and later), it may be used by any Microsoft Windows program capable of acting as a DDE, SuiteLink™, or OPC client that can also coexist with FactorySuite™ 2000 and greater.

## Generic Modbus Controllers

Starting in version 1.1 of the MBTCP DAServer, additional Modbus devices that are not listed in the supported hardware will be supported. These Modbus devices, referred to as Generic Modbus devices/controllers in this document and in the implementation of the DAServer, must be capable of supporting the Modbus protocol function codes, exception codes, and data types as described in Appendix B, The Modbus Protocol.

Compared to the Schneider PLCs listed in Appendix A, Supported DASMBTCP Hardware and Firmware, the Generic Modbus devices/controllers offer the following additional capabilities:

- Configurable TCP Port Number.
- Support of Modbus devices that cannot handle multiple-coil write in one message.
- Support of Modbus devices that cannot handle multiple-holding-register write in one message.
- Configurable 4-digit, 5-digit, or 6-digit addressing.
- The maximum addressable register range will be verified by the Modbus devices and does not need to be configured into the DAServer.

## Communications Protocols

The MBTCP DAServer (Data Access Server) communicates with clients and PLCs using the following different communications protocols:

- Application communications protocols such as OPC, DDE, and SuiteLink to communicate with clients located on either local or remote nodes.
- Modbus TPC/IP Ethernet protocol to communicate with the Modicon controllers, either directly or through a Modbus or NR&D Pen-T Bridge and RS232 or RS485 serial cable.

---

**Important!** This DAServer is compliant with the OPC Data Access (DA) 2.05 specifications.

---

For more information about the DAServer architecture, see the Reference section.

## Application Communications Protocols

The DAServer utilizes the following application communications protocols to communicate with the clients.

## OPC

OPC (OLE for Process Control) is a non-proprietary set of standard interfaces based upon Microsoft's OLE/COM technology. This standard makes possible interoperability between automation/control applications, field systems/devices and business/office applications.

Avoiding the traditional requirement of software/application developers to write custom drivers to exchange data with field devices, OPC defines a common, high-performance interface that permits this work to be done once, and then easily reused by HMI, SCADA, control and custom applications.

Over the network, OPC uses DCOM (Distributed COM) for remote communications.

## SuiteLink

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 TCP/IP standard is supported on Windows NT and Windows NT-technology-based operating systems (for example, Windows 2003, Windows 2000, Windows XP, and Windows XP Embedded).

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 provides the following features:

- Value Time Quality (VTQ) places a timestamp 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 operating system's performance monitor. This feature is critical for the operation 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.

## FastDDE

FastDDE provides a means of packing many proprietary Wonderware Dynamic Data Exchange 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.

## DDE

DDE is a communications 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.

### **NetDDE**

NetDDE is a communications protocol that extends the standard DDE functionality to include communications 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-IBM-compatible computers under operating environments such as VMS and UNIX.

## **Network Communications Protocols**

The Modbus Ethernet (MBTCP) DAServer, part of the Schneider Data Access Server Family, is designed to provide connectivity to the family of Modicon controllers through the following supported network communications protocol:

- Modbus TCP/IP Ethernet protocol

---

**Note** The MBTCP DAServer is capable of supporting dual network (NIC) cards in a system.

---

### **Modbus TCP/IP Ethernet Protocol**

The Modbus TCP/IP Ethernet protocol is a part of the MBTCP DAServer, which must be installed on your computer and configured for the PLC with which you wish to communicate.

- This Modbus TCP/IP Ethernet protocol can be used in a network with up to 1024 slave devices.

---

**Note** For more information on the Modbus protocol and to better understand how to read and write data to the different Modicon controllers, please refer to the Modicon "Modbus Protocol Reference Guide" (PI-MBUS-300) from Schneider Electric.

---

- Direct Connectivity

The Modbus TCP/IP Ethernet protocol is utilized to directly connect to the following Modicon controllers through the TCP/IP port.

- TSX Quantum controllers
- TSX Momentum controllers
- TSX Premium controllers
- Generic Modbus TCP (4-Digit, 5-Digit, and 6-Digit) controllers

- Indirect Connectivity

The TCP/IP Ethernet protocol, through the TCP/IP port and down to either a Modbus Bridge (174CEV30010 or 174CEV30020) or NR&D Pen-T Bridge is used to communicate with the following controllers:

- Compact 984 controllers (via RS232)
- Modicon Micro controllers (via RS232)
- TSX Momentum controllers (via RS232 or RS485)
- Generic Modbus Serial (4-Digit, 5-Digit, 6-Digit) controllers (via Serial RS485)

## Accessing Items via the DAServer

The method for accessing items through the DAServer depends on the communications protocol being used.

### OPC

In the case of OPC communications, the protocol addresses an element of data in a conversation with six characteristics: node name, program name, group name, device group, link name, and item name.

- The node name (required for remote access) and device group are optional.
- A fully qualified OPC Item name (ItemID) is composed of the link name and item name.
- All other characteristics are specified through separate DAServer means.

To access an OPC item, the OPC client needs to connect to the DAServer (either in-process or out-of-process) and create an OPC group defining the data-acquisition properties for the collection of items to be added. OPC groups can be either public or private. Public OPC groups are shared across multiple clients, whereas private OPC groups are local to a single client. Optionally, a device group, which indicates the access path to the items for read/write, can be specified from the DAServer.

The following briefly describes each characteristic of the OPC protocol:

- **node name:** Computer (host) name identifying a specific node on the network (for Remote Access ONLY).
- **program name:** The registered OPC server name uniquely identifying a specific server (ProgID). For this DAServer, the program name is **Archestra.DASMBTCP.1**.
- **group name:** The OPC group created from the client for organizing a collection of items logically with the same data acquisition properties between the client and the server, such as update rate.

- **device group:** Meaningful names configured in the DAServer under a specific controller for the common custom attributes between the DAServer and the device, such as update interval. If not specified from the client, the default device group using the global configuration attribute values from the DAServer is assumed. Functionally, a device group is equivalent to an access path (optional).
- **link name:** The set of hierarchy node names, representing the specific devices on a communications path link from the hierarchy root to a specific controller as configured for this DAServer under the DAServer Manager, separated by delimiters.
- **item name:** A specific data element, the leaf of the hierarchy tree of this DAServer, within the specified group. For example, when using this DAServer, an item can be a relay, timer, counter, register, and so on, in the controller.

### DDE/SuiteLink

In the case of DDE/SuiteLink communications, the protocol addresses an element of data in a conversation that uses a four-part naming convention that includes the node name, application name, topic name, and item name. The fully qualified DDE/SuiteLink naming convention includes all four parts, although the node name part (required for remote access only) is optional. The following briefly describes each portion of this naming convention:

- **node name:** Computer (host) name identifying a specific node on the network (for Remote Access ONLY).
- **application name:** The name of the Windows program (this DAServer) that will be accessing the data element. In the case of data coming from or going to the Modicon devices via the DDE/SuiteLink PlugIn of this DAServer, the application name portion of the address is **DASMBTCP**.
- **topic name:** Meaningful names are configured in the DAServer to identify specific devices. These names are then used as the topic names in all conversations with that device. For example, the **ModiconPLC** topic name maps to a device group defined in the DAServer.

---

**Note** You can define multiple device-group (topic) names for the same device (PLC) to poll different points at different rates.

---

- **item name:** A specific data element within the specified topic. For example, when using this DAServer, an item can be a relay, timer, counter, register, and so on, in the PLC.

---

**Note** 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 of this user's guide.

## Features

The Wonderware MBTCP DAServer provides the following features:

- The ability to communicate over multiple application-level protocols at the same time.
- The ability to add new application-level protocols on the fly.
- The ability to be configured remotely.
- New, robust diagnostic abilities.
- Additional server-specific diagnostics.
- XML storage.  
For example, the storage of the .aacfg file that has the details of all the device groups and device items that can be stored in XML.
- Full existing item-name space.
- Log of errors, warnings, traces, and Modbus messages, individually adjustable for reading and writing.
- OPC browsing.

For more in-depth information on the DAServer architecture, see the Reference section.

## Demo Mode

You can install a fully functioning version of this DAServer for demonstration purposes without a license. Demo Mode allows you to test the functionality of the DAServer for 120 minutes. After that time, you must install a license to continue using the DAServer.

When you first start this DAServer, it checks for a license. If the DAServer cannot find a valid license installed on the local computer, it logs a warning message indicating a valid license cannot be retrieved, and enters Demo mode. Thereafter, the DAServer repeats its request for the license every 30 seconds. If no license is found, the DAServer again logs a warning message on the issue.

This process is repeated for 120 minutes, after which the server stops updating read/write on all device items (read from cache is allowed, but all non-system data would receive Bad quality status) and reject the addition of any new items. The DAServer continues to request for a license. Clients continue to function normally (for instance, you can still add or remove an item, but its quality is set to Bad until a license is obtained).

---

**Note** Use the \$\$SY\$\$Licensed system item, a read-only Boolean item, to check the status of your license: True for Licensed or in Demo Mode and False for Not Licensed.

---

If you subsequently add a license to the License Manager, the DAServer logs a message acknowledging the license, switches out of Demo mode, and runs normally.

---

**Note** Once a DAServer obtains a valid license, it no longer checks for a license. However, if your license expires, your DAServer will no longer function when you restart the DAServer.

---





## CHAPTER 2

# Configuration

Once the Wonderware MBTCP DAServer has been installed, a small amount of configuration is required. This configuration is performed using the DAServer Manager hosted in the System Management Console after it is started through the **Programs** menu of the Windows **Start** button.

Before the DAServer is activated, the device hierarchy, simulating the physical hardware layout, must first be built to establish communications to each of the controllers. Once the MBTCP hierarchy has been built, the respective devices for communications can be configured. Finally, the desired Device Groups for each controller may be created.

---

**Note** To run the MBTCP DAServer as a service, use the context menu on the DAServer name and select **Configure As Service**. You can configure it as an auto service or manual service. For more information about configuring your DAServer as a service see the Activation/Deactivation/Service Component of the DAServer Manager documentation.

---

---

**Note** The MBTCP DAServer must be run as a service if DDE/SuiteLink is used.

---

## Contents

- Getting Started Quickly with the DAServer
- Configuring the DAServer
- Configuring Device Group and Device Item Definitions
- Hot Configuration

## Getting Started Quickly with the DAServer

This section briefly describes the procedures required to prepare the MBTCP DAServer for use. Detailed descriptions of each step can be found in later sections of this documentation. This section is intended for people who are familiar with DAServers.

---

**Note** If you are not familiar with DAServer functionality, please proceed to the more-detailed procedures following this section.

---

The following procedures assume that you have:

- Configured the PLC with which you wish to communicate.

### To prepare the MBTCP DAServer

1. Install the Wonderware MBTCP DAServer on Windows by running the **Setup.exe** program.

---

**Note** DAServer installation instructions are included in a separate Help file (.chm extension).

---

- Accept all the default settings during installation.

---

**Important!** Since there are no default values for security settings, you must take note of the User Name and password selected during the install.

---

2. Start the Wonderware DAServer Manager by selecting the **Programs** menu from the **Start** button on the taskbar.
3. Navigate to the **Wonderware** folder that contains the System Management Console, then click **System Management Console**.
4. From the **System Management Console**, find the MBTCP DAServer in the **DAServer Manager** tree.
  - Under the **Local** branch node, the name of the DAServer is **ArchestrA.DASMBTCP.1**.

---

**Note** See the DAServer Manager documentation for general information about working in this snap-in environment.

---

5. The new MBTCP DAServer must now be configured.
  - Before proceeding, determine the hierarchical structure of the network/PLC environment to which you plan to connect.
6. Right-click the **Configuration** object that already exists in the tree, and select **Add TCPIP\_PORT Object** from the shortcut menu.
  - A new TCPIP\_PORT object is created as a node in the hierarchy tree and is named **New\_TCPIP\_PORT\_000** by default.
  - The DAServer allows only one TCPIP\_PORT object in the hierarchy.
  - If you do not rename the object at this time, a numeric sequencing system is applied.
  - Any hierarchy entry can be renamed at a later time.
7. Right-click on the new **TCPIP\_PORT** object, and from the shortcut menu, select one of the following:

**Add ModbusBridge Object**

**Add TSXQuantum Object**

**Add TSXMomentum Object**

**Add TSXPremium Object**

**Add ModbusPLC Object**

- If you add a ModbusBridge object, you must configure the ModbusBridge hierarchy further. Go to Step 8.
8. Right-click on the new **ModbusBridge** object, and from the shortcut menu, select one of the following:
    - Add Compact984 Object**
    - Add ModiconMicro Object**
    - Add TSXMomentumRS Object**
    - Add ModbusPLCRS Object**
  9. Configure the respective device objects, created in the preceding steps, with the appropriate parameter values, if applicable.
    - Optionally, the desired device groups can be created under the **Device Groups** tabbed page with each of the PLC objects.
    - Desired device items can also be optionally created under the **Device Items** tabbed page with each of the PLC objects.

---

**Note** The hierarchy entry is added in the "edit mode," providing a convenient place for you to appropriately describe components of your specific hardware environment. Both hierarchy node name and device group name are numerically sequenced by default. They can be renamed at any time.

---

The DAServer is now ready for use. In order to use the DAServer, you must activate it from the **DAServer Manager** using either the shortcut menu's **Activate Server** command from the **ArchestrA.DASMBTCP.1** node, or an OPC Client.

---

**Note** To run the MBTCP DAServer as a service, right-click on the **DAServer name** and select **Configure As Service** from the shortcut menu. You can configure it as an auto service or manual service. For more information about configuring your DAServer as a service, see the Activation/Deactivation/Service Component of the DAServer Manager documentation.

---

## Configuring the DAServer

---

**Note** This DAServer is hosted by the DAServer Manager, a Microsoft Management Console (MMC) snap-in, which is a part of the ArchestrA System Management Console (SMC) suite of utilities. Many high-level functions and user-interface elements of the DAServer Manager are universal to all DAServers, and **only** the documentation for the DAServer Manager contains descriptions of those universal functions/UI elements. Therefore, reading the documentation for both the MMC and the DAServer Manager is critical to understanding this user's guide. To read the documentation about the MMC and DAServer Manager, right-click the **DAServer Manager** icon and select the **Help** menu. Both the MMC Help and the DAServer Manager Help are displayed. An Adobe Acrobat version of the DAServer Manager documentation (DAServerManager.pdf) is also available in the CD-ROM directory\User Docs\English.

---

---

**Note** The shortcut menu items described in this document typically represent only a subset of any actual shortcut menu. Most items in each shortcut menu are standard Windows commands. For more information about those commands, please see **Help**, by right-clicking the **System Management Console** icon.

---

**Note** For more information on the Modbus protocol and to better understand how to read and write data to the different Modicon controllers, please refer to the Modicon Modbus Protocol Reference Guide (PI-MBUS-300) from Schneider Electric.

---

### To prepare the MBTCP DAServer

1. Install the Wonderware MBTCP DAServer on Windows by running the **Setup.exe** program.

---

**Note** DAServer installation instructions are included in a separate Help file (.chm extension).

---

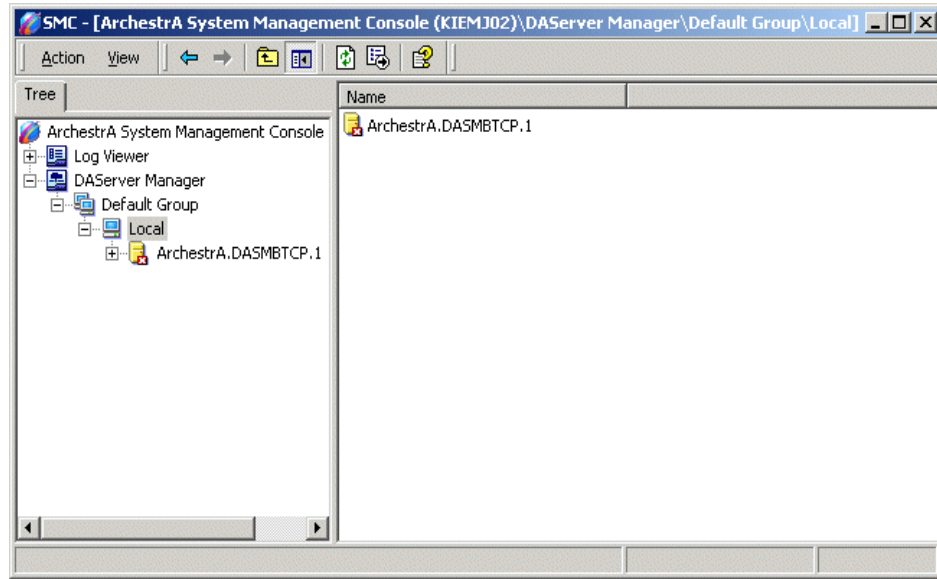
2. Accept all the default settings during the installation.

---

**Important!** Since there are no default values for security settings, you must take note of the User Name and password selected during the install.

---

3. After the DAServer has been installed, start the System Manager Console by clicking the **Start** button on the Windows taskbar and pointing to **Programs**.
4. Point to the **Wonderware** folder that contains the System Management Console, then click **System Management Console**.
5. From the System Management Console tree, click on **DAServer Manager**.
6. Click on **Default Group**, then the **Local** node.
  - Under the Local node, the DAServer name is **ArchestrA.DASMBTCP.1**.




---

**Note** See the DAServer Manager documentation for general information about working in this snap-in environment.

---

7. Before the DAServer is started, you must first build the device hierarchy to establish communications to each of the controllers.

---

**Important!** For step-by-step procedures on how to build the device hierarchy, please see the following section, "MBTCP Hierarchy in the DAServer Manager."

---

**Note** Selecting the **Configuration** object of the hierarchy tree displays the **Global Parameters** configuration view for this DAServer. The default Poke Mode settings for the DAServer is Optimization mode. Configure all other global parameters as required for this DAServer. For more information about the **Global Parameters** configuration view, including descriptions of the different Poke Modes, see the DAServer Manager documentation. You can access that documentation by right-clicking the **DAServer Manager** icon and selecting the **Help** menu, and then navigating through the **DAServer Manager** book.

---

**Important!** Any Global Parameters that appear dimmed are either not supported or cannot be configured for this DAServer. Simulation Mode is not supported.

---

8. When the MBTCP hierarchy build has been completed, you can start configuring the respective devices for communications.
9. You may create the desired Device Groups for each controller by:
  - Navigating to the object of interest in the **DAServer Manager** tree view.
  - Clicking on the **Device Groups** tab.

- Right-clicking in the **Device Groups** dialog box and selecting the **Add** command from the shortcut menu.

---

**Important!** For step-by-step procedures on configuring Device Groups, please see the following section, "Device Group Definitions."

---

10. Finally, you may create the desired Device Items for each controller by:

- Navigating to the object of interest in the **DAServer Manager** tree view.
- Clicking on the **Device Items** tab.
- Right-clicking in the **Device Items** dialog box and selecting the **Add** command from the shortcut menu.

---

**Important!** For step-by-step procedures on configuring Device Items, please see the following section, "Device Item Definitions."

---

**Note** When any configuration view is in an open state and you open the same server the second time, the DAServer locks the second instance of this same-server access for any update or configuration activities. Access to this second opening instance will resume after the first one has been closed.

---

The DAServer is now ready for use. In order to use the DAServer, you must activate it.

- If you are using an OPC Client, the DAServer can be auto-started.
- If you are using DDE/SuiteLink, you must start the DAServer either as a manual or automatic service.
- To activate the DAServer, right-click on **ArchestrA.DASMBTCP.1** and select **Activate Server** from the shortcut menu.

---

**Note** To run the MBTCP DAServer as a service, right-click on the **DAServer name** and select **Configure As Service** from the shortcut menu. You can configure it as an auto service or manual service. For more information about configuring your DAServer as a service, see the Activation/Deactivation/Service Component of the DAServer Manager documentation.

---

## MBTCP Hierarchy in the DAServer Manager

Before attempting to configure your DAServer, you should determine the hierarchical structure of your network/PLC environment.

### TCPIP\_PORT Object

The server-specific configuration portion of the MBTCP DAServer hierarchy tree under the DAServer Manager starts at the TCPIP\_PORT object.

1. Configure the TCPIP\_PORT object from the **Configuration** branch of the hierarchy after the DAServer has been installed.
2. Rename this object as appropriate.

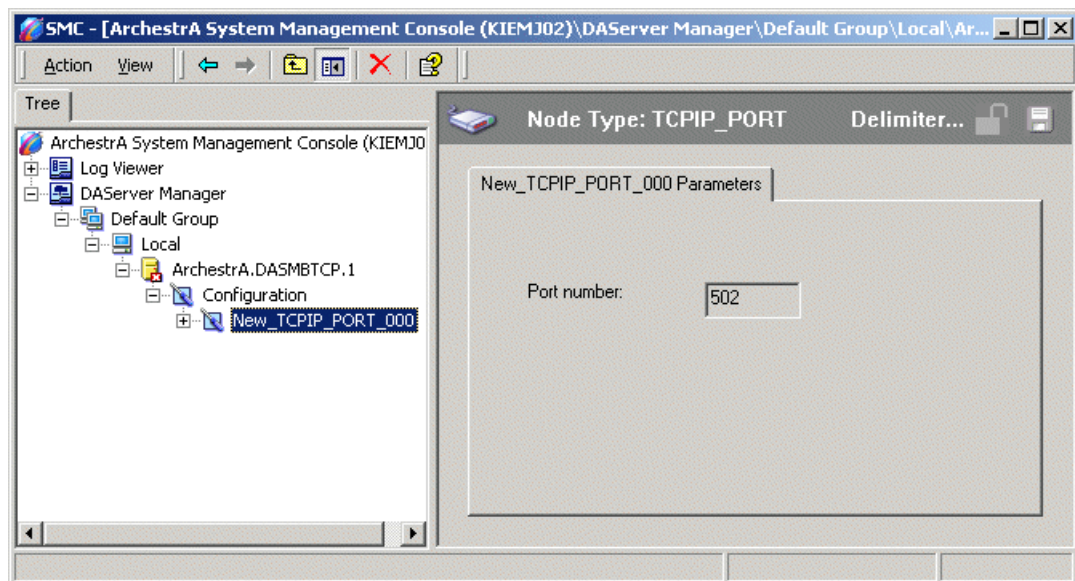
---

**Important!** If you subsequently clear your configuration hierarchy, you must create this TCPIP\_PORT object from the **Configuration** branch of the hierarchy. From this point, all of the following instructions apply.

---

### To create a TCPIP\_PORT object from the Configuration branch

1. Right-click on **Configuration**.
2. Select **Add TCPIP\_PORT Object** from the shortcut menu.
  - A new TCPIP\_PORT object is created as a node in the hierarchy tree.
  - It is named **New\_TCPIP\_PORT\_000** by default.
3. Rename the newly created object as appropriate.
  - The **New\_TCPIP\_PORT\_000 Parameters** configuration view (right pane) is displayed.



This configuration view has one element:

- **Port number:** Displays the default port (socket) number, which is 502.

---

**Note** The MBTCP DAServer uses port 502 as the default port number to contact the PLC. The port number in this display is dimmed and non changeable. However, the actual port to be used by the Generic Modbus PLCs directly under the TCPIP\_PORT object is available at that node and is configurable using the ModbusPLC Object.

---

From the New\_PORT\_TCPIP\_000 branch of the DAServer hierarchy, the following objects can be created:

- ModbusBridge Object
- TSXQuantum Object (representing the TSXQuantum controller)
- TSXMomentum Object (representing the TSXMomentum controller)
- TSXPremium Object (representing the TSXPremium controller)

- ModbusPLC Object (representing either the Modbus Generic 4-Digit, 5-Digit, or 6-Digit controller)

---

**Important!** Each hardware configured has a limitation to the number of connections it can support at one time. Please refer to the respective hardware's documentation for the maximum number of simultaneous Modbus/TCP server connections it can support.

---

**Note** The TSXQuantum, TSXMomentum, TSXPremium, and ModbusPLC objects represent the logical endpoint to the hardware hierarchy. If you add a ModbusBridge object, you must configure an additional leaf on the hierarchy.

---

## ModbusBridge Object

### To add ModbusBridge objects to your MBTCP hierarchy

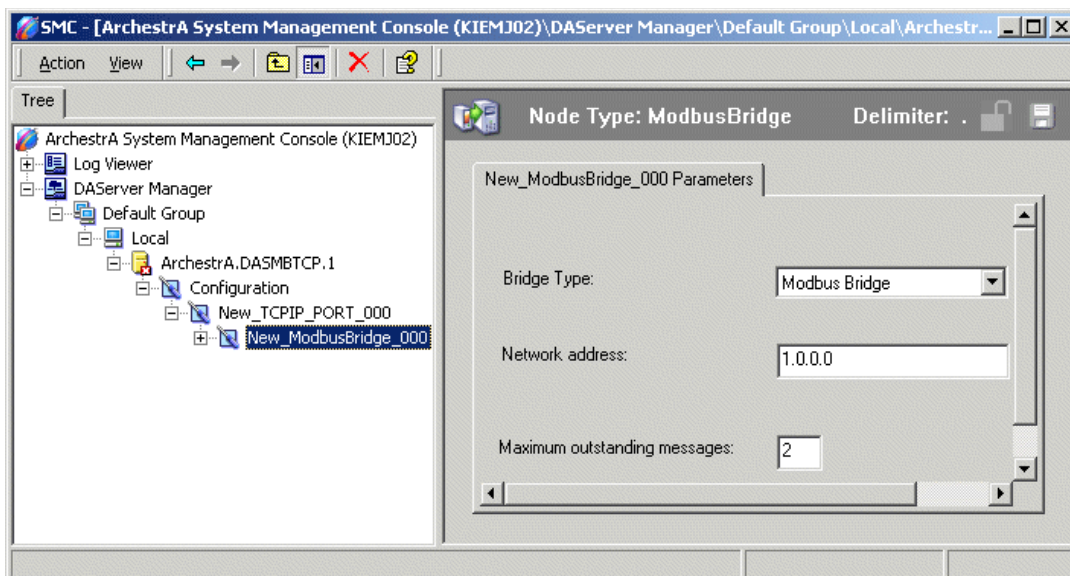
1. Right-click on the **TCPIP\_PORT** branch.
2. Select **Add ModbusBridge Object** from the shortcut menu.
  - It is named **New\_ModbusBridge\_000** by default.

---

**Note** You can add up to 247 of each type object to the hierarchy. However, the bridge itself limits the number of PLCs that can be connected to the serial line.

---

3. Rename as appropriate.
  - The **New\_ModbusBridge\_000 Parameters** configuration view is displayed.



This configuration view has three configurable elements.

- **Bridge type:** From the drop-down menu, select the type of communications bridge to use for the connection to the TCP/IP Port.



- The default bridge type is the Modbus Bridge.
- The alternative bridge is the NR&D Pen-T Bridge.
- **Network address:** Enter the host name or IP address of the bridge.
  - The maximum number of characters is 255.
  - The minimum number of characters is 1 (one).
  - The first and last characters must be alphanumeric, and the remaining characters must be alphanumeric, a "-" (hyphen) or a "." (period). No blank spaces are allowed.
  - The default value is 1.0.0.0.
- **Maximum outstanding messages:** Enter the maximum number of queued messages allowed in the Modbus Bridge.

---

**Note** The Bridge Type used governs the value configured.

---

Modbus Bridge:

- The Bridge Type Modbus is the default bridge.
- This attribute is editable.
- The maximum number is 4 (four).
- The minimum number is 1 (one).
- The default value is 2 (two).

NR&D Pen-T Bridge:

- The value of this attribute is set to 1 (one).
- This attribute is not editable.

From the **ModbusBridge** branch of the DAServer hierarchy, the following objects can be created:

- Compact984 Object
- ModiconMicro Object
- TSXMomentumRS Object
- ModbusPLCRS Object

For further instructions on adding these objects, see "Adding the Compact984, ModiconMicro, TSXMomentumRS, or ModbusPLCRS Objects to the MBTCP Hierarchy."

## TSXQuantum Object

### To add TSXQuantum objects to your MBTCP hierarchy

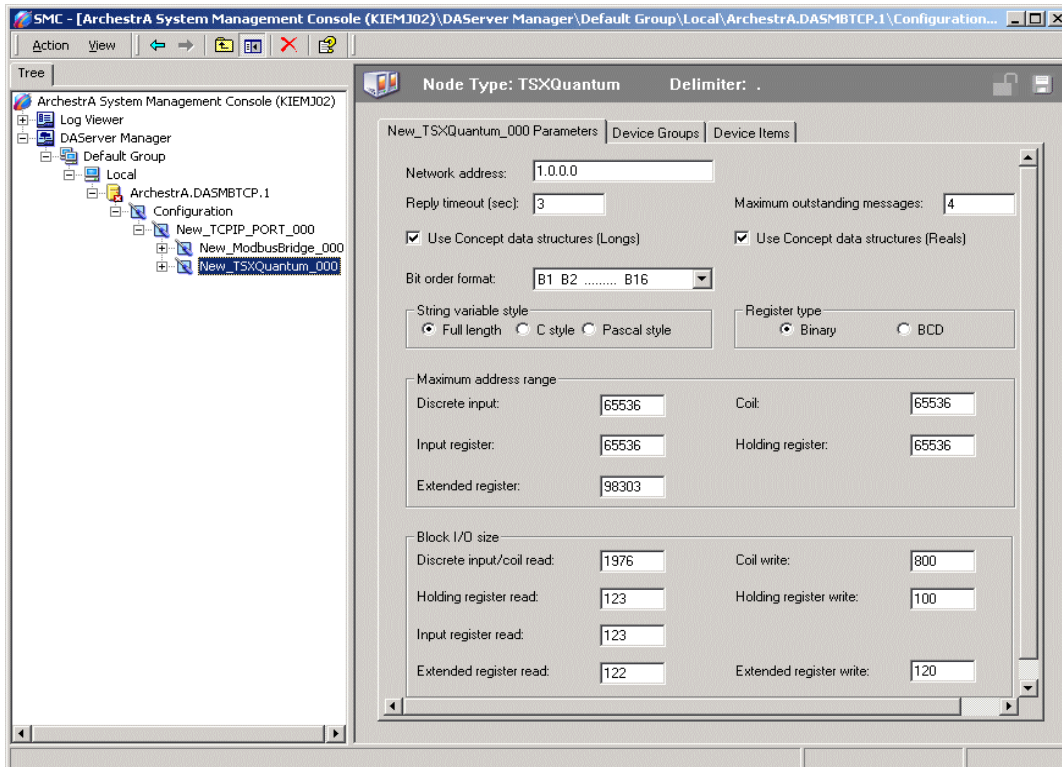
1. Right-click on the **TCPIP\_PORT** branch.
2. Select **Add TSXQuantum Object** from the shortcut menu.
  - A new TSXQuantum object is created as a node in the hierarchy tree.
  - It is named **New\_TSXQuantum\_000** by default.

---

**Note** You can add up to 1024 of this type of object to the hierarchy.

---

3. Rename as appropriate.
  - The **New\_TSXQuantum\_000 Parameters** configuration view is displayed.



This configuration view has 10 elements that are configurable:

- **Network address:** Enter the host name or the IP address of the PLC.
  - The maximum number of characters is 255.
  - The minimum number of characters is 1 (one).
  - The first and last characters must be alphanumeric, and the remaining characters must be alphanumeric, a "-" (hyphen), or a "." (period). No blank spaces are allowed.
  - The default value is 1.0.0.0.
- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).
  - The maximum value is 60.
  - The default value is 3 (three).
- **Maximum outstanding messages:** Enter the maximum number of outstanding messages in the queue.

- The minimum value is 1 (one).
- The maximum value is 20.
- The default value is 4 (four).
- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.
- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length (space padded)
  - C style (null terminated)
  - Pascal style (includes length specifier)
  - The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD
  - The default register type is Binary.
- **Maximum address range:** There are five sub-elements in this Maximum addressable registers box. The maximum addressable registers can be obtained from the Modicon Concept or Modsoft configuration programs. The PLC will return an error if a register within the configured range is used to read data but does not exist in the PLC. The MBTCP DAServer filters out registers outside of this range and logs error messages.

- **Discrete input:** Enter the maximum number of addressable discrete inputs (read coils) in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Coil:** Enter the maximum number of addressable write coils in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Input register:** Enter the maximum number of addressable input registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Holding register:** Enter the maximum number of addressable holding registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Extended register:** Enter the maximum number of addressable extended registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 98303.
  - The default value is 98303.
- **Block I/O size:** The Block I/O Sizes box contains seven sub-elements. The DAServer uses the Block I/O sizes to maximize data throughput. The MBTCP DAServer uses a 256-byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.
  - **Discrete input/coil read:** Enter the maximum number of discrete inputs or coils to read at one time.
    - The minimum value is 1 (one).
    - The maximum value is 1976.
    - The default value is 1976.
  - **Coil write:** Enter the maximum number of coils to write at one time.
    - The minimum value is 1 (one).
    - The maximum value is 800.
    - The default value is 800.

- **Holding register read:** Enter the maximum number of holding registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.
- **Holding register write:** Enter the maximum number of holding registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 100.
  - The default value is 100.
- **Input register read:** Enter the maximum number of input registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.
- **Extended register read:** Enter the maximum number of extended registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 122.
  - The default value is 122.
- **Extended register write:** Enter the maximum number of extended registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 120.
  - The default value is 120.

## TSXMomentum Object

From the TCPIP\_PORT branch of the DAServer hierarchy, you can also create a TSXMomentum object.

### To add TSXMomentum objects to your MBTCP hierarchy

1. Right-click on your TCPIP\_PORT branch.
2. Select **Add TSXMomentum Object**.
  - A new TSXMomentum object is created as a node in the hierarchy tree.
  - It is named **New\_TSXMomentum\_000** by default.

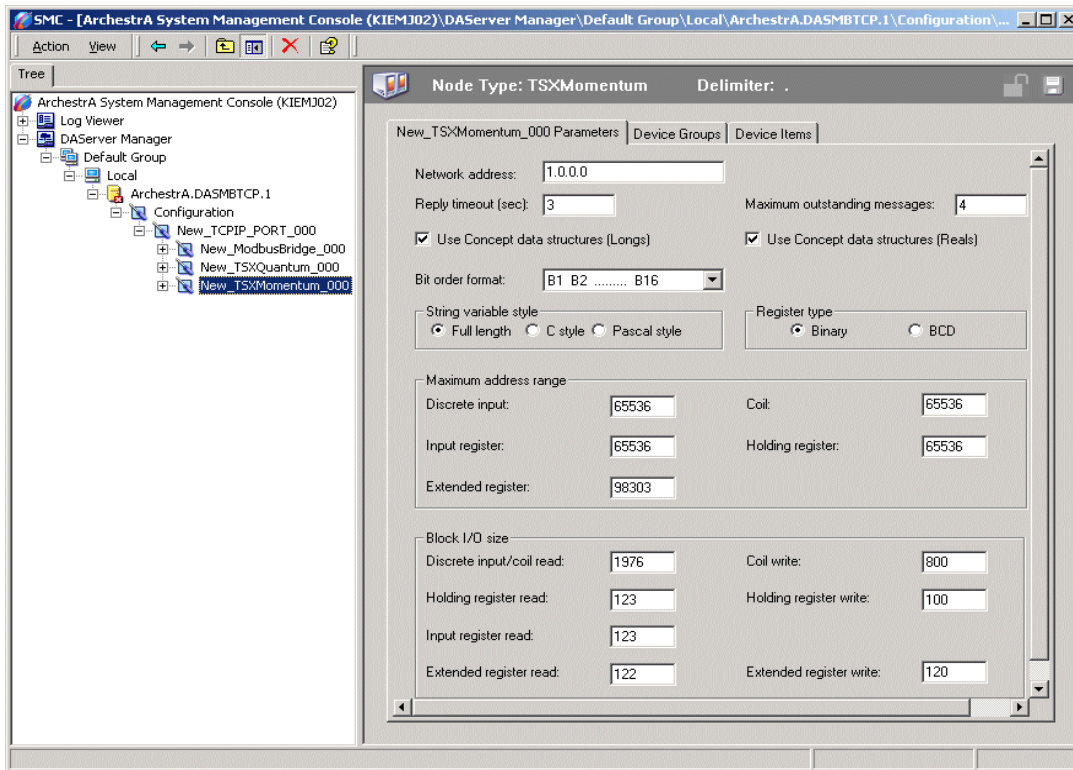
---

**Note** You can add up to 1024 of this type of object to the hierarchy.

---

3. Rename as appropriate.

- The **New\_TSXMomentum\_000 Parameters** configuration view is displayed.



This configuration view has 10 elements that are configurable.

- **Network address:** Enter the host name or IP address of the PLC.
  - The maximum number of characters is 255.
  - The minimum number of characters is 1 (one).
  - The first and last characters must be alphanumeric, and the remaining characters must be alphanumeric, a "-" (hyphen), or a "." (period). No blank spaces are allowed.
  - The default value is 1.0.0.0.
- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).
  - The maximum value is 60.
  - The default value is 3 (three).
- **Maximum outstanding messages:** Enter the maximum number of outstanding messages that can be in the queue for the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 20.
  - The default value is 4 (four).

- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.
- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length (space padded)
  - C style (null terminated)
  - Pascal style (includes length specifier)
  - The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD
  - The default register type is Binary.
- **Maximum address range:** There are five sub-elements in this Maximum addressable registers box. The maximum addressable registers can be obtained from the Modicon Concept or Modsoft configuration programs. The PLC will return an error if a register outside of this range is used to read data. The MBTCP DAServer filters out registers outside of this range and logs error messages.
  - **Discrete input:** Enter the maximum number of addressable discrete inputs/read coils in the PLC.
    - The minimum value is 1 (one).

- The maximum value is 65536.
- The default value is 65536.
- **Coil:** Enter the maximum number of addressable write coils in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Input register:** Enter the maximum number of addressable input registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Holding register:** Enter the maximum number of addressable holding registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Extended register:** Enter the maximum number of addressable extended registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 98303.
  - The default value is 98303.
- **Block I/O size:** This Block I/O Sizes box contains seven sub-elements. The DAServer uses the block I/O sizes to maximize data throughput. The MBTCP DAServer uses a 256-byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.
  - **Discrete input/coil read:** Enter the maximum number of discrete inputs or coils to read at one time.
    - The minimum value is 1 (one).
    - The maximum value is 1976.
    - The default value is 1976.
  - **Coil write:** Enter the maximum number of coils to write at one time.
    - The minimum value is 1 (one).
    - The maximum value is 800.
    - The default value is 800.
  - **Holding register read:** Enter the maximum number of holding registers to read at one time.
    - The minimum value is 1 (one).



- The maximum value is 123.
- The default value is 123.
- **Holding register write:** Enter the maximum number of holding registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 100.
  - The default value is 100.
- **Input register read:** Enter the maximum number of input registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.
- **Extended register read:** Enter the maximum number of extended registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 122.
  - The default value is 122.
- **Extended register write:** Enter the maximum number of extended registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 120.
  - The default value is 120.

## TSXPremium Object

From the TCPIP\_PORT branch of the DAServer hierarchy, you can also create a TSXPremium object.

### To add TSXPremium objects to your MBTCP hierarchy

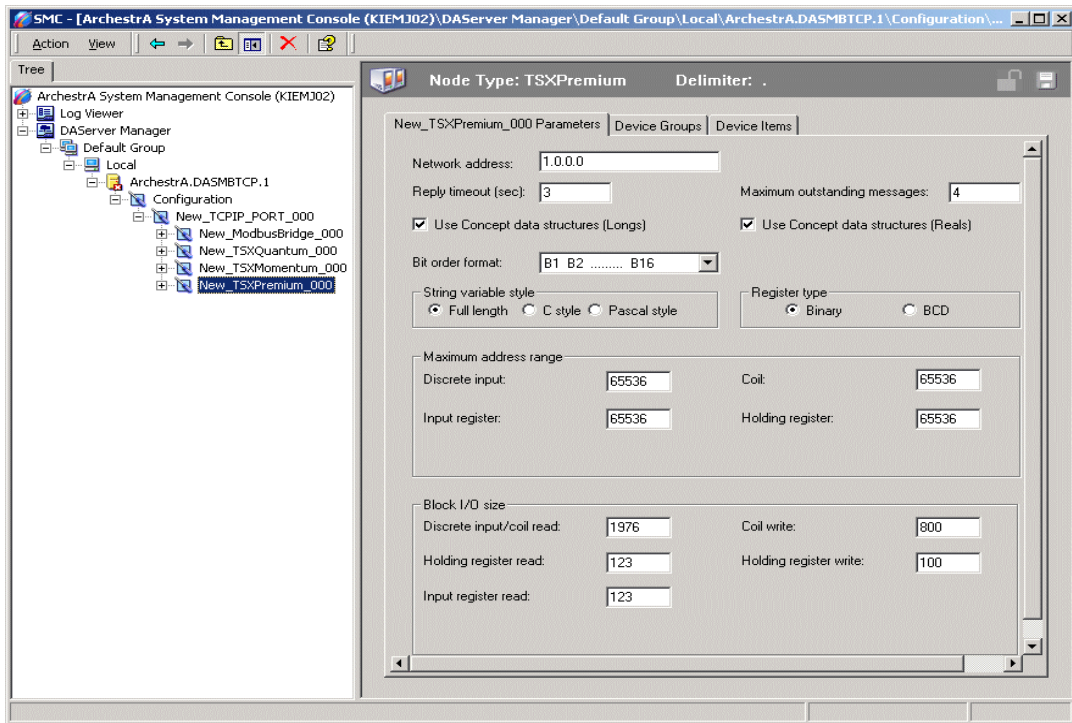
1. Right-click on your **TCPIP\_PORT** branch.
2. Select **Add TSXPremium Object**.
  - A new TSXPremium object is created as a node in the hierarchy tree.
  - It is named **New\_TSXPremium\_000** by default.

---

**Note** You can add up to 1024 of this type of object to the hierarchy.

---

3. Rename as appropriate.
  - The **New\_TSXPremium\_000 Parameters** configuration view is displayed.



This configuration view has 10 elements that are configurable.

- **Network address:** Enter the host name or IP address of the PLC.
  - The maximum number of characters is 255.
  - The minimum number of characters is 1 (one).
  - The first and last characters must be alphanumeric, and the remaining characters must be alphanumeric, a "-" (hyphen), or a "." (period). No blank spaces are allowed.
  - The default value is 1.0.0.0.
- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).
  - The maximum value is 60.
  - The default value is 3 (three).
- **Maximum outstanding messages:** Enter the maximum number of outstanding messages in the queue for the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 20.
  - The default value is 4 (four).
- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.

- Checked — selected
- Not checked — not selected
- The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.
- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length (space padded)
  - C style (null terminated)
  - Pascal style (includes length specifier)
  - The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD
  - The default register type is Binary.
- **Maximum address range:** There are four sub-elements in this Maximum addressable registers box. The maximum addressable registers can be obtained from the Modicon Concept or Modsoft configuration programs. The PLC will return an error if a register outside of this range is used to read data. The MBTCP DAServer filters out registers outside of this range and logs error messages.
  - **Discrete input:** Enter the maximum number of addressable discrete inputs or read coils in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 65536.
    - The default value is 65536.

- **Coil:** Enter the maximum number of addressable write coils in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Input register:** Enter the maximum number of addressable input registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
  - The default value is 65536.
- **Holding register:** Enter the maximum number of addressable holding registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Block I/O size:** The Block I/O Sizes box contains five sub-elements. The DAServer uses the block I/O sizes to maximize data throughput. The MBTCP DAServer uses a 256-byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.
  - **Discrete input/coil read:** Enter the maximum number of discrete inputs or coils to read at one time.
    - The minimum value is 1 (one).
    - The maximum value is 1000.
    - The default value is 1000.
  - **Coil write:** Enter the maximum number of coils to write at one time.
    - The minimum value is 1 (one).
    - The maximum value is 800.
    - The default value is 800.
  - **Holding register read:** Enter the maximum number of holding registers to read at one time.
    - The minimum value is 1 (one).
    - The maximum value is 123.
    - The default value is 123.
  - **Holding register write:** Enter the maximum number of holding registers to write at one time.
    - The minimum value is 1 (one).
    - The maximum value is 100.

- The default value is 100.
- **Input register read:** Enter the maximum number of input registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.

## ModbusPLC Object

The ModbusPLC object is created from the TCPIP\_PORT branch of the DAServer hierarchy. It is intended for PLCs/controllers that use the Modbus protocol but not in the list of the supported hardware described in Appendix A, Supported DASMBTCP Hardware and Firmware. However, the PLCs/controllers need to conform to and comply with the Modbus specifications as listed in Appendix B, The Modbus Protocol.

### To add ModbusPLC objects to your MBTCP hierarchy

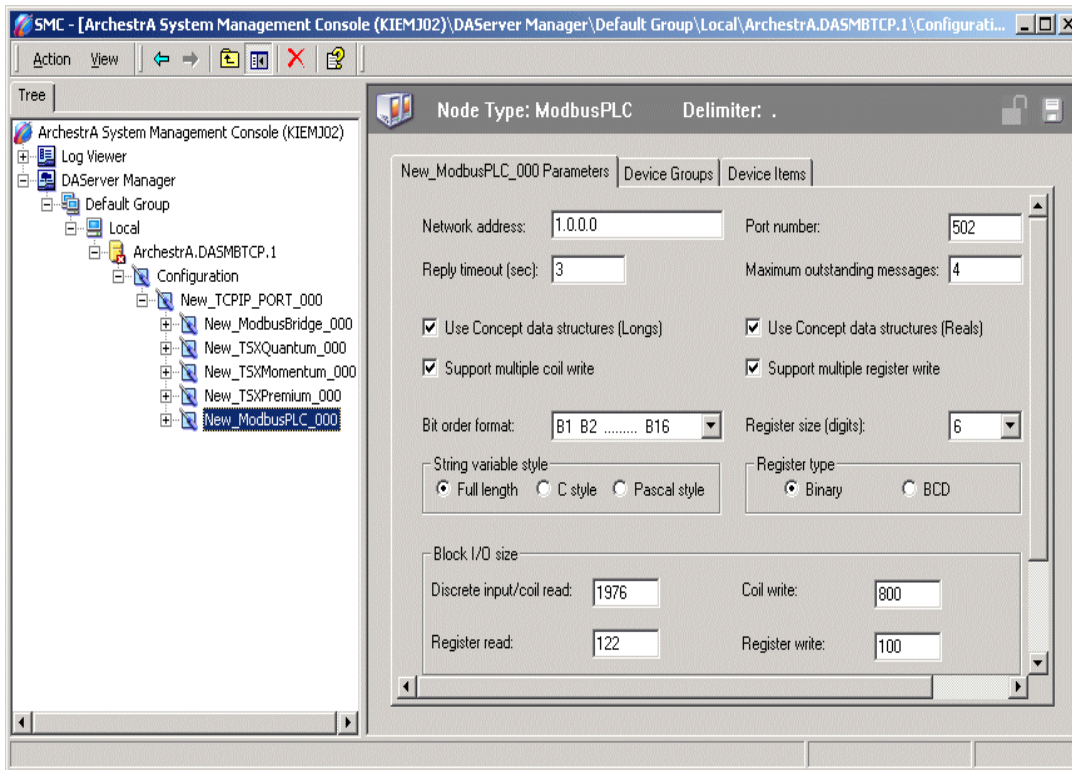
1. Right-click on your **TCPIP\_PORT** branch.
2. Select **Add ModbusPLC Object**.
  - A new ModbusPLC object is created as a node in the hierarchy tree.
  - It is named **New\_ModbusPLC\_000** by default.

---

**Note** You can add up to 1024 of this type of object to the hierarchy.

---

3. Rename as appropriate.
  - The **New\_ModbusPLC\_000 Parameters** configuration view is displayed.



This configuration view has 13 elements that are configurable.

- **Network address:** Enter the host name or IP address of the PLC.
  - The maximum number of characters is 255.
  - The minimum number of characters is 1 (one).
  - The first and last characters must be alphanumeric, and the remaining characters must be alphanumeric, a "-" (hyphen) or a "." (period). No blank spaces are allowed.
  - The default value is 1.0.0.0.
- **Port number:** Enter the port (socket) number.
  - The default port number is 502.

---

**Note** The MBTCP DA Server uses port 502 as the default port number to contact the PLC. The port number in this display is dimmed and non changeable. However, the actual port to be used by the Generic Modbus PLCs directly under the TCPIP\_PORT object is available at that node and is configurable using the ModbusPLC Object. Controllers configured under the ModbusBridge object will always use port number 502.

---

- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).
  - The maximum value is 60.

- The default value is 3 (three).
- **Maximum outstanding messages:** Enter the maximum number of outstanding messages in the queue for the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 20.
  - The default value is 4 (four).
- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Support multiple coil write:** Select for the PLC to write to multiple coils in one message with the Modbus protocol function code 15 (0x0F). If not selected, the PLC will write to a single coil in one message with the Modbus protocol function code 5 (0x05).
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Support multiple register write:** Select for the PLC to write to multiple registers in one message with the Modbus protocol function code 16 (0x10). If not selected, the PLC will write to a single register in one message with the Modbus protocol function code 6 (0x06).
  - Checked — selected
  - Not checked — not selected
  - The default is checked.

---

**Note** Certain things have to be understood when the Support Multiple Register Write Parameter is not selected in the Generic PLC configuration. When this parameter is not selected, it implies that the PLC does not support multiple register writes and the server will only write single registers to the PLC.

What this implies is that items that contain more than one register cannot be written either. For example, items such as 4xxxxx L, 4xxxxx I, 4xxxxx U, 4xxxxx F, 4xxxxx-4xxxxx M, 5 HRL, 5 HRF, 5 PV, 5 HRU, and 4xxxxx-4xxxxx cannot be written. When you try to write to the PLC with this parameter not selected, the following error message will be logged to the logger, "Cannot write to multiple register item: 4xxxxx L on Node: TCPPort.GenPLC. The PLC configurable parameter Support Multiple Register Write is not checked."

---

- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.
- **Register size (digits):** Select the correct register size for addressing the PLC.
  - 4-digit is used for addressing the Modbus Generic 4-Digit PLCs.
  - 5-digit applies to the Modbus Generic 5-Digit PLCs.
  - 6-digit is used for addressing the Modbus Generic 6-Digit PLCs.
  - The default value is 6, for the Modbus Generic 6-Digit PLCs.

---

**Note** The selection for the Register size determines the maximum address range. They are changeable as in other supported PLCs listed in Appendix A, Supported DASMBTCP Hardware and Firmware.  
 For 4-digit, the maximum value is 999.  
 For 5-digit, the maximum value is 9999.  
 For 6-digit, the maximum value is 65536.

---

- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length ( space padded)
  - C style (null terminated)
  - Pascal style (includes length specifier)
  - The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD



- The default register type is Binary.
- **Block I/O size:** This Block I/O Sizes box contains four sub-elements. The DAServer uses the block I/O sizes to maximize data throughput. The MBTCP DAServer uses a 256 byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.
- **Discrete input/coil read:** Enter the maximum number of coils to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 1976.
  - The default value is 1976.
- **Coil write:** Enter the maximum number of coils to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 800.
  - The default value is 800.
- **Register read:** Enter the maximum number of extended registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 122.
  - The default value is 122.
- **Register write:** Enter the maximum number of holding registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 100.
  - The default value is 100.

## Adding the Compact984, ModiconMicro, TSXMomentumRS, or ModbusPLCRS Objects to the MBTCP Hierarchy

The Compact984, ModiconMicro, TSXMomentumRS, and ModbusPLCRS objects can be added from the **New\_ModbusBridge\_000** branch of the DAServer Manager hierarchy.

### Compact984 Object

The Compact984 object can be created only from the ModbusBridge branch.

#### To add Compact984 objects to your MBTCP hierarchy

1. Right-click on your **ModbusBridge** branch.
2. Select **Add Compact984 Object** from the shortcut menu.
  - A new Compact984 object is created as a node in the hierarchy tree.

- It is named **New\_Compact984\_000** by default.

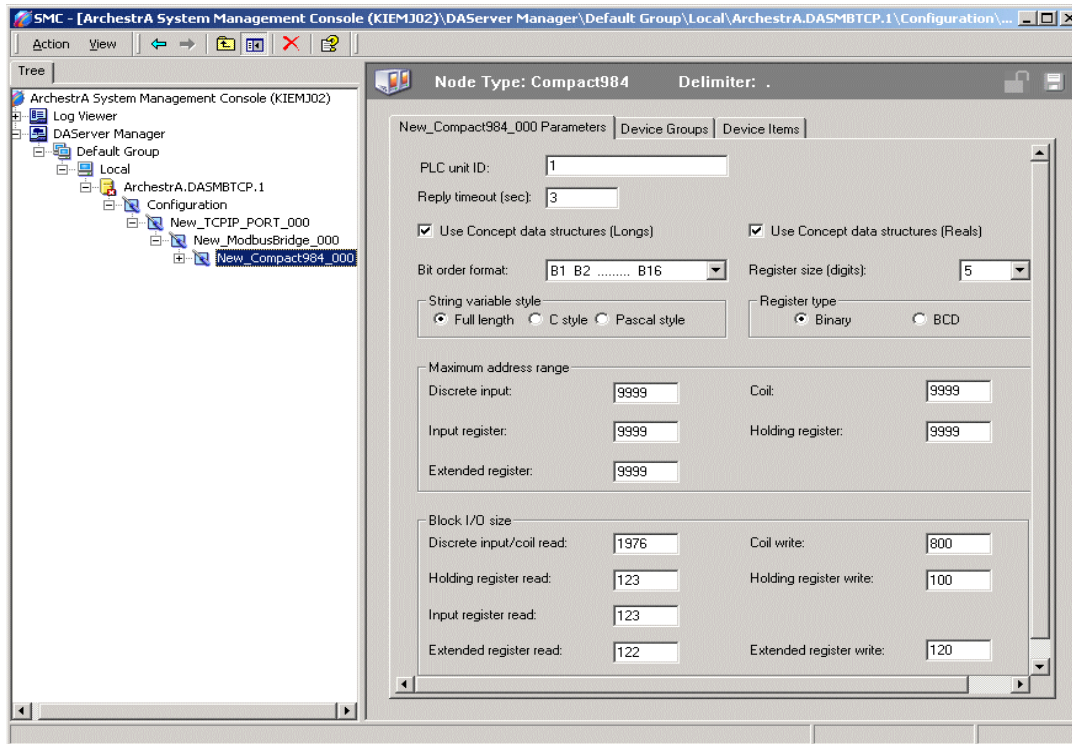
---

**Note** You can add up to 247 of this type of object to the hierarchy.

---

3. Rename as appropriate.

- The **New\_Compact984\_000 Parameters** configuration view is displayed.



This configuration view has 10 configurable elements.

- **PLC unit ID:** Enter the PLC unit ID.
 

The Bridge's internal configuration contains a UnitID parameter which can be set to override the Unit\_ID address received in the message from the server. In other words, when the Unit\_ID box is 0 (zero) the bridge routes the message to its configured Slave device. If the server's Unit\_ID is set to 0 (zero), the message will be delivered to the Slave device whose address is defined in the UnitID box of the Bridge. If the server's Unit\_ID is set to a non-zero value (range 1...255), the message will be delivered to the Slave device at that numerical address, regardless of the contents of the UnitID box in the Bridge.

  - The minimum value is 0 (zero).
  - The maximum value is 255.
  - The default value is 1 (one).
- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).

- The maximum value is 60.
- The default value is 3 (three).
- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.
- **Register size (digits):** Select the correct register size for addressing the PLC.
  - 5-digit register size (applies to 984-145 Compact PLCs).
  - 6-digit register size (applies to 984-265 Compact PLCs).
  - The default value is 5, for the 984-145 Compact PLCs.
- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length (space padded)
  - C style (null terminated)
  - Pascal style (includes length specifier)
  - The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD
  - The default register type is Binary.

- **Maximum address range:** There are five sub-elements in this Maximum addressable registers box. The maximum addressable registers can be obtained from the Modicon Concept or Modsoft configuration programs. The PLC will return an error if a register outside of this range is used to read data. The MBTCP DAServer filters out registers outside of this range and logs error messages.
  - **Discrete input:** Enter the maximum number of addressable discrete inputs/read coils in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999 for 984-145 Compact PLCs and 65536 for 984-265 Compact PLCs.
    - The default value is 9999.
  - **Coil:** Enter the maximum number of addressable write coils in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999 for 984-145 Compact PLCs and 65536 for 984-265 Compact PLCs.
    - The default value is 9999.
  - **Input register:** Enter the maximum number of addressable input registers in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999 for 984-145 Compact PLCs and 65536 for 984-265 Compact PLCs.
    - The default value is 9999.
  - **Holding register:** Enter the maximum number of addressable holding registers in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999 for 984-145 Compact PLCs and 65536 for 984-265 Compact PLCs.
    - The default value is 9999.
  - **Extended register:** Enter the maximum number of addressable extended registers in the PLC.

---

**Note** This option is not available if you set **Register size** to 6.

---

- The minimum value is 1 (one).
- The maximum value is 9999 for 984-145 Compact PLCs and 65536 for 984-265 Compact PLCs.
- The default value is 9999.
- **Block I/O size:** This Block I/O Sizes box contains seven sub-elements. The DAServer uses the block I/O sizes to maximize data throughput. The MBTCP DAServer uses a 256-byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.

- **Discrete input/coil read:** Enter the maximum number of discrete inputs or coils to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 1976.
  - The default value is 1976.
- **Coil write:** Enter the maximum number of coils to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 800.
  - The default value is 800.
- **Holding register read:** Enter the maximum number of holding registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.
- **Holding register write:** Enter the maximum number of holding registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 100.
  - The default value is 100.
- **Input register read:** Enter the maximum number of input registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.
- **Extended register read:** Enter the maximum number of extended registers to read at one time.

---

**Note** This option is unavailable if you set **Register size** to 6.

---

- The minimum value is 1 (one).
- The maximum value is 122.
- The default value is 122.
- **Extended register write:** Enter the maximum number of extended registers to write at one time.

---

**Note** This option is unavailable if you set **Register size** to 6.

---

- The minimum value is 1 (one).
- The maximum value is 120.
- The default value is 120.

## ModiconMicro Object

From the ModbusBridge branch of the DAServer hierarchy, the ModiconMicro object can be created.

### To add ModiconMicro objects to your MBTCP hierarchy

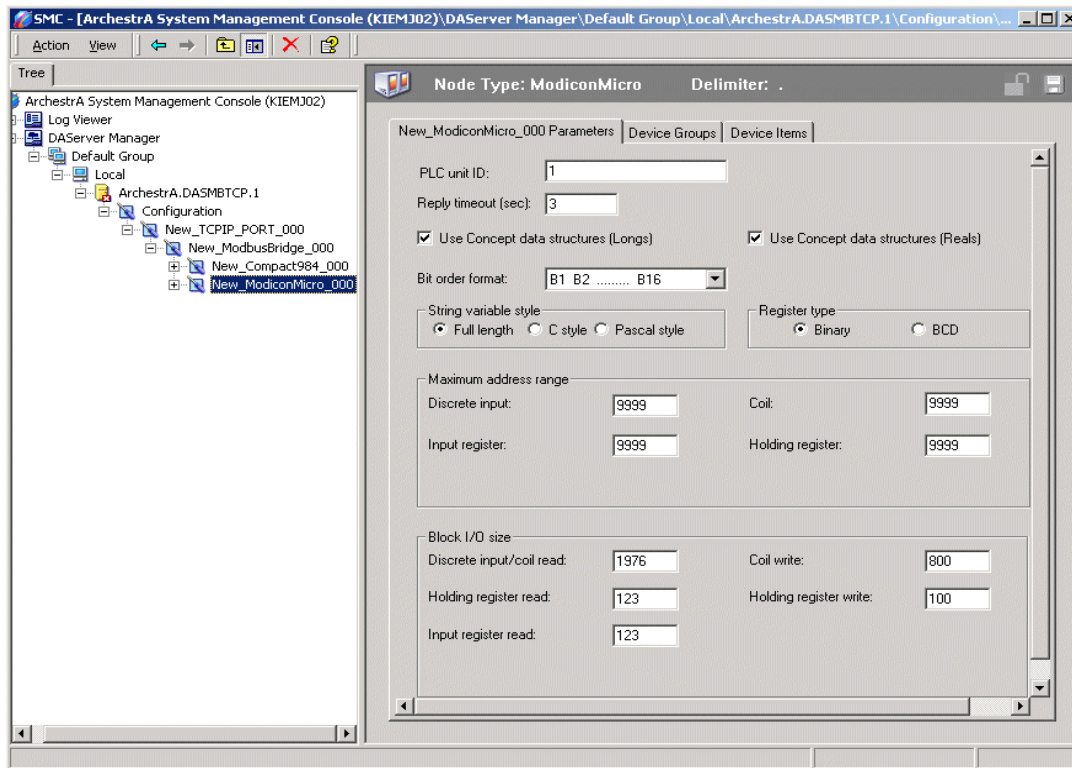
1. Right-click on your **ModbusBridge** branch.
2. Select **Add ModiconMicro Object** from the shortcut menu.
  - A new ModiconMicro object is created as a node in the hierarchy tree.
  - It is named **New\_ModiconMicro\_000** by default.

---

**Note** You can add up to 247 of this type of object to the hierarchy.

---

3. Rename as appropriate.
  - The **New\_ModiconMicro\_000 Parameters** configuration view is displayed.



This configuration view has 9 elements that are configurable.

- **PLC unit ID:** Enter the PLC unit ID.

The Bridge's internal configuration contains a UnitID parameter which can be set to override the Unit\_ID address received in the message from the server. In other words, when the Unit\_ID box is 0 (zero) the bridge routes the message to its configured Slave device. If the server's Unit\_ID is set to 0 (zero), the message will be delivered to the Slave device whose address is defined in the UnitID box of the Bridge. If the server's Unit\_ID is set to a non-zero value (range 1...255), the message will be delivered to the Slave device at that numerical address, regardless of the contents of the UnitID box in the Bridge.

  - The minimum value is 0 (zero).
  - The maximum value is 255.
  - The default value is 1 (one).
- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).
  - The maximum value is 60.
  - The default value is 3 (three).
- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.
- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length (space padded)

- C style (null terminated)
- Pascal style (includes length specifier)
- The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD
  - The default register type is Binary.
- **Maximum address range:** There are four sub-elements in this Maximum addressable registers box. The maximum addressable registers can be obtained from the Modicon Concept or Modsoft configuration programs. The PLC will return an error if a register outside of this range is used to read data. The MBTCP DAServer filters out registers outside of this range and logs error messages.
  - **Discrete input:** Enter the maximum number of addressable discrete inputs/read coils in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999.
    - The default value is 9999.
  - **Coil:** Enter the maximum number of addressable write coils in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999.
    - The default value is 9999.
  - **Input register:** Enter the maximum number of addressable input registers in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999.
    - The default value is 9999.
  - **Holding register:** Enter the maximum number of addressable holding registers in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 9999.
    - The default value is 9999.
- **Block I/O size:** This Block I/O Sizes box contains five sub-elements. The DAServer uses the Block I/O sizes to maximize data throughput. The MBTCP DAServer uses a 256-byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.
  - **Discrete input/coil read:** Enter the maximum number of discrete inputs or coils to read at one time.



- The minimum value is 1 (one).
- The maximum value is 1976.
- The default value is 1976.
- **Coil write:** Enter the maximum number of coils to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 800.
  - The default value is 800.
- **Holding register read:** Enter the maximum number of holding registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.
- **Holding register write:** Enter the maximum number of holding registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 100.
  - The default value is 100.
- **Input register read:** Enter the maximum number of input registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.

## TSXMomentumRS Object

The TSXMomentumRS object is created from the ModbusBridge branch of the DAServer Manager hierarchy.

### To add TSXMomentumRS objects to your MBTCP hierarchy

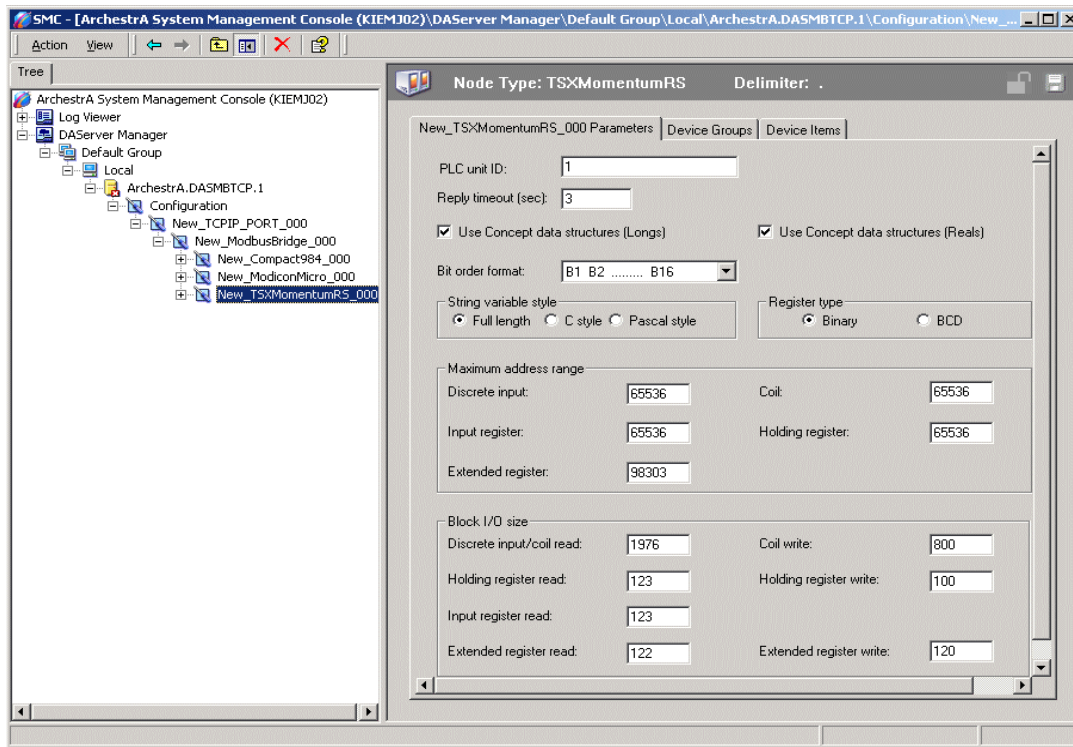
1. Right-click on your **ModbusBridge** branch.
2. Select **Add TSXMomentumRS Object** from the shortcut menu.
  - A new TSXMomentumRS object is created as a node in the hierarchy tree.
  - It is named **New\_TSXMomentumRS\_000** by default.

---

**Note** You can add up to 247 of this type of object to the hierarchy.

---

3. Rename as appropriate.
  - The **New\_TSXMomentumRS\_000 Parameters** configuration view is displayed.



This configuration view has 9 elements that are configurable.

- **PLC unit ID:** Enter the PLC unit ID.  
The Bridge's internal configuration contains a UnitID parameter which can be set to override the Unit\_ID address received in the message from the server. In other words, when the Unit\_ID box is 0 (zero) the bridge routes the message to its configured Slave device. If the server's Unit\_ID is set to 0 (zero), the message will be delivered to the Slave device whose address is defined in the UnitID box of the Bridge. If the server's Unit\_ID is set to a non-zero value (range 1...255), the message will be delivered to the Slave device at that numerical address, regardless of the contents of the UnitID box in the Bridge.
  - The minimum value is 0 (zero).
  - The maximum value is 255.
  - The default value is 1 (one).
- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).
  - The maximum value is 60.
  - The default value is 3 (three).
- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.

- Checked — selected
- Not checked — not selected
- The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.
- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length (space padded)
  - C style (null terminated)
  - Pascal style (includes length specifier)
  - The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD
  - The default register type is Binary.
- **Maximum address range:** There are five sub-elements in this Maximum addressable registers box. The maximum addressable registers can be obtained from the Modicon Concept or Modsoft configuration programs. The PLC will return an error if a register outside of this range is used to read data. The MBTCP DAServer filters out registers outside of this range and logs error messages.
  - **Discrete input:** Enter the maximum number of addressable discrete inputs or read coils in the PLC.
    - The minimum value is 1 (one).
    - The maximum value is 65536.
    - The default value is 65536.

- **Coil:** Enter the maximum number of addressable write coils in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Input register:** Enter the maximum number of addressable input registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Holding register:** Enter the maximum number of addressable holding registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 65536.
  - The default value is 65536.
- **Extended register:** Enter the maximum number of addressable extended registers in the PLC.
  - The minimum value is 1 (one).
  - The maximum value is 98303.
  - The default value is 98303.
- **Block I/O size:** The Block I/O Sizes box contains seven sub-elements. The DAServer uses Block I/O Sizes to maximize data throughput. The MBTCP DAServer uses a 256-byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.
  - **Discrete input/coil read:** Enter the maximum number of discrete inputs or coils to read at one time.
    - The minimum value is 1 (one).
    - The maximum value is 1976.
    - The default value is 1976.
  - **Coil write:** Enter the maximum number of coils to write at one time.
    - The minimum value is 1 (one).
    - The maximum value is 800.
    - The default value is 800.
  - **Holding register read:** Enter the maximum number of holding registers to read at one time.
    - The minimum value is 1 (one).
    - The maximum value is 123.
    - The default value is 123.

- **Holding register write:** Enter the maximum number of holding registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 100.
  - The default value is 100.
- **Input register read:** Enter the maximum number of input registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 123.
  - The default value is 123.
- **Extended register read:** Enter the maximum number of extended registers to read at one time.
  - The minimum value is 1 (one).
  - The maximum value is 122.
  - The default value is 122.
- **Extended register write:** Enter the maximum number of extended registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 120.
  - The default value is 120.

## ModbusPLCRS Object

The ModbusPLCRS object is added to the DAServer hierarchy from the ModbusBridge branch.

### To add ModbusPLCRS objects to your MBTCP hierarchy

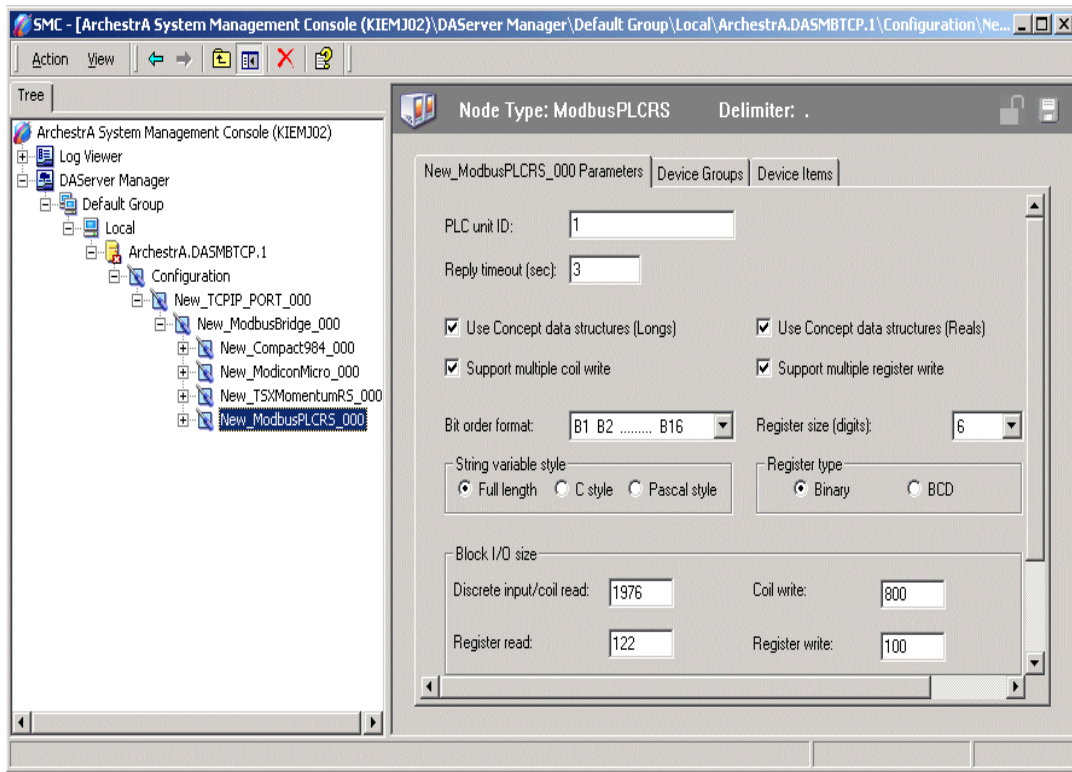
1. Right-click on your **ModbusBridge** branch.
2. Select **Add ModbusPLCRS Object** from the shortcut menu.
  - A new ModbusPLCRS object is created as a node in the hierarchy tree.
  - It is named **New\_ModbusPLCRS\_000** by default.

---

**Note** You can add up to 247 of this type of object to the hierarchy.

---

3. Rename as appropriate.
  - The **New\_ModbusPLCRS\_000 Parameters** configuration view is displayed.



This configuration view has 11 configurable elements.

- **PLC unit ID:** Enter the PLC unit ID.
 

The Bridge's internal configuration contains a UnitID parameter which can be set to override the Unit\_ID address received in the message from the server. In other words, when the Unit\_ID box is 0 (zero) the bridge routes the message to its configured Slave device. If the server's Unit\_ID is set to 0 (zero), the message will be delivered to the Slave device whose address is defined in the UnitID box of the Bridge. If the server's Unit\_ID is set to a non-zero value (range 1...255), the message will be delivered to the Slave device at that numerical address, regardless of the contents of the UnitID box in the Bridge.

  - The minimum value is 0 (zero).
  - The maximum value is 255.
  - The default value is 1 (one).
- **Reply timeout (sec):** Enter the amount of time the server will wait for an acknowledgment.
  - The minimum value is 1 (one).
  - The maximum value is 60.
  - The default value is 3 (three).
- **Use Concept data structures (Longs):** Select to read data from the PLC in concept data structure format for Long item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.

- Checked — selected
- Not checked — not selected
- The default is checked.
- **Use Concept data structures (Reals):** Select to read data from the PLC in concept data structure format for Real item types. If checked, the DAServer will process the data in the same register order as the Concept programming software.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Support multiple coil write:** Select for the PLC to write to multiple coils in one message. If not selected, the PLC will write to a single coil in one message.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.
- **Support multiple register write:** Select for the PLC to write to multiple registers in one message. If not selected, the PLC will write to a single register in one message.
  - Checked — selected
  - Not checked — not selected
  - The default is checked.

---

**Note** Certain things have to be understood when the Support Multiple Register Write Parameter is not selected in the Generic PLC configuration. When this parameter is not selected, it implies that the PLC does not support multiple register writes and the server will only write single registers to the PLC.

What this implies is that items that contain more than one register cannot be written either. For example, items such as 4xxxxx L, 4xxxxx I, 4xxxxx U, 4xxxxx F, 4xxxxx-4xxxxx M, 5 HRL, 5 HRF, 5 PV, 5 HRU, and 4xxxxx-4xxxxx cannot be written. When you try to write to the PLC with this parameter not selected, the following error message will be logged to the logger, "Cannot write to multiple register item: 4xxxxx L on Node: TCPPort.GenPLC. The PLC configurable parameter Support Multiple Register Write is not checked."

---

- **Bit order format:** The format of the bit order entered into the PLC.
  - When the Bit order format is selected as B1 B2 ... B16, it means the bit order starts from left to right (the Most Significant Bit = Bit 1 and the Least Significant Bit = Bit 16).
  - When the Bit order format is selected as B16 B15 ... B1, it indicates that the bit order starts from right to left (MSB = Bit 16 and LSB = Bit 1).
  - The default is bit order starts from left to right.

- **Register size (digits):** Select the correct register size for addressing the PLC.
  - 4-digit is used for addressing the Modbus Generic 4-Digit PLCs.
  - 5-digit applies to the Modbus Generic 5-Digit PLCs.
  - 6-digit is used for addressing the Modbus Generic 6-Digit PLCs.
  - The default value is 6, for the Modbus Generic 6-Digit PLCs.

---

**Note** The selection for the Register size determines the maximum address range. They are changeable as in other supported PLCs listed in Appendix A, Supported DASMBTCP Hardware and Firmware.

For 4-digit, the maximum value is 999.

For 5-digit, the maximum value is 9999.

For 6-digit, the maximum value is 65536.

---

- **String variable style:** PLC string-data format. Select the option for the style used by the device to store strings in its registers.
  - Full length (space padded)
  - C style (null terminated)
  - Pascal style (includes length specifier)
  - The default style is Full length.
- **Register type:** Select either Binary or BCD for the register type being used.
  - Binary
  - BCD
  - The default register type is Binary.
- **Block I/O size:** This Block I/O Sizes box contains four sub-elements. The DAServer uses Block I/O Sizes to maximize data throughput. The MBTCP DAServer uses a 256-byte buffer to read or write data to the PLC. The maximum value is the maximum number of registers that can be read or written from/to the PLC in one command.
  - **Discrete input/coil read:** Enter the maximum number of discrete inputs or coils to read at one time.
    - The minimum value is 1 (one).
    - The maximum value is 1976.
    - The default value is 1976.
  - **Coil write:** Enter the maximum number of coils to write at one time.
    - The minimum value is 1 (one).
    - The maximum value is 800.
    - The default value is 800.
  - **Register read:** Enter the maximum number of extended registers to read at one time.
    - The minimum value is 1 (one).



- The maximum value is 122.
- The default value is 122.
- **Register write:** Enter the maximum number of holding registers to write at one time.
  - The minimum value is 1 (one).
  - The maximum value is 100.
  - The default value is 100.

The logical endpoint for each branch of the MBTCP hierarchy tree is a hardware device (PLC).

---

**Note** The default name created from adding a hierarchy object is in the format of **New\_ObjectName\_###**, where **ObjectName** is the name of the object type and **###** is a numeric value starting from "000" enumerated sequentially per hierarchy object. The hierarchy object name is up to 32 characters long. The link name for the OPC items is constructed by assembling the respective object names of the nodes along the hierarchy tree in the logical order, starting from this DASServer's TCPIP\_PORT root down to the leaf. Therefore, the link name is always unique for the DASServer.

---

---

**Note** In order to use the DASServer, you must activate it. See the DASServer Manager documentation for information about how to activate and deactivate the DASServer.

---

## String-Data Handling

The MBTCP DASServer can process three different configurable string variable styles:

- Full Length
- C Style
- Pascal

Depending on what string style the PLC is using, you can configure the server using the user interface in the PLC configuration view in order to use the appropriate string variable style.

### Full Length Style

If strings are read using the Full Length Style, each string always uses all of the registers allocated. The PLC string is stored in the server string as is.

If a string is written and the string is shorter than the allocation of registers, it is padded with ASCII space characters (hex 20).

For example:

If the string "Wonderware" is stored in the string item "400001-400010 m," registers 400001 through 400005 contain the string "Wonderware" and registers 400006 through 400010 contain spaces.

If the string "Wonderware" is stored in the string item "400001-400005 m," registers 400001 through 400005 contain the string "Wonderware" and no spaces are stored.

If the string "Wonderwareee" is stored in the string item "400001-400005 m," registers 400001 through 400005 contain the string "Wonderware" and no spaces are stored.

A message is placed in the logger indicating that the string was truncated.

## C Style

If a string is read using the C Style, the string always uses all of the registers allocated. The PLC string is stored in the server string as is, except that the last character contained in the last register of the string is replaced with a null character (hex 00).

If a string is written and the string is shorter than the allocation of registers, it is padded with ASCII null characters (hex 00).

For example:

If the string "Wonderware" is stored in the string item "400001-400010 m," registers 400001 through 400005 contain the string "Wonderware" and registers 400006 through 400010 contain nulls.

If the string "Wonderware" is stored in the string item "400001-400005 m," registers 400001 through 400005 contain the truncated string "Wonderwar0" with a null character replacing the last character "e."

A message is placed in the logger indicating that the string was truncated.

## Pascal Style

If strings are read using the Pascal Style, each string uses a length obtained from the first byte of the string to store data in the server. The PLC string is stored in the server string as is, up to the length obtained from the first byte of the string. If the length is greater than the number of registers defined in the item, then the PLC string is stored in the server string as is, up to the maximum number of registers.

The first byte written of any string of this style contains the character count. The string being written starts in the second byte. If a string is written and the string plus the character count are shorter than the allocation of registers, it is padded with ASCII null characters (hex 00).

For example:

If the string "Wonderware" is stored in the string item "400001-400010 m," registers 400001 through 400006 contain the string "(10)Wonderware0" and registers 400007 through 400010 contain nulls. The (10) in the string implies one byte containing the character count.

If the string "Wonderware" is stored in the string item "400001-400005 m," registers 400001 through 400005 contain the truncated string "(9)Wonderwar."

A message is placed in the logger indicating that the string was truncated.

---

## Message Optimization

The MBTCP DAServer uses Multi read and write commands to optimize PLC read/write messages. The MBTCP DAServer optimizes the reading and writing of data by grouping points that are in consecutive registers. The Block I/O sizes parameters control the buffer size. The default is to maximize the buffer size.

---

**Note** The number of bytes for the query and response buffers must not exceed the Modbus maximum buffer size of 256 bytes.

---

## Configuring Device Group and Device Item Definitions

The **Device Groups** tab in the DAServer Manager user interface is used to create new, modify, or delete device group definitions for an object. For DDE/SuiteLink communications, one or more device group definitions must exist for each PLC that the DAServer will communicate with.

Each device group (topic) definition should contain a unique name for the PLC associated with it.

### Device Group Definitions

The **Device Groups** dialog box, which is displayed by clicking the **Device Groups** tab in the **New\_<Name>PLC\_000 Parameters** configuration view, is used to perform the following activities:

- Adding, defining, and deleting device groups.

---

**Note** When you add a new device group, enter a unique name.

---

- Configuring the DAServer to receive unsolicited messages.
- Configuring default update intervals.
- Editing update intervals for the device groups.

---

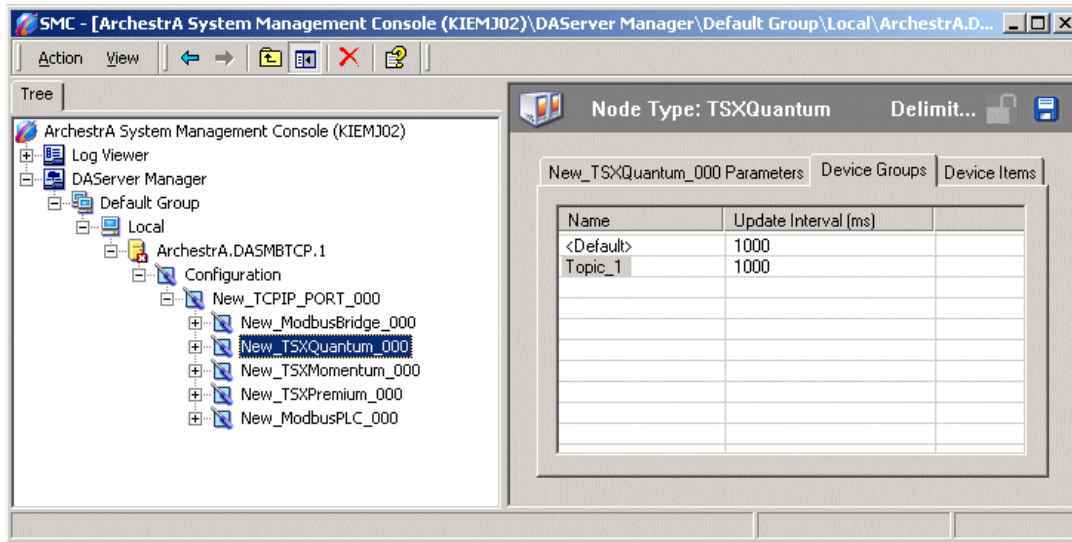
**Note** When you select another part of the DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.

---

#### To create or add device groups

1. Right-click in the **Device Groups** dialog box.
2. Select the **Add** command from the shortcut menu.

- When you add a new device group, enter a unique name (up to 32 characters long).



### To make changes on device groups' names

Change a device group's name for an object as follows:

- In the **Name** column, double-click on the device group's name to be modified and make the change.

### To delete device groups

Deleting a device group from the list can be performed as follows:

1. Right-click on the device group to be deleted.
2. Select the **Delete** command from the shortcut menu.

---

**Note** When you select another part of the MBTCP DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.

---

### To configure the MBTCP DAServer to receive unsolicited messages

This option is available only to the PLC nodes that are directly connected to the TCPIP\_Port node. PLC nodes that are connected underneath the ModbusBridge node do not have this capability.

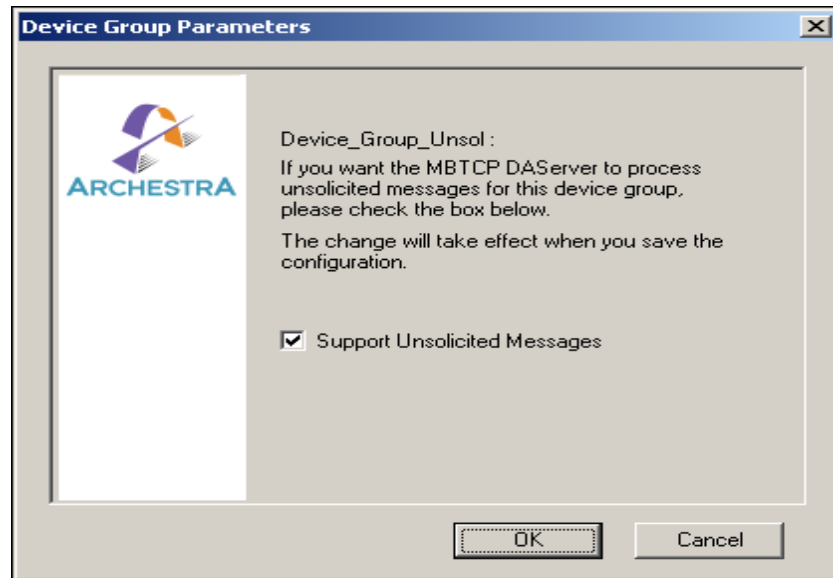
---

**Note** Some PLCs, for example the TSX Premium PLC, can send Holding Register and Coil unsolicited data to the MBTCP DAServer, while other PLCs, for example the TSX Quantum PLC, can send the Holding Register unsolicited data only to the DAServer.

---

1. Click on the PLC's name in the PLC branch of the DAServer hierarchy.
2. Select the **Device Group** tab of the configuration view pane at right.

3. Add a new device group or select an existing device group.
4. Right-click on the device group name, then select **Edit** from the shortcut menu.
  - The **Device Group Parameters** dialog box is displayed.



5. In the **Device Group Parameters** dialog box, select the **Support Unsolicited Messages** check box.
6. Click the **OK** button to close the dialog box.

---

**Note** Since the status of **Support Unsolicited Messages** check box cannot be readily viewed from the **Device Groups** tab, proper naming of device groups which support unsolicited messages is strongly recommended.

---

7. Save the configuration change by clicking the **Save** icon located at the upper-right corner of the configuration view pane.

---

**Note** For more information on unsolicited messages, see [Unsolicited Message Handling](#).

---

### To configure default update intervals

1. To configure a default update interval for the object, right-click in the **Device Groups** dialog box.
2. Select **Config Default Update Interval** from the shortcut menu.

### To make changes on update intervals

A change on an update interval for an object can be made as follows:

- Double-click on the value to be modified in the **Update Interval** column and make the change.

- Update Interval is the frequency (in milliseconds) that the DAServer acquires data from the topics associated with that device group.
- Different topics can be polled at different rates in a PLC by defining multiple device-group names for the same PLC and setting a different Update Interval for each device group.
- If you also select the "Support Unsolicited Message" checkbox as described above, duplicate updates could be received by the items associated with the device group. To avoid duplicate updates, set the Update Interval to 0 (zero).

---

**Note** When you select another part of the MBTCP DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.

---

Each configuration view associated with nodes/objects in the DAServer hierarchy tree has a common feature, the **Save** button.

1. When you modify any parameters in the **Device Groups** dialog box, click **Save** to save and implement the new modifications.
  - If you do not click **Save**, the configuration is reset to its original condition (since the last save).
2. After all modifications, you must save when prompted for the new data to be saved to the configuration set.

## Device Item Definitions

The **Device Items** tab in the **New\_<Name>PLC\_000 Parameters** configuration view is used to define aliases to actual PLC items. The **Device Items** dialog box is the place where the following activities are performed:

- Creating new device item definitions for PLC items.
- Modifying the existing device items.
- Deleting device items.
- Archiving the created list of device items to a .csv file, a file with values separated by commas.
- Importing a .csv file into the **Device Items** tab.

Each device item definition should contain a unique name for the PLC associated with it.

The **Device Items** dialog box has the following two columns:

- **Name:** This column defines the alias names to actual PLC items.
- **Item Reference:** The actual PLC item names, linked to the created aliases, are defined in this column.

For example:

For Modicon holding register 400001 and 400010, the following entries can be created.

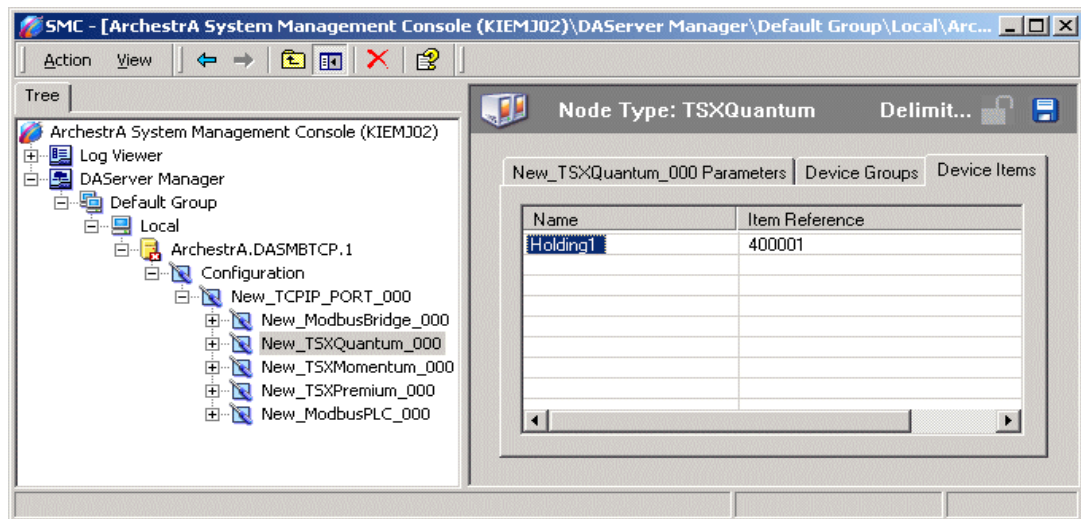
Name	Item Reference
Holding1	400001
Holding10F	400010 F

**Note** When you create or add a new device item, a unique name needs to be entered for it.

Once the Device Items feature is utilized to configure item names, it provides the DASServer with the capability to perform OPC Item browsing. When the DASServer is running and an OPC client requests item information, the configured items will show up under the PLC hierarchy node.

### To create or add device items

1. Right-click in the **Device Items** dialog box.
2. Select the **Add** command from the shortcut menu.
  - A device item is created in the **Name** column, and it is numerically named by default. For example, Item\_0, Item\_1, and so on.
3. Change the default name by double-clicking on it and entering the new name.
  - Enter a unique name for the new device item. For example, "Holding1."



### To add item references

Item references for each of the device items that have been created can be added as follows:

1. In the **Item Reference** column, double-click on the area in the same horizontal line as the selected device item.

2. Type in the actual PLC item name in the frame that appears.
  - For example, "400001."
3. Click anywhere in the dialog box or press the ENTER key to have the change take effect.

---

**Note** System items are not valid item reference, but DAServer-specific system items are ok.

---

#### **To rename a device item from the list**

1. Right-click on the device item to be renamed.
2. Select the **Rename** command from the shortcut menu and enter the new device item name.
3. Click anywhere in the dialog box or press the ENTER key to apply the change.

#### **To delete a device item from the list**

1. Right-click on the device item to be deleted.
2. Select the **Delete** command from the shortcut menu.
  - The device item and its corresponding actual PLC item name will be deleted from the dialog box.

---

**Note** When you select another part of the MBTCP DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.

---

#### **To clear all device items**

1. Right-click anywhere in the **Device Items** dialog box.
2. Select the **Clear All** command from the shortcut menu.
  - All the device items listed in the dialog box, including their corresponding actual PLC item names, will be deleted.

The **Export** and **Import** features on the shortcut menu of the **Device Items** dialog box enable you to export and import the DAServer device item data to and from a CSV file, after the configuration of the Device Items has been completed. These features provide you with the following capabilities:

- Archive lists of device items.
- Import an archived list of device items into the **Device Items** dialog box when you need to utilize or reconfigure any of the device items on the archived list.
- Perform an off-line, large-scale edit on the item data configured for a device item list.
- Import what has been edited back into the Device Items configuration.

#### **To export device items**

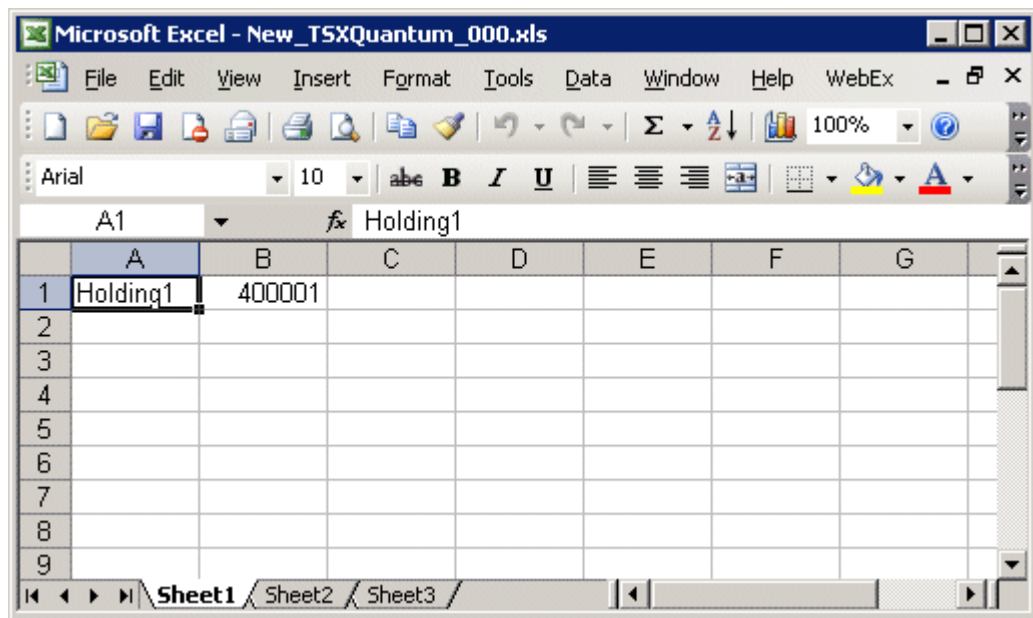
1. Right-click anywhere in the **Device Items** dialog box.



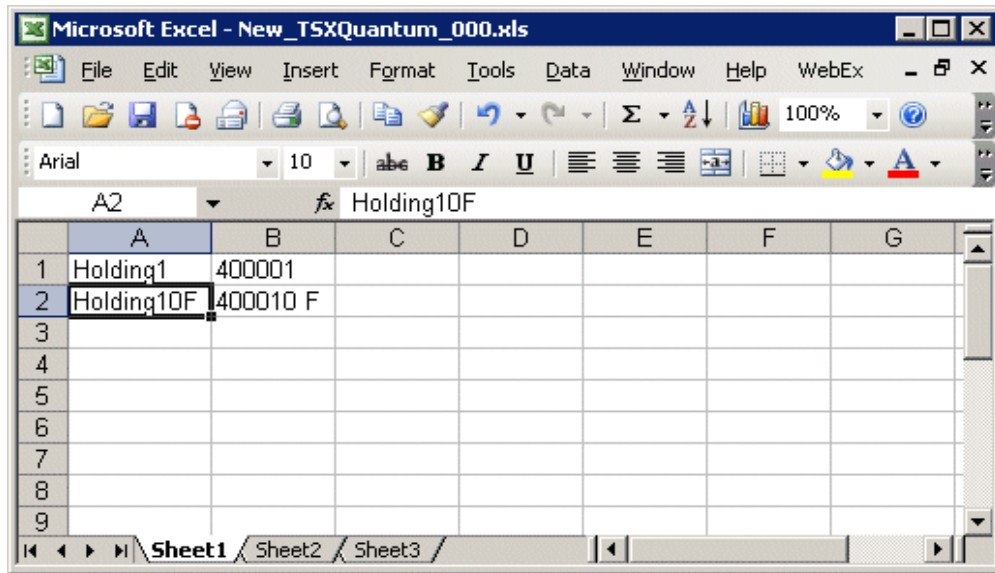
2. Select the **Export** command from the shortcut menu.
  - The standard **Save As** dialog box appears.
  - The file name has defaulted into "PLC Hierarchyname.csv," within the current-system-configured default directory.
3. Accept the defaults to save the file.
  - The file is saved as New\_<PLC Name>\_000.csv.
  - It is editable in Microsoft Excel.

However, if you prefer to save the list someplace else and rename it, perform the following steps after step 2.

4. Select the folder into which the list is to be saved.
5. Name the list to be archived.
6. Click the **Save** button.
  - The whole list will be saved as a .csv file in Excel.



The file can now be edited off-line. It contains one row for each item configured with two columns, Name and Item Reference, respectively.



### To import device items

1. To import the list, right-click anywhere in the **Device Items** dialog box.
2. Select the **Import** command from the shortcut menu.
3. Select the archived list (.csv file) to be imported from the folder in which it is saved.
4. Click the **Open** button.
  - The whole list will be brought into the **Device Items** dialog box.

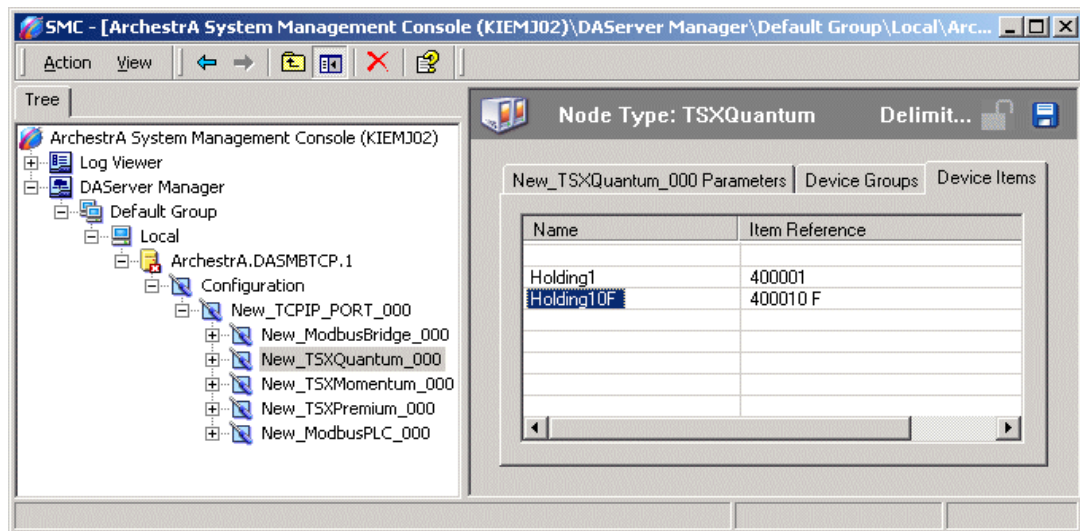
---

**Note** When the list to be imported contains duplicate names as found in the current list but the Item References are different, a dialog box will appear to prompt you to make a selection.

---

### To import device item data that has been edited off-line

1. Right-click anywhere in the **Device Items** dialog box.
2. Clear all the item data you wish to replace with the edited .csv file by selecting the **Clear All** command.
  - The data will be cleared after you click on **Yes** to confirm the deletion.
3. Select the **Import** command from the shortcut menu.
  - The standard **Open** dialog box appears.
  - It defaults to the .csv file extension within the current-system-configured default directory.
4. Browse for the specific CSV file you want to import, select it, then click on the **Open** button.
  - The DAServer Manager will import the edited file and deposit it in the **Device Items** dialog box.



- During the imported file processing:
  - New item references will be added based on unique names.
  - If there are duplicate names, you will be provided with the option to replace the existing entry with the new entry, or ignore the new entry.

## Scan-Based Message Handling

Wonderware's DAServers are based on the concept of polling a hardware device for information. This polling is driven by a need which is expressed in the form of requests from one or more clients. Once a particular piece of information has been requested by a client, the DAServer formulates its own request and sends that request to the hardware device. The DAServer then waits for a response to its request. Once the information has been received, the DAServer passes that information back to the client, and repeats the process until all clients have ceased requesting information.

The rate at which the DAServer will poll a particular device for a specific piece of information is defined in the device group (topic definition) inside the DAServer, using a parameter called the Update Interval. When setting this parameter, there is always a trade-off between the update speed of the device group and the resulting decrease in system responsiveness.

Since you more than likely want very fast response, the temptation is to set the Update Interval to a value close to 0 (zero) seconds. However, if every point is polled at this rate, the entire system will suffer due to slow response time. Therefore, you should compromise, and set the Update Interval to a more reasonable value. You could also create multiple device groups for each device, setting the Update Interval to different values, then assigning different items to different device groups depending on how quickly the values change and how quickly you want to see an update of those changes.

Some items, like alarms, change very infrequently but because of their importance require very fast updates. For those kinds of items, you should set the Update Interval at a very small value. If you desire an immediate response, set the Update Interval at 1 (one).

## Unsolicited Message Handling

In the world of PLCs and DAServers, it is obvious that a PLC will know when a critical event has occurred before the DAsServer will have a chance to poll for that data. Therefore, it would seem natural that if a critical event occurs, the PLC should have the capability to inform the DAsServer immediately, without having to wait for the DAsServer to poll it.

This is the role of an unsolicited message. Once a PLC has determined that a critical condition (such as an alarm event) exists, it can generate a message that is immediately sent to the DAsServer without a prior request from the DAsServer. Compared to polling the PLC for an irregular event at fixed intervals, sending unsolicited messages from the PLC to the DAsServer when the event occurs will not only cut down the network traffic, but may also get the data to the server more promptly.

---

**Note** The unsolicited messaging feature is available only to controllers that are directly connected to the TCPIP\_PORT. Controllers under the ModbusBridge Object hierarchy cannot utilize this feature.

---

In accordance with the protocol, the receiver of unsolicited messages must acknowledge the receipt of unsolicited messages.

---

**Note** For the step-by-step procedure on how to configure the DAsServer to receive unsolicited messages, see Device Group Definitions.

---

## Archiving Configuration Sets

After your DAsServer has been configured, you can archive that specific configuration. You can archive more than one configuration set, and subsequently choose different configurations for different purposes.

### To archive configuration sets

1. In the DAsServer Manager, right-click on the **Configuration** node in the hierarchy below your DAsServer.
2. Select **Archive Configuration Set** from the shortcut menu.
3. In the **Archive Configuration Set** configuration view, provide a Configuration Set Name.
4. Click **Archive**.
  - All current configuration values are saved to the archived set.

Once you have archived at least one configuration set, you can select it for use.

### To use different configuration sets from the current one

1. Make sure the DAsServer is not running.

2. In the DAServer Manager, right-click the **Configuration** node in the hierarchy below your DAServer.
3. Select **Use Another Configuration Set** from the shortcut menu and click on a configuration set in the sub-menu.
  - All parameters in the DAServer configuration hierarchy change to the chosen configuration set.

## Hot Configuration

If a parameter value change takes effect right away while the DAServer is running, the parameter is a hot-configurable parameter. Certain parameters in the MBTCP DAServer are hot-configurable. Incorporated in the DAServer are the following hot-configuration functionalities:

- Modifying Global Configuration parameters.
- Adding, deleting, or modifying device nodes (without affecting any other device nodes, excluding the children of the modified device nodes).
- Adding, deleting, or modifying device groups in the **Name** and **Update Interval** columns in the **Device Groups** tab.
- Adding, deleting, or modifying Name and Item Reference in the **Device Items** tab.

Limited support is provided for the hot configuration for the server-specific configuration parameters in this release. You can modify server-specific parameters while the server is active. However, to have those changes take effect, you have to restart the DAServer.

The following parameters are hot configurable. They can be modified online and changes will take affect without restarting the DAServer.

- Reply timeout
- String variable style
- Register type

---

**Note** If changes are made to server-specific parameters while the server is active, the DAServer will issue a warning message to the logger.

---



## CHAPTER 3

# Item Names

The Wonderware MBTCP DAServer supports a variety of data types for the Modicon controllers. It also supports item names that follow the conventions described for the following Modicon PLCs:

- TSX Quantum (6-Digit)
- TSX Momentum (6-Digit)
- TSX Premium (6-Digit)
- Generic Modbus (4-Digit, 5-Digit, 6-Digit)
- Compact 984 (RS232, 5-digit and 6-Digit)
- Modicon Micro (RS232, 5-digit)
- TSX Momentum (RS232/RS485, 6-Digit)
- Generic Modbus (RS232/RS485, 4-Digit, 5-Digit, and 6-Digit)

## Contents

- Data and Register Types
- Modbus Item Naming
- DAServer Standard System Items
- Generic OPC Syntax

## Data and Register Types

When a client sends a read/write request to the MBTCP DAServer, the DAServer needs to know its data type and size. In order to determine this information, the MBTCP DAServer parses the item name to get the register number, data type, and size. The DAServer builds messages with items sorted by PLC, register type, register number, and topic name, allowing the DAServer to optimize the number of registers that can be read in one scan command.

The following table contains the types of data for the Modicon controllers, TSX Quantum, TSX Momentum, TSX Premium, Generic Modbus (4-Digit, 5-Digit, and 6-Digit), Compact 984, and Modicon Micro, supported by the MBTCP DAServer.

<b>TSX Quantum/TSX Momentum/TSX Premium/Generic Modbus/Compact 984/Modicon Micro Data Type</b>	<b>Range</b>
Discrete (bit/Boolean)	0 (zero), 1 (one)
Signed Short Integer (signed 16-bit integer)	0 (zero) to 65535
Unsigned Short Integer (unsigned 16-bit integer)	-32768 to 32767
Signed Long Integer (signed 32-bit integer)	-2147483648 to 2147483647
Unsigned Long Integer (unsigned 32-bit integer)	0 (zero) to 4294967295
REAL (32-bit float)	32-bit IEEE
String (ASCII)	246 characters (Read) 200 characters (Write)

---

**Note** The unsigned integer data type "U" has a valid range of 0 (zero) to 2147483647 when accessed through DDE/SL client; for OPC clients the valid range is from 0 to 4294967295.

---

**Note** System-defined types will not be supported as block reads. A read on any of these data types will return only the first element.

---

The following table lists the PLC register types, the data types contained in the registers, and what each is processed as.

<b>PLC Register Type</b>	<b>Data Type Contained in the Register</b>	<b>Processed As</b>
Discrete Output (Coil)	Discrete	Real Time Data
Discrete Input	Discrete	Real Time Data
Holding Register	Discrete, Integer, Float, and String	Real Time Data
Input Register	Discrete, Integer, Float, and String	Real Time Data
Extended Register	Discrete, Integer, Float, and String	Real Time Data

## Modbus Item Naming

The Modbus-family controllers store data in the Registers. The MBTCP DAServer supports item names that are consistent with the point naming conventions used by the Modicon PLCs.

The following item naming conventions are described in this section:



- Register-Number Item Names
- Item Names Using the Modicon PLC Register Addresses
- Item Names Using the Absolute Naming Convention
- Item Names Using the Modulo-10000 Naming Convention
- Modulo-10000 Items, BCD Register Type, and Concept Data Structures

---

**Note** The tag-name length with SuiteLink is limited to 32 characters.

---

## Register-Number Item Names

The register number, which is consistent with the point naming convention used by Modicon PLCs, is used as the item name. The Modbus-family PLC address ranges, supported by the DAServer, for the TSX Quantum, TSX Momentum, TSX Premium, Generic Modbus (4-Digit, 5-Digit, and 6-Digit), Compact 984, and Modicon Micro PLCs are shown in the following tables.

The MBTCP DAServer will adhere to this address range for native mode.

Register Type	TSX Quantum/ TSX Momentum	Modicon Micro	Compact 984-265	Compact 984-145	Tag Type	Access
Output Coils	1-65536	1-9999	1-65536	1-9999	Discrete	Read/Write
Contacts	100001-165536	10001-19999	100001-165536	10001-19999	Discrete	Read-Only
Input	300001-365536	30001-39999	300001-365536	30001-39999	Analog	Read-Only
Holding	400001-465536	40001-49999	400001-465536	40001-49999	Analog	Read/Write
Extended	6x0000-6x9999			6x000-6x999	Analog	Read/Write

Register Type	TSX Premium	Tag Type	Access
Output Coils	1-65536	Discrete	Read/Write. Mapped to %M PLC object; that is, Output Coils and Contacts Registers are the same.
Contacts	100001-165536	Discrete	Read-Only. Mapped to %M PLC object; that is, Output Coils and Contacts Registers are the same.
Input	300001-365536	Analog	Read-Only. Mapped to %MW PLC object; that is, Input Registers and Holding Registers are the same.
Holding	400001-465536	Analog	Read/Write. Mapped to %MW PLC object; that is, Input Registers and Holding Registers are the same. Poking a value to a particular bit in the Holding register in this TSX Premium PLC is not supported.

Register Type	Generic Modbus (6-Digit)	Generic Modbus (5-Digit)	Generic Modbus (4-Digit)	Tag Type	Access
Output Coils	1-65536	1-9999	1-999	Discrete	Read/Write
Contacts	100001-165536	10001-19999	1001-1999	Discrete	Read-Only
Input	300001-365536	30001-39999	3001-3999	Analog	Read-Only
Holding	400001-465536	40001-49999	4001-4999	Analog	Read/Write
Extended	6x0000-6x9999	6x000-6x999		Analog	Read/Write

**Note** The x in the Extended register number indicates the file number, where x = 0 implies file number 1, x = 1 implies file number 2, up to x = 9 implies file number 10. The extended memory size in the PLC determines how many extended memory files exist. Each file contains up to 10000 registers. The last file in the PLC will always contain less than 10000 registers.

For example:

A 24K-extended-memory-size PLC contains three (3) files, where the last file contains 4576 registers.

A 72K-extended-memory-size PLC contains eight (8) files, where the last file contains 3728 registers.

A 96K-extended-memory-size PLC contains 10 files, where the last file contains 8304 registers.

1K is 1024 registers.

## Item Names Using the Modicon PLC Register Addresses

The following table lists other item name formats that are consistent with the point naming convention with the MBTCP DAServer suffix.

Item Name	Description
400001	When no spaces and no letters follow the register number, the register contents are treated as a 16-bit unsigned quantity.
400001 S	When a space and the letter "S" follow the register number, the register contents are treated as a 16-bit signed quantity.
400001 I	When a space and the letter "I" follow the register number, the register contents are treated as a 32-bit signed quantity. This takes up two consecutive registers.

Item Name	Description
400001 L	When a space and the letter "L" follow the register number, the register contents are treated as a 32-bit signed quantity.
400001 F	When a space and the letter "F" follow the register number, the register contents are treated as a floating-point quantity. This takes up two consecutive registers.
400001 U	When a space and the letter "U" follow the register number, the register contents are treated as a 32-bit unsigned quantity.
400001-400003 M	When a space and the letter "M" follow the register number or register number pair separated by a dash, the register contents are treated as ASCII data. Each register contains up to two (2) ASCII characters. This example represents six (6) ASCII characters.
300001:10	When a colon and a number from 1 (one) to 16 follow the register number, the register contents are treated as discrete data. This example represents bit 10 of the input register 300001.

## Absolute Notation Item Names

The MBTCP DAServer also uses another naming convention called the Absolute Notation. This naming convention is independent of the PLC model numbers.

The Absolute naming convention allows access to the four Modbus data types, each with an address from 0 to 65535. The data types are indicated by the item name suffix characters.

Item Name	Description
nnnnn <b>DO</b>	Discrete Output Refers to the same data Modbus calls "coils." Valid range is 0 (zero) <b>DO</b> through 65535 <b>DO</b> .
nnnnn <b>DI</b>	Discrete Input Refers to the same data called "contacts" by Modbus. Valid range is 0 (zero) <b>DI</b> through 65535 <b>DI</b> .
nnnnn <b>IR</b>	Input Register Refers to the same data called "input register." Valid range is 0 (zero) <b>IR</b> through 65535 <b>IR</b> .

Item Name	Description
nnnnn <b>HR</b>	Holding Register Refers to the same data Modbus calls "holding register." Valid range is 0 (zero) <b>HR</b> through 65535 <b>HR</b> .
nnnnn <b>PV</b>	Process Variable Refers to holding register, but treated as floating points and assumes two (2) registers per floating-point number. Valid range is 0 (zero) <b>PV</b> through 65535 <b>PV</b> .

The **IR** and **HR** absolute notation can also be combined with the following conversions: **L** (long), **F** (floating), **S** (signed), or **U** (unsigned).

For example:

- 219 **HRS** 16-bit signed integer
- 000 **HRL** 32-bit signed integer
- 100 **HRF** 32-bit floating point
- 000 **HRU** 32-bit unsigned integer

## Modulo-10000 Point Item Names

The MBTCP DAServer uses the Modulo-10000 Points naming convention, where the item name is two registers separated by a dash, with no spaces and no letters following the registers.

Two or three consecutive registers may be interpreted as a single numeric quantity, and each of the component registers must be in the range of 0-9999.

Item Name	Description
400001-400002	Can represent numbers between 0 and 99,999,999. Register 400001 = <9999> and Register 400002 = <9999>.
400005-400007	Can represent numbers between 0 and 2,147,483,646. Register 400005 = <21>, Register 400006 = <4748>, and Register 400007 = <3646>.

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

---

## Modulo-10000 Items, BCD Register Type, and Concept Data Structures

All the integer holding registers, 16- and 32-bit, signed and unsigned, and Modulo-10000 item types honor the configuration parameters Register type and Concept data structures.

When the Register type parameter is BCD and the Use Concept data structures (Longs) parameter is selected, the data is displayed in BCD and written to the PLC in the BCD format. In addition, data that takes up two registers (Longs), except Reals, is displayed and written in the Concept-data-structure format.

The same applies when the Register type parameter is Binary and the Use Concept data structures (Longs) parameter is selected; the data is displayed in Binary and written to the PLC in the Binary format. In addition, data that takes up two registers (Longs), except Reals, are displayed and written in the Concept-data-structure format.

When the Register type parameter is BCD and the Use Concept data structures (Longs) parameter is not selected, the data is displayed in BCD and written to the PLC in the BCD format. In addition, data that takes up two registers (Longs), except Reals, is displayed and written in the non-Concept-data-structure format.

The same applies when the Register type parameter is Binary and the Use Concept data structures (Longs) parameter is not selected; the data is displayed in Binary and written to the PLC in the Binary format. In addition, data that takes up two registers (Longs), except Reals, is displayed and written in the non-Concept-data-structure format.

The Concept-data-structure format implies that data is displayed and written the same way that the Concept program from Schneider Automation handles data.

Concept-data-structure format is where the data is displayed and written in the last-register-to-first-register order.

For example:

When writing the value 2147483646 to the Modulo-10000 item 400001-400003, the value 21 is written first to register 400003, then the value 4748 is written to register 400002, and then the value 3646 is written to register 400001.

The non-Concept-data-structure format is the opposite of the Concept-data-structure format.

The value 21 is written first to register 400001, then the value 4748 is written to register 400002, and then the value 3646 is written to register 400003.

Modulo-10000 items can be displayed and written in the BCD and Binary formats. When the Modulo-10000 item occupies two registers, the maximum value that can be displayed and written is 99999999. When the Modulo-10000 item occupies three registers, the maximum value that can be displayed and written is 2147483646.

Warning messages are logged and the client value status is updated when data is clamped high when reading or writing data. Warning messages will also be displayed when the PLC data does not convert to BCD correctly.

## DAServer Standard System Items

System items provide you with easy access to the DAServer status and diagnostics information. They are treated just like ordinary items with respect to the client. However, in most cases these items are not directly acquired via the communications layer. System item values are usually generated through internal calculations, measurements, and the tracking of the DAS Engine.

System items, like ordinary items, are defined by the following properties:

- **Group** (client group/OPC group): The arbitrary collection of items, not correlated.
- **Hierarchical location** (link name/OPC path. The hierarchical node section of the fully qualified OPC Item ID.): The device the item is attached to.
- **Device group** (OPC access path/topic, or a Scan Group on a hierarchical branch.): A collection of items on the same physical location with the same protocol update rate.

Example:

To check the status of an external device, the reference might be:

```
AREA10.VESSEL1.TIC1.PLC.$SYS$Status
```

---

**Note** This syntax does not refer to the access path/device group. As long as the data requested is from the same external device, the value will always be the same.

---

---

**Note** For DDE/SuiteLink clients, \$SYS\$Status always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node. For OPC clients, \$SYS\$Status can be accessed at all hierarchy levels. \$SYS\$Status at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference \$SYS\$Status at any hierarchy levels other than the root.

---

All system items follow the same naming convention:

- All system items start with \$SYS\$.
- System item name is not case-sensitive.

All system items can be accessed through subscriptions to a device group. However, while some system items return data for that device group, others are server-wide.

## DAServer Global System Item

The following system item refers to specific information regarding a global condition of the DAServer.

System Item Name	Type/ Access Rights	Description	Values
\$\$SYS\$Licensed	Boolean/ Read	Binary status indication of the existence of a valid license for the DAServer. If FALSE, this item causes the DAServer to stop updating existing tags, to refuse activation of new tags, and to reject write requests in addition to setting quality for all items to BAD. If TRUE, the DAServer functions as configured. All instances have the same value.	RANGE: 0, 1  1: Valid license exists. 0: No valid license exists.

## DAServer Device-Specific System Items

The following system items refer to specific information regarding the device(s) the DAServer is connected to.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SYS\$Status	Boolean/ Read	Binary status indication of the connection state to the device (hierarchy level) the item is attached to. The device group (OPC access path/topic) does not affect the value. The status can be good even if individual items have errors. For DDE/SuiteLink clients, \$\$SYS\$Status always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node. For OPC clients, \$\$SYS\$Status can be accessed at all hierarchy levels. \$\$SYS\$Status at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference \$\$SYS\$Status at any hierarchy levels other than the root.	RANGE: 0, 1  1: DAServer connection to the device is intact. 0: Error communicating with the device.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SY\$\$ErrorCode	Longint/ Read	Detailed error code of the communications state to the device. The device group (OPC access path/topic) does not affect the value.	>= 0: Good status (0 is the default state – connected. >0: is some device state like: connecting, initializing, and so on. <0: Error status (value indicates the error).
\$\$SY\$\$ErrorText	String/ Read	Detailed error string of the communications state of the device. The device group (OPC access path/topic) does not affect the value.	Descriptive text for the communications state corresponding to the error code.
\$\$SY\$\$StoreSettings	Integer/ ReadWrite	Used to make the temporary update interval changes via the \$\$SY\$\$UpdateInterval item permanent. If the client pokes a value of 1 into this system item, the currently set update interval is written to the server's configuration file. The value of this system item clears to 0 after being set, if the configuration file write is successful. If the write fails, then the value is set to -1. If the update interval has been changed via the \$\$SY\$\$UpdateInterval item and this item is not poked to 1, the DAServer uses the original update interval for that topic the next time it is started. Reading the item always provides 0. Read/Write values are persisted only if the user sets this system item. The values other than this persist only for the life of the DAServer.	RANGE: -1, 0, 1  -1: Error occurred during saving the configuration file. 0: Read value always if status is OK. 1: Persist settings (cleared immediately).



## DAServer Device-Group-Specific System Items

The following system items refer to specific information regarding device groups that have been configured in the DAServer.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SY\$\$UpdateInterval	DWord/ ReadWrite	Used to access the currently set update interval. It is the current update interval of the device group in milliseconds. A client can poke new values into this item. The value of zero indicates that no non-system items on that topic are updated (data for these items are not acquired from the device).	RANGE: 1...2147483647  0: Topic inactive, no items are updated. Data acquisition is stopped. >0: Expected updated interval for the set of all items in the device group.
\$\$SY\$\$MaxInterval	DWord/ Read	Used to access the currently measured maximum update interval in milliseconds of all items of the corresponding device group. This item is read-only. The value of the slowest item is displayed.	RANGE: 0...2147483647  0: If update interval is 0 or if the status is false. >0: Measured update interval
\$\$SY\$\$WriteComplete	Integer/ ReadWrite	Used to access the state of pending write activities on the corresponding device group. On device group creation (adding items to an OPC group), the value of this system item is initially 1, indicating all write activities are complete – no pokes are pending. If values are poked into any items of the device group, the value of this item changes to 0, indicating write activity is currently in progress. If the server has completed all write activities, the value of this item changes to 1 if all pokes were successful or to -1 if at least one poke has failed. If the value of this item is not zero, the client can poke 1 or -1 to it (poke a 1 to clear errors, or a -1 to test a client reaction on write errors). If the value of this item is zero, it cannot be poked.	RANGE: -1, 0, 1  1: Write complete (no writes are pending – initial state). 0: Writes are pending. -1: Writes completed with errors.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SY\$\$ReadComplete	Integer/ ReadWrite	Used to access the state of initial reads on all items in the corresponding device group. The value is 1 if all active items in a device group have been read at least once. If at least one item in the device group is activated, this item changes to 0. It changes to 1 if all items have been read successfully, or to -1 if at least one item has a non-good quality. Poking a 0 to this item resets the internal read states of all items in this device group. This resets this item to 0. If all items are read again after this poke, this item changes back to 1 or -1.	RANGE: -1, 0, 1  1: Read complete (all values have been read). 0: Not all values have been read. -1: All values have been read but some have a non-good quality.
\$\$SY\$\$ItemCount	DWord/ Read	Used to access the number of items in the corresponding device group. This item is read-only.	RANGE: 0...2147483647  >=0: Number of active items.
\$\$SY\$\$ActiveItemCount	DWord/ Read	Used to access the number of active items in the corresponding device group. This item is read-only.	RANGE: 0...2147483647  >=0: Number of active items.
\$\$SY\$\$ErrorCount	DWord/ Read	Used to access the number of all items (active and inactive) that have errors (non-good OPC quality) in the corresponding topic. If the communications status of a device group is bad, all items have errors. This item is read-only.	RANGE: 0...2147483647  >=0: Number of all items (active and inactive) with errors.
\$\$SY\$\$PollNow	Boolean/ ReadWrite	Poking a 1 to this item forces all items in the corresponding device group to be read immediately (all messages in this device group become due). This is useful if you want to force to get the newest values from the device, regardless of its update interval. This also works on device groups with a zero update interval (manual protocol triggering).	RANGE: 0, 1

---

## Generic OPC Syntax

A DAServer serves as a container for the OPC Groups, which provide the mechanism for containing and logically organizing OPC items. Within each OPC Group, an OPC-compliant client can register OPC items, which represent connections to data sources in the field device. In other words, all access to OPC items is maintained through the OPC Group.

The fully qualified name for an OPC item is called the Item ID (equivalent to Item Name). The syntax for specifying a unique Item ID is DAServer-dependent. In OPC data acquisition DAServers, the syntax can be as follows:

```
AREA10.VESSEL1.TIC1.PLC.400001
```

where each component (delimited by a period) represents a branch or leaf of the field device's hierarchy.

In this example:

- AREA10.VESSEL1.TIC1 is the link name for a DAServer.
- PLC is the name of the target PLC.
- 400001 is the specific data point (Item) desired.
- An item is typically a single value such as an analog, digital, or string value.

Where Item ID describes the syntax for defining the desired data point, OPC provides for another parameter, called Access Path, that defines optional specifications for obtaining that data.

In DAServers, Access Paths are equivalent to Device Groups; it is this parameter that is used to define the update interval between the DAServer and the field device for accessing the values of data points in the PLC.



## CHAPTER 4

# Troubleshooting

This chapter describes troubleshooting tools that can be used to deal with the MBTCP DAServer problems you may encounter.

The DAServer Manager provides access to diagnostics and other statistical data, and the Log Viewer provides access to event messages logged during the operation of a DAServer. Also, your client (for example, InTouch) can monitor connectivity with the PLC through the `$$SYS$$Status` item. Use these tools together with the information in this section to troubleshoot your MBTCP DAServer.

---

**Note** In order to determine the version of your DAServer, perform the following steps. Search for `DASMBTCP.dll`, right-click on the **File Name**, select **Properties** on the context menu, and select the **Version** tab on the **Properties** dialog box. The version of your DAServer is listed under **File Version**.

---

## Contents

- Monitoring Connectivity Status with the PLC
- Monitoring the Status of DAS Conversations
- Error Messages and Codes

## Monitoring Connectivity Status with the PLC

The built-in discrete item, `$$SYS$$Status`, can be used to monitor the status of communications with the PLC. This item is set to the following:

- 0 (zero) when communications with the PLC fails.
- 1 (one) when communications is successful.

---

**Note** For DDE/SuiteLink clients, `$$SYS$$Status` always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node. For OPC clients, `$$SYS$$Status` can be accessed at all hierarchy levels. `$$SYS$$Status` at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference `$$SYS$$Status` at any hierarchy levels other than the root.

---

Enter the following DDE reference formula in the appropriate place in your client:

**=DASMBTCP|ModiconPLC!\$SYS\$Status**

where:

<b>DASMBTCP</b>	is the name of the DAServer application.
<b>ModiconPLC</b>	is the exact device group defined in the DAServer for the PLC.
<b>\$SYS\$Status</b>	is the discrete item used to monitor the status of connectivity with the PLC.

Enter the following OPC item reference syntax when adding the item in your OPC client:

**YourLinkName.\$SYS\$Status**

where:

<b>YourLinkName</b>	is the assembly of hierarchy node names leading to a specific controller device.
<b>\$SYS\$Status</b>	is the discrete item used to monitor the status of connectivity with the controller device.

---

**Note** In the case of a PLC disconnect, the DAServer will retry three times before entering into slow poll mode. In the case of reply time-out, the DAServer will go into slow poll mode immediately.

---

## Monitoring the Status of DAS Conversations

The **InTouch WindowViewer** supports built-in topic names, called **DDEStatus** and **IOStatus**, that can be used to monitor the status of specific DAS conversations.

For example, let us assume that **WindowViewer (VIEW)** is communicating with the MBTCP DAServer to a PLC that has been defined in the DAServer with the topic name **ModiconPLC**. The discrete items, **DDEStatus** and **IOStatus**, are set to:

- 0 (zero) when this DAS conversation failed.
- 1 (one) when this DAS conversation is successful.

## Using DDEStatus and IOStatus in Excel

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

**=view|DDEStatus!ModiconPLC**

or

**=view|IOStatus!ModiconPLC**

where:

<b>view</b>	is the name of the InTouch application.
<b>[DDE][IO] Status</b>	is the built-in topic name used to monitor the status of communications between the DAServer and InTouch.
<b>ModiconPLC</b>	is the exact topic name defined in the server for the PLC.

## Reading Values from the DAServer into Excel

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

**=applicationname|<devicegroup>!itemname**

Example formula:

**=DASMBTCP|ModiconPLC!'<tagname>'**

where:

<b>DASMBTCP</b>	is the name of the DAServer application.
<b>ModiconPLC</b>	is the exact device group name defined in the DAServer for the PLC.
<b>&lt;tagname&gt;</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 **<tagname>** changes in the PLC, the DAServer 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 DAServer from Excel

Values may be written to the DAServer 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 DAServer) in a particular application name (the executable name less the **.exe**).
- 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 **"application name"** and **"topic name"** 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 on the DAServer CD. Also refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

---

## Error Messages and Codes

Generic DAServer error messages and MBTCP-DAServer-specific messages are supported. Use the Log Flag data to customize the messages logged to the Log Viewer. See the Log Viewer online documentation for more information about using log flags.

To troubleshoot DAServer problems, use the following error messages together with the DAServer Manager Diagnostics root data.

In the following DAServer Error Messages table:

- <Message ID> corresponds to the message ID displayed in the DAServer's Diagnostics root in the DAServer Manager.
- <Device> refers to the node name of the device.



## DAServer Error Messages

The following table lists all the generic-DAServer and MBTCP-DAServer-specific error messages that are logged to the Log Viewer.

Error Message	Explanation	Probable Cause	Solution	Log Flag
"CoilRead" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <CoilRead> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"CoilWrite" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <CoilWrite> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"ExtendedRegisterRead" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <ExtendedRegisterRead> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"ExtendedRegisterWrite" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <ExtendedRegisterWrite> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"HoldingRegisterRead" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <HoldingRegisterRead> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"HoldingRegisterWrite" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <HoldingRegisterWrite> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail

<b>Error Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>	<b>Log Flag</b>
"InputRegisterRead" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <InputRegisterRead> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"IPAddress" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <IPAddress> field is absent from the DeviceNode named <PLC Name> in the Configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to build the file.	DASProtFail
"MaxAddrExtendedRegisters" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <MaxAddrExtendedRegisters> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"MaxAddrHoldingRegisters" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <MaxAddrHoldingRegisters> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"MaxAddrInputRegisters" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <MaxAddrInputRegisters> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"MaxAddrReadCoils" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <MaxAddrReadCoils> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"MaxAddrWriteCoils" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <MaxAddrWriteCoils> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail

<b>Error Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>	<b>Log Flag</b>
"MaxQueuedMsgs" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <MaxQueuedMsgs> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"RegisterSize" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <RegisterSize> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"RegisterType" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <RegisterType> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"ReplyTimeout" is missing from the DASMBTCP.AAcfg file under <PLC name>."	The mandatory <ReplyTimeout> field is absent from the DeviceNode named <PLC Name> in the Configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to build the file.	DASProtFail
"StringVariableStyle" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <StringVariableStyle> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
"UnitID" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <UnitID> field is absent from the DeviceNode named <PLC Name> in the Configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to build the file.	DASProtFail
"UseLongConceptData Struct" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <UseLongConceptDataStruct> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail

Error Message	Explanation	Probable Cause	Solution	Log Flag
"UseRealConceptData Struct" is missing from the DASMBTCP.AAcfg file under <PLC name>.	The mandatory <UseRealConceptData Struct> field is absent from the DeviceNode named <PLC Name> in the configuration file.	The entry is deleted from the file manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
<PLC Name> does not allow extended register <Item name>.	The PLC does support extended registers.	The client defined an extended register and it is not supported.	Remove extended register from client.	DASProtFail
<PLC Name>: Clamping a write to <Item Name> - Received <client value>, writing <clamped value>.	The client wrote a value that exceeded the item limits.	The client wrote an invalid value.	The client must write a smaller value.	DASProtWarn
An invalid floating point value was returned by the PLC for item name <PLC Name>.<Item Name>. The value was converted to a negative 3.4e38.	Invalid floating-point number (Negative Infinity). Set the item value to Negative Infinity 3.4e+38.	The values read from the PLC registers cannot be converted to a valid float number.	Check the PLC for the value in the registers.	DASProtFail
An invalid floating point value was returned by the PLC for item name <PLC Name>.<Item Name>. The value was converted to a positive 3.4e38.	Invalid floating-point number (Positive Infinity). Set the item value to Positive Infinity 3.4e+38.	The values read from the PLC registers cannot be converted to a valid float number.	Check the PLC for the value in the registers.	DASProtFail
Connection to PLC <PLC name> at IP Address <Host name> closed, error code = <Error code>.	The PLC closed the connection.	The PLC may be having problems.	Check the PLC.	DASProtFail
CreateItem failed <PLC name>.<Item Name>.	The server encountered an invalid item name.	An invalid item name was defined by the client.	Correct the item name defined by the client.	DASProtFail
CreateItem failed due to invalid item name <PLC name>.<Item Name>.	The server encountered an invalid item name.	The client defined an invalid item name.	The client must define a valid item name.	DASProtFail
DASMBTCP failed to allocate memory.	The server could not allocate memory to continue operating.	Too many items have been defined by client or too many programs are running in this PC.	Stop unwanted programs, define less points by client, or add more memory to the PC.	DASProtFail

Error Message	Explanation	Probable Cause	Solution	Log Flag
DASMBTCPPLCSocket::OnSocketRead(1) failed with errorcode <Error code>.	The PLC returned an error code on a request for data.	The PLC may be having problems.	Check the PLC.	DASProtFail
DASMBTCPPLCSocket::OnSocketRead(2) failed with errorcode <Error code>.	The PLC returned an error code on a request for data.	The PLC may be having problems.	Check the PLC.	DASProtFail
DASMBTCPPLCSocket::OnSocketRead(3) failed with errorcode <Error code>.	The PLC returned an error code on a request for data.	The PLC may be having problems.	Check the PLC.	DASProtFail
DASMBTCPPLCSocket::OnSocketWrite failed with errorcode <Error code>.	The PLC returned an error code on a write to PLC command.	The PLC may be having problems.	Check the PLC.	DASProtFail
Error in Reading from PLC for item <PLC name>.<Item Name>.	The PLC returned an exception.	The PLC may be having problems.	Check the PLC.	DASProtFail
Error in Writing to PLC for item <PLC name>.<Item Name>.	The PLC returned an exception.	The PLC may be having problems.	Check the PLC.	DASProtFail
Fail to Connect to PLC <PLC name> at IP Address <Host name>, error code = <Error code>.	The host name used in the IP Address field is invalid.	Invalid IP Address was configured.	A valid host name or IP Address must be configured.	DASProtFail
Failed to retrieve host information from a host database. Error code = <Error number>.	The host name used in the IP Address field is invalid.	Invalid IP Address was configured.	A valid host name or IP Address must be configured.	DASProtFail
Failed to split the message from ReceiveBuffer for Sequence Number <Sequence Number>, message possibly revoked.	A message returned by the PLC was not found in the requested messages queue.	The message may have timed out and was removed from the queue.	Increase the ReplyTimeout timer.	DASProtFail
Invalid value read for Mod 10000 register: <Item Name> on Node: <PLC Name>. Not a valid BCD-value. Convert to 2147483647.	The Modulo-10000 point value overflows. The value is clamped to 2147483647 to indicate overflow condition.	The maximum usable value represented in three Modulo-10000 registers is 2,147,483,646.	Check the PLC for the value in the registers.	DASProtWarn

Error Message	Explanation	Probable Cause	Solution	Log Flag
Read value beyond limits for Mod 10000 register: <Item Name> on Node: <PLC Name>. Value clamped to 9999.	The Modulo-10000 point value overflows. The value is clamped to 9999.	The maximum usable value represented in one Modulo-10000 registers is 9999.	Check the PLC for the value in the registers.	DASProtWarn
recv() failed with errorcode <Error Code>.	The PLC responded with an error.	The PLC may be down or there is a communications problem.	Check the PLC and the communications link.	DASProtFail
Register Type is 'BCD'. Value clamped to 9999.	The Modulo-10000 point output value overflows. The value is clamped.	The Modulo-10000 value written to the PLC exceeded the max limit.	The client must write a smaller value.	DASProtWarn
Register Type is hot-configured to <Register type>.	The Register Type parameter was modified at run time.	The user modified the Register Type parameter while the server was running.	The Register Type parameter can be modified at run time.	DASProtWarn
Reply Timeout is hot-configured to <Reply timeout value>.	The Reply Timeout parameter was modified at run time.	The user modified the Reply Timeout parameter while the server was running.	The Reply Timeout parameter can be modified at run time.	DASProtWarn
String Variable Style is hot-configured to <String variable style>.	The String Variable Style parameter was modified at run time.	The user modified the String Variable Style parameter while the server was running.	The String Variable Style parameter can be modified at run time.	DASProtWarn
The item name <PLC Name>. <Item Name> is Invalid. It is non-numeric.	The item name is invalid.	A non-numeric register was defined where numeric was required.	Make register a numeric register.	DASProtFail
The item name <PLC Name>. <Item Name> is invalid. The bit number is invalid.	The item name has an invalid bit number.	The client defined an item with an invalid bit number.	The item bit number must be between 1 (one) and 16.	DASProtFail
The item name <PLC Name>. <Item Name> is invalid. The bit number is out of range.	The item name has an invalid bit number.	The client defined an item with an invalid bit number.	The item bit number must be between 1 (one) and 16.	DASProtFail
The item name <PLC Name>. <Item Name> is invalid. The data type is invalid.	The server encountered an invalid data type.	The client defined a data type not supported by the server.	The client must use a valid data type.	DASProtFail

Error Message	Explanation	Probable Cause	Solution	Log Flag
The item name <PLC Name>.<Item Name> is invalid. The register range is invalid.	The Modulo-10000 item defined by the client exceeds the 3 register limit.	The client defined an invalid item name.	The client must use two or three registers when defining Modulo-10000 item names.	DASProtFail
The item name <PLC Name>.<Item Name> written string value <String Value> was truncated to <Truncated String Value>. Not enough registers were defined to hold the string value.	The server truncated the string being written to the PLC.	Not enough registers were defined by the client to hold the string data being written.	Add more registers or write less data.	DASProtFail
The PLC <PLC Name> message timed out (OnPLCReceiverTimeout), revoking message <Message ID>.	The server did not get a response from the PLC in the <ReplyTimeout> allotted time.	The PLC is offline or the data request has an invalid register number obtained from the configuration file.	Check PLC's network connection or configuration file.	DASProtFail
The PLC <PLC name> reported receiving an ILLEGAL FUNCTION code from the DAServer. PLC errorcode 01.	The function code received in the query is not an allowable action for the slave.	This may be because the function code is only applicable to newer controllers, and was not implemented in the unit selected. It could also indicate that the slave is in the wrong state to process a request of this type; for example, because it is unconfigured and is being asked to return register values.	Check the item definition and PLC configuration.	DASProtFail

<b>Error Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>	<b>Log Flag</b>
The PLC <PLC name> reported receiving an ILLEGAL DATA ADDRESS from the DAServer. PLC errorcode 02.	The data address received in the query is not an allowable address for the slave.	The item address is out of the configured-slave address range, or the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02.	Check the item definition and PLC configuration.	DASProtFail
The PLC <PLC name> reported receiving an ILLEGAL DATA VALUE from the DAServer. PLC errorcode 03.	A value contained in the query data field is not an allowable value for the slave.	This indicates a fault in the structure of the remainder of a complex request. For example, the implied length is incorrect.	Check the data value sent to the PLC.	DASProtFail
The PLC <PLC name> reported ILLEGAL RESPONSE LENGTH while working on the current request. PLC errorcode 04.	An unrecoverable error occurred while the slave was attempting to perform the requested action.	Indicates that the request as framed would generate a response whose size exceeds the available MODBUS data size.	Check the PLC.	DASProtFail
The PLC <PLC name> reported ACKNOWLEDGE while working on the current request. PLC errorcode 05.	The slave has accepted the request and is processing it, but a long duration of time will be required to do so.	This response is returned to prevent a timeout error from occurring in the client (or master).	Check the PLC.	DASProtFail
The PLC <PLC name> reported SLAVE DEVICE BUSY and cannot process the current request. PLC errorcode 06.	The slave is engaged in processing a long-duration program command.	The PLC program is taking too long to respond to the DAServer.	Check the PLC.	DASProtFail
The PLC <PLC name> reported NEGATIVE ACKNOWLEDGE and cannot process the current request. PLC errorcode 07.	The slave cannot perform the program function received in the query.	The communications data is corrupted or a DAServer problem is encountered.	Check the PLC.	DASProtFail



Error Message	Explanation	Probable Cause	Solution	Log Flag
The PLC <PLC name> reported MEMORY PARITY ERROR while attempting to read extended memory. PLC errorcode 08.	The slave attempted to read extended memory, but detected a parity error in the memory.	The PLC may need to be serviced.	Check the PLC's extended memory.	DASProtFail
The PLC <PLC name> reported GATEWAY PATH UNAVAILABLE while attempting to read extended memory. PLC errorcode 0A.	Specialized use in conjunction with gateways; it indicates that the gateway was unable to allocate a Modbus Plus PATH to use to process the request.	It usually means that the gateway is misconfigured.	Check the Bridge connected to the PLC.	DASProtFail
The PLC <PLC name> reported GATEWAY TARGET DEVICE FAILED TO RESPOND while attempting to read extended memory. PLC errorcode 0B.	Specialized use in conjunction with gateways; it indicates that no response was obtained from the target device.	It usually means that the device is not present on the network.	Check the Bridge connected to the PLC.	DASProtFail
The PLC <PLC name> reported unknown exception code. PLC errorcode <Error Number>.	The PLC returned an unknown error code.	The communications data is corrupted.	Check the PLC.	DASProtFail
The property RegisterSize value <Value> is invalid for PLC <PLC Name>.	The RegisterSize value in the configuration file is invalid.	The entry in the file was modified manually.	Use the DAServer Manager to rebuild the file.	DASProtFail
The register number for the item name <PLC name>.<Item Name> exceeds the configured PLC Maximum Addressable Registers.	The register defined by the client is out of range compared to the maximum addressable register number.	The client defined an invalid register number.	Client must use a register number less than or equal to the configured maximum addressable register.	DASProtFail
The socket is marked as nonblocking and the connection cannot be completed immediately for host <Host name>.	The connection to the PLC failed.	The PLC or the network may be having problems.	Check the PLC or the network.	DASProtFail
The write Item <Item Name> is invalid. The item is a read only item.	The register being written to is a read-only register.	The client is writing to a read-only register.	Select a different register.	DASProtFail

<b>Error Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>	<b>Log Flag</b>
Write attempt beyond limits for Mod 10000 register: <Item Name> on Node: <PLC Name>. Value clamped to <new value>.	The output value was clamped to a predefined value because the output value exceeded certain limits.	The client wrote a value to the PLC that was too large for the register.	The client must write a smaller value.	DASProtFail

## Server-Specific Error Codes

There are two server-specific error codes, shown in the following table, that augment those provided by the DAS Toolkit.

<b>Error Code</b>	<b>Logger Message</b>	<b>Log Flag</b>
-10001	PLC not connected	DASProtFail
-10002	PLC timeout	DASProtFail

## CHAPTER 5

# Reference

## Contents

- DAServer Architecture
- Component Environments

## DAServer Architecture

---

**Note** DAServers are supported on Microsoft Windows 2003, Windows 2000, Windows XP, and Windows XP Embedded only. NetDDE protocol is not supported by DAServers.

---

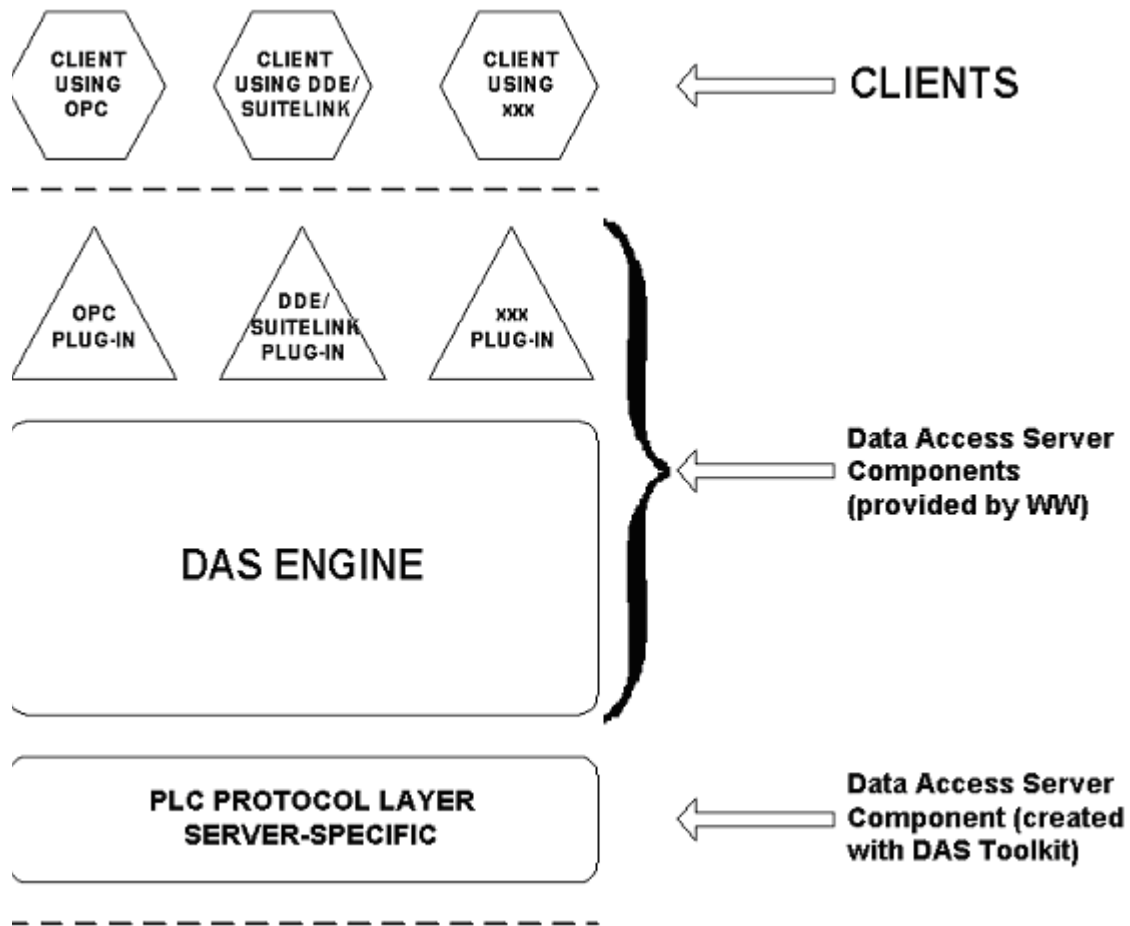
This DAServer is a collection of components that work in concert to provide communications access with hardware field devices. These components include:

- **DAServer Manager:** This is the Microsoft Management Console (MMC) snap-in, that is part of the ArchedrA System Management Console suite of utilities, supplied with the DAServer. It provides the necessary user-interface for diagnostics, configuration, and activation.
- **Client Plug-ins:** These are the components that are added to a DAServer to enable communications with clients. Examples are: OPC, DDE/Suitelink, and so on.
- **DAS Engine:** This is the library that contains all the common logic to drive data access.
- **Device Protocol:** This is the custom code provided by this DAServer to define the communications with a particular device.

## DAServers

A DAServer is comprised of three physical parts (see the following figure). They are the following:

- **Plug-in Component(s):** Responsible for communicating with clients.
- **DAS Engine:** This common component is used by all DAServers.
- **PLC Protocol Layer, DAServer-specific:** This component is responsible for communicating with the hardware.



### DAServer Architecture

Each physical part of a DAServer is comprised of a set of .exe and/or .dll modules. Wonderware provides the Plug-ins and the DAS Engine. The DAS Toolkit user creates the PLC Protocol Layer (DAServer-specific) modules. All three sets of modules are required for a fully functioning DAServer.

### Plug-ins

Plug-ins provide a protocol-translation function for device integration clients. Typical Plug-ins communicate in DDE, SuiteLink, or OPC protocol, and serve as interfaces between their clients and the DAS Engine.

**Note** Items of an array are not supported in the DDE/SL plug-in. These arrays are converted to HEX strings, which provide legacy behavior for DAServers that support this in the DAServer-specific code.

## DAS Engine

The DAS Engine is a middleware component that exposes two sets of unique interfaces, one for communicating with the Plug-ins and the other one for communicating with the PLC Protocol Layer components.

## PLC Protocol Layer

The PLC Protocol Layer provides a protocol-translation function for specific hardware, such as ModBus TCP/IP Ethernet; and it serves as an interface between the DAS Engine and the hardware.

---

**Note** Reads/writes of complete arrays will not be supported.

---

## Component Environments

Stand-alone DAServers have the following characteristics:

- The DAS Engine is dynamically linked to the other DAServer components. In other words, a new DAS Engine (feature enhancement or bug fix) would not require relinking to the other components nor re-QA of those other components. When deployed to the system, the new DAS Engine would attach to all existing DAServer components.
- Newly deployed Plug-ins (feature enhancements or bug fixes) do not require relinking nor re-QA of associated components. Even new Plug-ins (for example, OPC Alarm & Events) would not require any development changes to the other components, and therefore no relinking in a customer-installed base. In fact, it is feasible to implement new functionality in a Plug-in to enhance the DAServer without any involvement of the code of the other components.
- DAServers can be configured in one stand-alone configuration utility (DAServer Manager), and the DAServer Manager is capable of displaying specific configuration views for all DAServers. This utility allows the browsing and editing of DAServers on different nodes.
- The DAServer Manager diagnostics tool displays generic diagnostic objects common to all DAServers, in addition to the DAServer-specific/DAServer-developer-defined diagnostic data.

The DAServer data configuration format is XML. Any XML-enabled program (for example, XML Editor) can read this format.



## A P P E N D I X A

# Supported DASMBTCP Hardware and Firmware

The following table lists the Modbus hardware and firmware supported by the MBTCP DAServer.

Legend	Model	Description	Firmware
TSX Quantum	140 CPU 21304	Controller	Exec ID 0871 HID 0405 Rev 02.10
	140 NOE 771 00	Ethernet TCP/IP Option Module	
	140 NOE 21100	Ethernet TCP/IP Module	
TSX Momentum		Controller	
	170 ENT 110 01	Ethernet TCP/IP Module	
TSX Premium	TSX P57	Controller	
	TSX ETY 110	EtherNet TCP/IP Module	
TSX Momentum (RS232)		Controller	
	171 CCC 980 20	Ethernet TCP/IP Module and Modbus Ports	
TSX Momentum (RS485)		Controller	
	171 CCC 980 20	Ethernet TCP/IP Module and Modbus Ports	
Compact 984 (RS232, 5-Digit Addressing)	PC-A984-145	Controller	Exec ID 084E Model A145
Compact 984 (RS232, 6-Digit Addressing)	PC-E984-265	Controller	
Modicon Micro (RS232)	110 CPU 311 01	Controller	Exec ID 0863 HID 0701 Rev 01.10 Model 311/01

<b>Legend</b>	<b>Model</b>	<b>Description</b>	<b>Firmware</b>
Generic Modbus Controller (4-Digit, 5-Digit, 6-Digit)		Generic Modbus Controller. Controller must conform to the Modbus "Application Protocol specifications."	
NR&D Bridge	Pen-T	Ethernet-to-Serial Bridge	
Modbus Bridge	174CEV30010 174CEV30020	Bridge	



## A P P E N D I X B

# The Modbus Protocol

This appendix describes the Modbus codes, TCP Port, and Data Types that are supported by the MBTCP DAServer.

## Contents

- Controller Function Codes
- Modbus Exception Codes
- TCP Port
- Data Types

## Controller Function Codes

The MBTCP DAServer uses function codes to communicate with the various controllers supporting the Modbus protocol. The implementation for the MBTCP DAServer uses the document, "Open Modbus/TCP Specification," Release 1.1, dated December 6, 2002 as a reference.

These function codes and their descriptions are listed in the following table.

---

**Note** In communicating with the Generic Modbus PLCs, the MBTCP DAServer acts as a master and sends out the function codes to the PLCs.

---

Code	Name	Description
01 (0x01)	Read Coils	Reads the ON/OFF status of discrete outputs (0X references, coils) in the slave.
02 (0x02)	Read Discrete Inputs	Reads the ON/OFF status of discrete inputs (1XXXXX references) in the slave.
03 (0x03)	Read Holding Registers	Reads the binary contents of holding registers (4XXXXX references) in the slave.
04 (0x04)	Read Input Registers	Reads the binary contents of input registers (3XXXXX references) in the slave.

Code	Name	Description
05 (0x05)	Write Single Coil	Forces a single coil (0X reference) to either ON or OFF.
06 (0x06)	Write Single Register	Presets a value into a single holding register (4XXXXXX reference).
15 (0x0F)	Write Multiple Coils	Forces each coil (0XXXXXX reference) in a sequence of coils to either ON or OFF.
16 (0x10)	Write Multiple Registers	Presets values into a sequence of holding registers (4XXXXXX references).
20 (0x14)	Read General Reference	Returns the contents of registers in the Extended Memory file (6XXXXXX) references.
21 (0x15)	Write General Reference	Writes the contents of registers in the Extended Memory file (6XXXXXX) references.
22 (0x16)	Mask Write Holding Register	Modifies the contents of a specified (4XXXXXX reference) holding register using a combination of an AND mask, an OR mask, and the register's current contents. The function can be used to set or clear individual bits in the register.

## Modbus Exception Codes

Additionally, there are exception codes generated by Modbus. The accompanying table shows these exceptions and their explanations.

Exception Code (Hex)	Name	Explanation
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave. This may be because the function code is only applicable to newer controllers, and was not implemented in the unit selected. It could also indicate that the slave is in the wrong state to process a request of this type; for example, because it is unconfigured and is being asked to return register values.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02.

Exception Code (Hex)	Name	Explanation
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04	ILLEGAL RESPONSE LENGTH	Indicates that the request as framed would generate a response whose size exceeds the available MODBUS data size. Used only by functions generating a multi-part response, such as functions 20 and 21.
05	ACKNOWLEDGE	Specialized use in conjunction with programming commands.
06	SLAVE DEVICE BUSY	Specialized use in conjunction with programming commands.
07	NEGATIVE ACKNOWLEDGE	Specialized use in conjunction with programming commands.
08	MEMORY PARITY ERROR	Specialized use in conjunction with function codes 20 and 21 to indicate that the extended file area failed to pass a consistency check.
0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways; it indicates that the gateway was unable to allocate a Modbus Plus PATH to use to process the request. It usually means that the gateway is misconfigured.
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Specialized use in conjunction with gateways; it indicates that no response was obtained from the target device. It usually means that the device is not present on the network.

## TCP Port

The MBTCP DAServer uses port 502 as the default to contact all PLCs. This includes the port used for unsolicited messages from the PLCs that are programmed to provide unsolicited data (that is, the PLCs that are directly connected to the TCPIP\_PORT; controllers under the ModbusBridge Object hierarchy cannot utilize this unsolicited messaging feature).

However, you have the option to configure the actual port to be used through the ModbusPLC object. When a non-502 port is used by the Generic Modbus PLC to communicate with the MBTCP DAServer, it will not be able to send unsolicited data to the DAServer.

## Data Types

The MBTCP DAServer supports the following data types:

- Boolean
- 16-bit signed integer
- 16-bit unsigned integer
- 32-bit signed integer
- 32-bit unsigned integer
- ASCII string
- 32-bit single precision floating point

---

**Note** For detailed information on these data types, see Data and Register Types in Chapter 3.

---

# Index

## Symbols

\$\$Y\$\$ActiveItemCount 82  
 \$\$Y\$\$ErrorCode 80  
 \$\$Y\$\$ErrorCount 82  
 \$\$Y\$\$ErrorText 80  
 \$\$Y\$\$ItemCount 82  
 \$\$Y\$\$Licensed 15, 79  
 \$\$Y\$\$MaxInterval 81  
 \$\$Y\$\$PollNow 82  
 \$\$Y\$\$ReadComplete 82  
 \$\$Y\$\$Status 78, 79, 85  
 \$\$Y\$\$StoreSettings 80  
 \$\$Y\$\$UpdateInterval 81  
 \$\$Y\$\$WriteComplete 81  
 .csv file 62  
 .csv file in Excel 65

## A

Absolute Notation Naming Convention 75  
 Access Path 83  
 Activate Server 22  
 Actual PLC item names 62  
 Add command 63  
 Add Compaq984 Object 19  
 Add item references 63  
 Add ModbusBridge Object 18  
 Add ModbusBridge objects 24  
 Add ModbusPLC Object 18  
 Add ModbusPLC objects 37  
 Add ModbusPLCRS Object 19  
 Add ModiconMicro objects 46  
 Add TCPIP\_PORT Object. 18  
 Add TSXMomentum Object 18, 19  
 Add TSXMomentum objects 29  
 Add TSXMomentum-RS Object 19  
 Add TSXMomentumRS objects 49  
 Add TSXPremium Object 18  
 Add TSXPremiumPLC objects 33  
 Add TSXQuantum Object 18  
 Add TSXQuantum objects 25  
 AddModiconMicro Object 19  
 Alarm event 68  
 Alias names 62  
 Aliases 62  
 Application name 14  
 ArchestrA System Management Console 19  
 ArchestrA.DASMBTCP.1 18, 19, 20, 22  
 Archive Configuration Set 68  
 Archive configuration sets 68  
 Archiving a Configuration Set 68  
 Auto service 19, 22

## B

Before 7  
 Bit order format 27, 31, 35, 40, 43, 47, 51, 55  
 Block I/O size 28, 32, 36, 41, 44, 48, 56  
 Block I/O sizes 52  
 Boolean item 15

Bridge type 24

## C

C style 40, 56  
 Clear All command 64, 66  
 Clear all device items 64  
 Coil 28, 32, 36, 44, 48, 52  
 Coil write 28, 32, 36, 41, 45, 49, 52, 56  
 Communication Protocols 10  
 Compact 984 controllers 13  
 Compact 984 PLC 71  
 Config Default Update Interval 61  
 Configuration 17  
 Configuration node 69  
 Configuration set 60, 64  
 Configuration Set Name 68  
 Configure default update intervals 61  
 Configure the DAServer to receive unsolicited messages 60  
 Configuring the DAServer 19  
 Create a TCPIP\_PORT object 23  
 Create or add device groups 59  
 Create or add device items 63  
 CSV file 64

## D

DAS Engine 99  
 DAS Toolkit 100  
 DAServer Manager 17, 19, 20, 99, 101  
 DAServer Manager book 21  
 DAServer Manager documentation 18, 19, 21, 22, 57  
 DAServer Manager tree 18, 21, 22  
 DAServer version 85  
 Data Access Server 10  
 Data and Register Types 71  
 Data Type 72  
 DCOM 11  
 DDE 11  
 DDE communications protocol 14  
 DDEStatus 86  
 Default Group 20  
 Default update interval 61  
 Delete a device item from the list 64  
 Delete command 64  
 Delete device groups 60  
 Demo Mode 15  
 Device group 14, 78  
 Device Group Definitions 59  
 Device Group Parameters dialog box 61  
 Device Groups 17, 19, 21, 22  
 Device Groups dialog box 22, 59  
 Device Groups tab 21, 59, 69  
 Device groups. 69  
 Device hierarchy 17, 21  
 Device Item Definitions 22, 59, 62  
 Device Items 19  
 Device Items box 66  
 Device Items configuration 64  
 Device Items dialog box 22, 62, 63, 64, 66

Device Items tab 22, 62, 69  
 Device nodes 69  
 DI (Discrete Input) 75  
 Diagnostics 11  
 Direct Connectivity 12  
 Discrete input 28, 31, 35, 44, 48, 51  
 Discrete input/Coil read 32, 36  
 Discrete input/coil read 28, 41, 45, 48, 52, 56  
 Distributed COM 11  
 DO (Discrete Output) 75  
 Dynamic Data Exchange 11

## E

Edit mode 19  
 error messages 88  
 Export command 65  
 Export DAServer device item data 64  
 Export DAServer device item data to a CSV file 64  
 Export feature 64  
 Extended register 28, 32, 44, 52  
 Extended register number 74  
 Extended register read 29, 33, 45, 53  
 Extended register write 29, 33, 45, 53

## F

F (floating) 76  
 FastDDE 11  
 Features 11, 14  
 Full length 40, 56

## G

Generic Modbus (RS232/RS485) PLC 71  
 Generic Modbus controllers 12  
 Generic Modbus PLC 71  
 Generic Modbus Serial controllers 13  
 Getting Started Quickly 17  
 Global Configuration parameters 69  
 Global Parameters 21  
 Group 78  
 Group name 13

## H

Help menu 21  
 Hierarchical location 78  
 Hierarchy 22  
 HMI 11  
 Holding register 28, 32, 36, 48, 52  
 Holding register read 29, 32, 36, 45, 49, 52  
 Holding register read/write 44  
 Holding register write 29, 33, 36, 45, 49, 53  
 Host name 25, 34  
 Hot configurable 69  
 Hot Configuration 69  
 HR (Holding Register) 76

## I

import 66  
 Import command 66  
 Import DAServer device item data 64

Import DAServer device item data from a CSV file 66  
 Import feature 64  
 Import off-line-edited DAServer device item data from a CSV file 66  
 Indirect Connectivity 13  
 Input register 28, 32, 36, 44, 48, 52  
 Input register read 29, 33, 37, 45, 49, 53  
 InTouch WindowViewer 86  
 IOStatus 86  
 IP address 25, 34  
 IR (Input Register) 75  
 Item ID 83  
 Item Name 83  
 Item name 14  
 Item Names 71  
 Item Reference column 62, 63, 65  
 ItemID 13

## L

L (long) 76  
 Large-scale edit on item data 64  
 License Manager 15  
 Link name 14, 83  
 Local node 20  
 Log Flag data 88  
 Log Viewer 88  
 Logger 69

## M

Make change on device groups 60  
 Make changes on update intervals 61  
 Manual or automatic service 22  
 Manual service 19, 22  
 Maximum address range 44  
 Maximum address range 27, 31, 35, 48, 51  
 Maximum outstanding messages 25, 26, 30, 34, 39  
 MBTCP hierarchy 21  
 MBTCP Hierarchy in the DAServer Manager 21  
 Message Optimization 59  
 Microsoft Management Console 19  
 Microsoft Windows 2000 99  
 Microsoft Windows 2003 99  
 Microsoft Windows XP 99  
 Microsoft Windows XP Embedded 99  
 MMC 19  
 Modbus 12  
 Modbus Bridge 10, 13, 25  
 Modbus Generic 4-Digit PLCs 40, 56  
 Modbus Generic 5-Digit PLCs 40, 56  
 Modbus Generic 6-Digit PLCs 40, 56  
 Modbus Item Naming 72  
 Modbus Protocol Reference Guide 12, 20  
 Modbus TCP/IP Ethernet Protocol 12  
 Modbus TPC/IP Ethernet protocol 10  
 Modbus-family controllers 72  
 Modicon Micro (RS232) PLC 71  
 Modicon Micro controllers 13  
 Modicon PLC Naming Convention 74  
 Modicon PLCs 72  
 ModiconPLC 14  
 Modulo-10000 Points Naming Convention 76

## N

Name column 60, 63, 65  
 NetDDE 11, 12  
 NetDDE protocol 99  
 Network address 25, 26, 30, 34, 38  
 Network transport protocol 11  
 New\_Compact984\_000 42  
 New\_Compact984\_000 Parameters 42  
 New\_ModbusBridge\_000 24  
 New\_ModbusBridge\_000 Parameters 24  
 New\_ModbusPLC\_000 37  
 New\_ModbusPLC\_000 Parameters 37  
 New\_ModbusPLCRS\_000 53  
 New\_ModbusPLCRS\_000 Parameters 53  
 New\_ModiconMicro\_000 46  
 New\_ModiconMicro\_000 Parameters 46  
 New\_ModiconMicroPLC\_000 49  
 New\_PLC\_000 Parameters 59, 62  
 New\_TCPIP\_Port\_000 18  
 New\_TCPIP\_PORT\_000 Parameters 23  
 New\_TSXMomentum\_000 29  
 New\_TSXMomentum\_000 Parameters 30  
 New\_TSXMomentumRS\_000 Parameters 49  
 New\_TSXPremium\_000 Parameters 33  
 New\_TSXPremiumPLC\_000 33  
 New\_TSXQuantum\_000 Parameters 26  
 New\_TSXQuantumPLC\_000 25  
 Nmae column 62  
 Node name 13, 14  
 NR&D Pen-T Bridge 10, 13  
 NRD Pen-T Bridge 25

## O

Off-line edit on item data 64  
 OLE for Process Control 11  
 OLE/COM technology 11  
 OPC 11  
 OPC client 63  
 OPC communications protocol 13  
 OPC Item browsing 63  
 OPC protocol 13  
 Open button 66  
 Open Modbus/TCP Specification 105

## P

Pascal style 40, 56  
 PLC 71  
 PLC disconnect 86  
 PLC hierarchy node 63  
 PLC Hierarchyname.csv 65  
 PLC Protocol Layer 99, 100  
 PLC register types 72  
 PLC Unit ID 50  
 PLC unit ID 42, 47, 54  
 Plug-in Component(s) 99  
 Plug-ins 101  
 Point Naming Convention 73  
 POKE command 87  
 Poke Mode settings 21  
 Port number 23, 38  
 Prepare the MBTCP DAsServer 18, 20

ProgID 13  
 Program name 13  
 PV (Process Variable) 76

## R

Reference 10  
 reference 99  
 Register number 71, 73  
 Register read 41, 56  
 Register size (digits) 40, 43, 56  
 Register type 27, 31, 35, 40, 43, 48, 51, 56, 71  
 Register write 41, 57  
 Remote Reference formulas for cells 88  
 Rename a device item from the list 64  
 Rename command 64  
 Reply time-out 86  
 Reply timeout 26, 30, 34, 38, 42, 47, 50, 54  
 RS232 serial cable 10  
 RS485 serial cable 10

## S

S (signed) 76  
 SCADA 11  
 Scan Based Message Handling 67  
 Scan-Based Message Handling 67  
 Service 17, 19, 22  
 Setup.exe 18, 20  
 Slow poll mode 86  
 SMC 19  
 Snap-in environment 18, 21  
 String variable style 27, 31, 35, 40, 43, 47, 51, 56  
 SuiteLink 11  
 SuiteLink communications protocol 14  
 Support multiple coil write 39, 55  
 Support multiple register write 39, 55  
 Support Unsolicited Messages 61  
 System Management Console 17, 18, 20

## T

Target PLC 83  
 TCP/IP Ethernet protocol 13  
 TCP/IP port 13  
 The file can be edited off-line 65  
 Topic name 14, 71  
 TSX Momentum (RS232/RS485) PLC 71  
 TSX Momentum controllers 12, 13  
 TSX Momentum PLC 71  
 TSX Premium controllers 12  
 TSX Premium PLC 71  
 TSX Quantum controllers 12  
 TSX Quantum PLC 71

## U

U (unsigned) 76  
 UNIX 12  
 Unsolicited Message Handling 68  
 Update Interval 67  
 Update Interval column 61  
 Use Another Configuration Set 69  
 Use Concept data structures (Longs) 27, 31, 34, 39, 43, 47, 50, 54

Use Concept data structures (Reals) 27, 31, 35, 39,  
43, 47, 51, 55  
Use different configuration sets 68

## V

Value Time Quality (VTQ) 11  
VMS 12

## W

WinSock 11  
Wonderware folder 18  
Wonderware InTouch 10