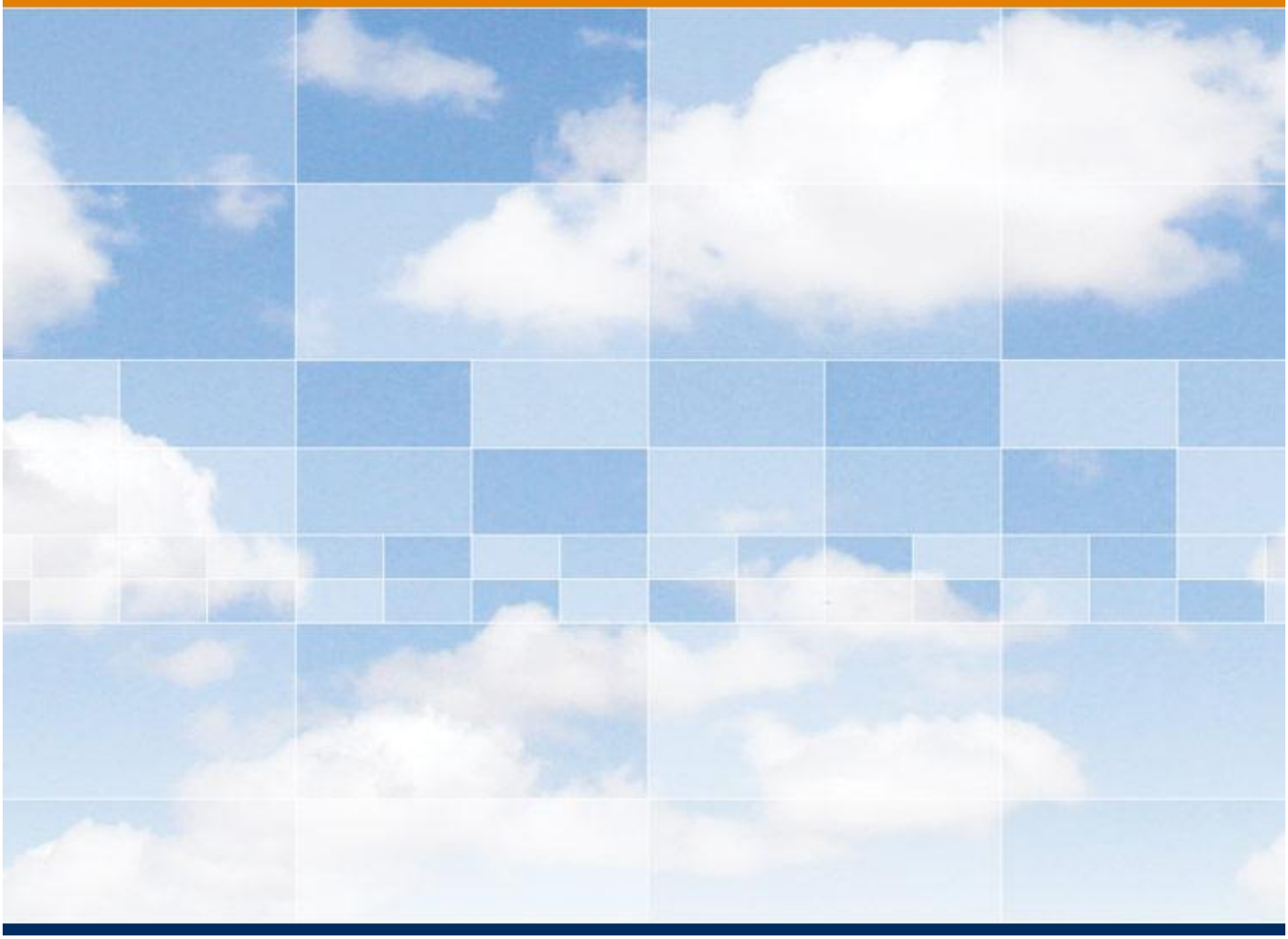
MatrikonOPC



MatrikonOPC Server for SCADA Modbus

User's Manual

MatrikonOPC Server for SCADA Modbus
User's Manual

This manual is a product of Matrikon International.

Matrikon International
Suite 1800, 10405 Jasper Avenue
Edmonton, AB T5J 3N4
Canada

Phone: +1.780.448.1010
Fax: +1.780.448.9191
www.matrikonopc.com

Document Revision History:

Date	Document Version	Description	Author
2002-01-16	1.0	Initial document.	RE
2003-01-01 2006-06-19 2008-01-17 2008-03-10	1.1 – 1.4	Converted to new format (master/source). Adapted to new document template. Updated with new option information. General update of manual.	DC, CD, JY, MH
2008-03-18	2.0	Converted to new template, general edit.	LB
2008-10-03	2.1	Added new option information	MNA
2008-12-11	3.0	Added Appendix E – Security, Appendix F – Offline Mode, Enter Password screen in Installation section, files to SCADA Modbus folder (Table 2) in Installed Files section. Replaced Configuration screen graphic to show addition of View Tags and Security Settings icons, added descriptions of icons to Tools menu. Added Getting Data without Being Connected to a Device topic to OPC Data Items section.	LB
2008-1-06	4.0	Added Appendix G - SCADA Servers Connection Logic, Appendix H - Features and Examples. Added Slave emulation mode and new options information. Added Enter Password screenshot with default password. Replaced Configuration window screenshot to show View Tags and Security Settings icons on toolbar. Added Redundancy Device Link OPC Items table to the OPC Data Items section. Updated Appendix E – Security.	MNA, WD, LB

2009-03-09	4.1	OPC Data Items section, sub-heading Slave Serial Device Syntax re-named as Slave Device Syntax.	LB
2009-04-02	4.2	Updated product version.	JDJ
2009-04-14	5.0	Updated field S description in Table 35 – Available Items. Updated Create New Modbus Unit – Communication tab – Write optimism field description (Table 27). Added Limitations section.	JDJ
2009-05-14	5.1	Updated product version. Modified Schedule All Polls (Tables 21, 24, 64) to note the new default of checked. Updated Contacting Support section. Updated DCOM Manual link. Added Redundancy Group statistical items issue to the Troubleshooting section.	JDJ
2009-06-04	5.2	Updated Common folder (Table 3) in Installed Files section.	ZB
2009-06-25	5.3	Added note to Creating and Configuring a Redundancy Group Object section. Updated Contacting Support section.	JDJ
2009-07-17	5.4	Updated version to 1.1.3.0.	JFY
2009-08-12	6.0	Updated software version to 1.1.4.0. Notes under OPC Data Items – SCADA Modbus Items updated.	JFY, LB
2009-08-14	6.1	Note added to Load Distribution Tab.	ZB, LB
2009-08-20	7.0	Applied latest OPC Framework v1.11.1.0 (Outer Layer)/1.7.1.0 (Inner Layer). Updated Installed Files section. Updated Destination Folder and MatrikonOPC Server for SCADA Modbus Setup Complete screenshots. Updated General Options tab screenshot. Added Advanced Options section. Added Writes field to Create New Network Channel and Create New Serial Channel (Data Acquisition tab) screenshots and components descriptions. Added Ping Unit to Create New Modbus Unit (Communication tab) screenshot and components description. Alias Configuration section updated. Item Path Browser sub-section added. Appendix B – Aliases updated including addition of Scaling Calculations.	LB, ZB
2009-11-27	8.0	Added Creating and Configuring a Redundancy Group Object.	MM, LB
2009-12-07	8.1	Added description and screenshot for Discard Unsolicited Packets feature in the host device link. Added statement to Software Requirements about Windows Server 2008 support. Modified OPC Items section to include information about the 7 and 8 registers, and the R and T item suffixes.	DOC, LB
2009-12-17	8.2	Updated Table 43.	MM

2010-04-23	9.0	Converted manual to plug-in standard.	LB
2010-05-27	10.0	Converted manual to stand-alone server standard. Replaced Create New Network Host Object screenshot in Figure 11 and updated Table 11. Updated Contacting Support section.	LB
2010-06-01	10.1	Added clarification for 'Writes' option in Network and Serial device links.	CHB
2010-06-04	11.0	Upgraded to latest OPC Framework version. Converted user documentation to standard template.	LB
2010-06-08	11.1	Corrected installed files listing.	CHB
2010-06-15	12.0	Updated software version to 1.2.2. Clarified use of #Redundancy Group data item in OPC Data items section.	CHB, LB
2010-07-21	13.0	Software updated to version 1.2.3. Added description for #Interrogate data item to Appendix D – OPC Items. Clarified user interaction for those new device links added to Redundancy Groups. Clarified the repercussions of using the Stay Connected option in the Modbus Unit device link in the Troubleshooting section.	CHB, LB
2010-07-27	13.1	Updated Table 38 in OPC Data Items section. Updated Limitations section. Updated Appendix I - Installed Files.	CHB, LB
2011-02-25	13.2	Updated software version to 1.2.4.	LB
2011-04-12	14.0	Updated software version to 1.3.0. Updated Figure 35 and Table 30 to include CPU Throttle and CPU Sleep.	REK, LB
2012-06-13	15.0	Updated software version to 1.3.1. Updated Copyright Information and after-hours Support contact information.	LB
2012-07-11	16.0	Updated software version to 1.3.2. Updated Function Code 5 and typo in Function Code 16 in Table 44. Updated Max. write length in Table 29. Updated Host name/IP in Table 22. Updated Troubleshooting section.	LB
2012-07-25	16.1	General user documentation fixes. Updated Software Requirements, References, OPC Compliance sections, and Appendix I – Installed Files.	LB

2013-06-25	16.2	Updated software version to 1.3.3. Added SCADA Communication section. Updated Appendix H with Multiple instances on step-9. Updated Table-19 for Device Link Name description. Updated Table-22 for Reconnect Delay description. Updated the Table-67 text for SysWOW32 folder.	MK
2014-12-05	17.0	Updated Software version to 2.0.0. Updated Contacting support, Software and Hardware requirements, Limitations,Starting the MatrikonOPC server for SCADA Modbus, General Logging Options . Added Appendix F – Item Quality.	AC
2015-02-11	17.1	Updated Software version to 2.0.1.	AC

SOFTWARE VERSION

Version: 2.0.1

DOCUMENT VERSION

Version: 17.1

COPYRIGHT INFORMATION

CONFIDENTIAL

IMPORTANT NOTICE

Although every endeavor has been made to ensure that the information contained within this document is up to date and accurate, Matrikon cannot be held responsible for any inaccuracy or error in the information contained within this document. Matrikon makes no warranty of any kind with regard to the information contained within this document and Matrikon shall not be liable for any direct, indirect, incidental or consequential damages which may arise in connection with the furnishing, reliance, or use of the information contained within this document.

Specifications and statements as to performance in this document are Matrikon estimates, intended for general guidance. Matrikon reserves the right to change the information contained within this document and any product specification without notice.

Statements in this document are not part of a contract or program product licence insofar as they are incorporated into a contract or licence by express preference. Issue of this document does not entitle the recipient to access or use of the products described, and such access or use shall be subject to separate contracts or licenses.

The receiving party shall not disclose, publish, report, communicate, or otherwise transfer any information in this document to any third party, and shall protect all information contained herein from unauthorized disclosure. The receiving party shall permit access to this document only to its employees, agents, subcontractors, and affiliates who reasonably require access to such information contained herein, have been made aware of the confidential nature of this document and have executed a written employment or other confidentiality agreement party to maintain the confidential status of this document.

LICENSE AGREEMENT

TRADEMARK INFORMATION

Table of Contents

Introduction **14**
Who Should Use This Manual 14
Overview of Manual 14
References 15
Document Terminology 16
Getting Started **17**
System Requirements 17
Software Requirements *17*
Hardware Requirements *17*
Configuration **19**
Starting the MatrikonOPC Server for SCADA Modbus 19
Tool Tray Menu 19
Control Panel 20
Configuration Window 21
File Menu *23*
Edit Menu *23*
View Menu *24*
Tools Menu *24*
Help Menu *25*
Options Window 25
General Options *25*
General Logging Options *27*
Interface Logging Options *28*
Advanced Options *29*
Server Configuration 31
Creating Objects *31*
Creating and Configuring a Fail-Over Channel Object *33*
Creating and Configuring a Network Channel Object *38*
Creating and Configuring a Redundancy Group Object *41*
Creating and Configuring a Serial Channel Object *45*
Creating and Configuring a Network Host Object *50*
Creating and Configuring a Server Status List Object *53*
Changing Objects *56*
Removing Objects *57*
Creating and Configuring a Dial-Up Connection Object *57*
Creating and Configuring a Modbus Unit Link *63*
Creating and Configuring a Radio Link *71*
Changing Objects *73*
Removing Objects *73*
Alias Configuration 74
Inserting Alias Groups *74*
Renaming Alias Groups *74*
Inserting New Aliases *74*
Editing Aliases *77*
Removing Aliases *77*
Removing Alias Groups *78*
Exporting Aliases *78*
Importing Aliases *78*
Configure Alias Scaling *78*
Configure Alias Events *80*

Configure Categories *82*
Expression Wizard 83
Examples *83*
Item Path Browser 87
Saving a Configuration 88
Clearing a Configuration 88
Loading an Existing Configuration 88
Shutting Down the Server 89
OPC Data Items 90
SCADA Modbus Items 90
Slave Device Syntax *93*
Examples *94*
SCADA Modbus Access Paths 94
Access Path Descriptions *94*
Getting Data without Being Connected to a Device 95
SCADA Communication 96
Channels 96
Serial *96*
Radio *96*
Network *96*
Dial-Up *96*
Poll Optimization 97
Transactions 97
Continuous *97*
Scheduled *98*
Instantaneous *99*
Integrity 99
Transmission *99*
Reception *99*
Failure and Recovery *100*
Redundancy and Fail-Over *100*
MatrikonOPC Server for SCADA Modbus Implementation 104
Data Types 104
Available Item Types 105
Diagnostics 106
Logging 106
Limitations 107
Troubleshooting 108
Problems and Solutions 108
Questions and Answers 116
Licensing 117
Contacting Support 118
OPC Compliance 119
Installation 119
Common Interfaces 119
Data Access 119
Groups *120*
Items *120*
Browsing and Item Properties *120*
I/O Operations *120*
Performance Report 122

Table of Appendices

Appendix A Distributed COM (DCOM) 123
DCOM Configuration Utility 123
Default Properties 123
Security Permissions 124
Server Identity 127
Default Protocols 127
Remote Program ID 128
Appendix B Aliases 129
Scaling Expressions 129
Advanced Calculations 130
CSV File Format 132
Scaling Algorithms 134
Appendix C Standard Data Types 135
Appendix D Microsoft Excel as a DDE Client 136
DDE Read 136
DDE Write 136
Net DDE 137
Appendix E Configuring Dynamic Data Exchange (DDE) 138
Appendix F Item Quality 139
Appendix G Security 140
Introduction 140
OPC Security Implementation 140
MatrikonOPC Tag Security Configuration Utility 141
Security Details 163
Appendix H Offline Mode 166
Introduction 166
Offline Mode Toggling 166
Offline Mode Item Values 166
Operational Effects 167
Appendix I Installation 168
Appendix J Installed Files 179
Appendix K Un-Installation 182
Appendix L SCADA Servers Connection Logic 187
Introduction 187
Polling 187
Connection Failures and Reconnection Logic 187
Appendix M Features and Examples 188
Introduction 188
Round-Robin Example 188
Scanning and *Stay Connected* Option 191
writeLimit OPC Item 191
PacketSent and *PacketReceived* OPC Items 191
Total Reads and Writes Indicators 192
Appendix N OPC Items 194
Introduction 194
Base OPC Items 194
Generic OPC Items 194
Network Channel Object OPC Items 196
Network Host Object OPC Items 196
Modbus Unit Object OPC Items 196

Fail-Over Channel Object OPC Items 199
Serial Channel Object OPC Items 199
Radio Object OPC Items 201
Dial-Up Object OPC Items 202

Table of Figures

Figure 1 - Tool Tray 19
Figure 2 - Tool Tray Menu 20
Figure 3 - Control Panel 20
Figure 4 - Configuration Window 22
Figure 5 - General Options Tab 26
Figure 6 - General Logging Options Tab 27
Figure 7 - Interface Logging Options Tab 29
Figure 8 - Advanced Options Tab 30
Figure 9 - Insert New Object Window 31
Figure 10 - Create New Fail-Over Channel Window 34
Figure 11 - Fail-Over Channel - Communication Tab 35
Figure 12 - Fail-Over Channel - Load Distribution Tab 36
Figure 13 - New Fail-over Channel Object 38
Figure 14 - Create New Network Channel Window 39
Figure 15 - New Network Channel Object 41
Figure 16 - Create New Redundancy Group Window 43
Figure 17 - New Redundancy Group Object 45
Figure 18 - Create New Serial Channel Window - Communication Tab 46
Figure 19 - Create New Serial Channel Window - Data Acquisition Tab 48
Figure 20 - New Serial Channel Object 50
Figure 21 - Create New Network Host Window 51
Figure 22 - Insert New Object 52
Figure 23 - New Network Host Object 53
Figure 24 - Create New Server Status List Window 54
Figure 25 - New Server Status List Object 56
Figure 26 - Client Reference Warning Message 57
Figure 27 - Create New Dial-Up Connection Window - Dial-Up Tab 58
Figure 28 - Create New Dial-Up Connection Window - Modem Tab 60
Figure 29 - Create New Dial-Up Connection Window - Advanced Tab 61
Figure 30 - Insert New Object Window 62
Figure 31 - New Dial-Up Connection Object 63
Figure 32 - Create New Modbus Unit Window - Communication Tab 64
Figure 33 - Slave Mode Options 66
Figure 34 - Create New Modbus Unit Window - Data Acquisition Tab 67
Figure 35 - Create New Modbus Unit Window - Optimization Tab 69
Figure 36 - Create New Radio Window 71
Figure 37 - New Radio Link 73
Figure 38 - Client Reference Warning Message 73
Figure 39 - Insert New Alias Window – Alias Type 75
Figure 40 - Contents Table 77
Figure 41 - Configure Scaling Window 79
Figure 42 - Generate Events Options 81
Figure 43 - New Category Window 82
Figure 44 - Edit Categories Window 83
Figure 45 - Expression Engine Configuration 85

Figure 46 - Expression Wizard 85
Figure 47 - Select Function Window 86
Figure 48 - Select Variable Window 86
Figure 49 - User Defined Functions Editor 87
Figure 50 - Operators Buttons 87
Figure 51 - Server Configuration Scenario 100
Figure 52 - F0V1 Server Configuration Scenario 101
Figure 53 - Fail-Over Channel Settings for F0V1 101
Figure 54 - Registry Editor 109
Figure 55 - Program ID Branch 110
Figure 56 - Distributed COM Configuration Properties Window 124
Figure 57 - Distributed COM Configuration Security Tab 125
Figure 58 - Access Permissions Window 126
Figure 59 - Add Users and Groups Window 126
Figure 60 - DDE Share 137
Figure 61 - Start Menu Navigation 142
Figure 62 - Enter Server Password Window 142
Figure 63 - MatrikonOPC Tag Security Configuration Utility Main Screen 143
Figure 64 - MatrikonOPC Tag Security Configuration Utility Main Screen: Users & Groups 144
Figure 65 - Add User Window (NT) 145
Figure 66 - Defined Users and Groups 146
Figure 67 - Add User Window (Private) 147
Figure 68 - Add Group Window (NT) 148
Figure 69 - Add Group Window (Private) 149
Figure 70 - Enter New Name Window 149
Figure 71 - Edit <Username> Window (Permissions Tab) 151
Figure 72 - Add Permission Screen (Server Address Space) 152
Figure 73 - Add User Permission Window 153
Figure 74 - Add Permission Window: Browsing Server Address Space 154
Figure 75 - Edit <Username>: Example 155
Figure 76 - Edit <Username> Window (General Tab) 156
Figure 77 - Reset Password Window 156
Figure 78 - Edit <Groupname> Window (Permissions Tab) 157
Figure 79 - Edit <Groupname> Window (General Tab) 158
Figure 80 - Edit <Groupname> Window (Members Tab) 159
Figure 81 - Add Existing User Window 160
Figure 82 - Change Password Window 161
Figure 83 - InstallAware Wizard Verification Window 168
Figure 84 - Pre-Requisites Screen 169
Figure 85 - License Agreement Screen 170
Figure 86 - Setup Type Screen 171
Figure 87 - Destination Folder Screen 172
Figure 88 - Start Menu Screen 173
Figure 89 - Enter Password Screen 174
Figure 90 - Licensing Screen 175
Figure 91 - Ready to Install Screen 176
Figure 92 - Installing MatrikonOPC Server Screen 177
Figure 93 - MatrikonOPC Server Setup Complete Screen 178
Figure 94 - Add or Remove Programs 182
Figure 95 - Welcome to MatrikonOPC Server for SCADA Modbus Maintenance Screen . 183
Figure 96 - Ready to Uninstall Screen 184
Figure 97 - Uninstalling MatrikonOPC Server for SCADA Modbus Screen 185

Figure 98 - MatrikonOPC Server for SCADA Modbus Setup Complete Screen 186
Figure 99 - Example 187
Figure 100 - Example 1 188
Figure 101 - Unit B Fails 189
Figure 102 - Retry Applied on Unit B Items 189
Figure 103 - Failed Polls Allowed Counter 190
Figure 104 - Unit B Confirmed Failure 190
Figure 105 - Reconnect Delay 190
Figure 106 - Reconnect Retries 190
Figure 107 - Packets Sent and Received at a Certain Point 191
Figure 108 - Server Configuration 192
Figure 109 - Monitoring Request and Reply Counters from the Client 192

Table of Tables

Table 1 - Terms and Definitions 16
Table 2 - Tool Tray Menu Commands 20
Table 3 - Control Panel Commands 21
Table 4 - Configuration Window Commands 23
Table 5 - File Menu Commands 23
Table 6 - Edit Menu Commands (Server Configuration) 24
Table 7 - Edit Menu Commands (Alias Configuration) 24
Table 8 - View Menu Commands 24
Table 9 - Tools Menu Commands 25
Table 10 - Help Menu Commands 25
Table 11 - General Options Tab Components 27
Table 12 - General Logging Options Tab Components 28
Table 13 - Interface Logging Options Tab Components 29
Table 14 - Advanced Options Tab Components 31
Table 15 - Insert New Object Window Options 32
Table 16 - Create New Fail-Over Channel Window - Communication Tab Components 36
Table 17 - Create New Network Channel Window Components 40
Table 18 - Redundancy Modes 42
Table 19 - Create New Redundancy Group Window Components 44
Table 20 - Create New Serial Channel Window - Communication Tab Components 48
Table 21 - Create New Serial Channel Window - Data Acquisition Tab Components 49
Table 22 - Create New Network Host Window Components 52
Table 23 - Create New Server Status List Window Components 55
Table 24 - Create New Dial-Up Connection Window - Dial-Up Tab Components 59
Table 25 - Create New Dial-Up Connection Window - Modem Tab Components 60
Table 26 - Create New Dial-Up Connection Window - Advanced Tab Components 62
Table 27 - Create New Modbus Unit Window - Communication Tab Components 66
Table 28 - Slave Mode Options Components 66
Table 29 - Create New Modbus Unit Window - Data Acquisition Tab Components 68
Table 30 - Create New Modbus Unit Window - Optimization Tab Components 70
Table 31 - Create New Radio Window Components 72
Table 32 - Insert New Alias Window – Alias Type Options 76
Table 33 - Insert New Alias Window – Alias Properties Options 76
Table 34 - Configure Scaling Window Options 79
Table 35 - Generate Events Window Options 82
Table 36 - New Category Window Options 82

Table 37 - Expression Wizard Options 84
Table 38 – Item Path Browser Options 88
Table 39 - Item ID Field Descriptions 91
Table 40 - Standard Data Items 92
Table 41 - Server Data Items 92
Table 42 - Redundancy Device Link OPC Items 93
Table 43 – Server Status List Device Link OPC Items 93
Table 44 - Sample Item ID Descriptions 94
Table 45 - Access Path Descriptions 95
Table 46 - Supported Features 104
Table 47 - Data Types 105
Table 48 - Available Item Types 105
Table 49 - MatrikonOPC Support Regional Contact Information 118
Table 50 - After-Hours Support 118
Table 51 – Performance Report 122
Table 52 - Expressions Constructs for Alias Scaling 130
Table 53 - Calculation Operators for Alias Scaling 131
Table 54 - Calculation Functions for Alias Scaling 132
Table 55 - File Import/Export Field Descriptions 133
Table 56 - Additional File Import/Export Field Descriptions - Linear/Square Root Scaling 133
Table 57 - Additional File Import/Export Field Descriptions - Offset Scaling 133
Table 58 - Additional File Import/Export Field Descriptions - Expression Scaling 133
Table 59 - Additional File Import/Export Field Descriptions - Calculation Scaling 133
Table 60 - Scaling Algorithms 134
Table 61 - Standard Data Types 135
Table 62 – Item Quality 139
Table 63 - Default Tag Security 161
Table 64 - Example 1 162
Table 65 - Example 2: Permissions 162
Table 66 - Example 2: Groups and Members 162
Table 67 - Example 2: Permissions 165
Table 68 - Offline Mode Item Values 167
Table 69 - Files Installed in "SCADA Modbus" Folder 179
Table 70 - Files Installed in "Common" Folder 180
Table 71 - Files Installed in "system32" Folder 181
Table 72 - Root Device Link OPC Items 194
Table 73 - Generic OPC Items 195
Table 74 - Network Channel OPC Items 196
Table 75 - Network Host OPC Items 196
Table 76 - Modbus Unit OPC Items 199
Table 77 - Fail-Over Channel OPC Items 199
Table 78 - Serial Channel Object OPC Items 201
Table 79 - Radio Object OPC Items 202
Table 80 - Dial-Up Object OPC Items 203

Introduction

The MatrikonOPC™ Server for SCADA Modbus provides OPC access to field devices compatible with the Modbus protocol, satisfying the needs of supervisory control and data acquisition applications, even under adverse site conditions (e.g., poor throughput, line noise).

The MatrikonOPC Server for SCADA Modbus includes support for the following communication methods:

- Modbus Master and Modbus Slave modes of operation
- Modbus RTU, Modbus ASCII, and Modbus TCP protocols
- Direct Serial connection (RS-232/422/485)
- Radio, dial-up, and CDPD modems
- TCP/IP and UDP/IP network connections
- TCP/IP or UDP/IP terminal servers
- Read and write all types of Modbus registers
- 32-bit register support (Enron/Daniels)
- Single-precision IEEE floating-point analog registers

Who Should Use This Manual

This manual is intended for use by all users of the MatrikonOPC Server for SCADA Modbus.

This manual explains how to install and configure the software, and how to perform common tasks. In addition, technical information about OPC data items is included, along with sections on diagnostics and troubleshooting.

Overview of Manual

This document uses icons to highlight valuable information. Remember these icons and what they mean, as they will assist you throughout the manual.

	This symbol denotes important information that must be acknowledged. Failure to do so may result in the software not functioning properly.
BOLD	Font displayed in this color and style indicates a hyperlink to the applicable/associated information within this document, or if applicable, any external sources.

The *User's Manual* has been designed as such so that you can click on references in the document to jump to that referenced point without having to scroll through several pages (in some cases). For example, if you were to see the sentence "*Refer to Figure 1 for more information*", pressing the **CTRL** key and clicking your mouse on the text "*Figure 1*" automatically takes you to the location of Figure 1 within the document.

This manual consists of several sections and is structured as follows:

- **Introduction** – this introductory chapter.
- **Getting Started** – provides system requirements information.

- **Configuration** – shows how to start and configure the server, and describes each component in detail, including windows/screens, panels, tabs, and menu commands.
- **OPC Data Items** – describes the server's items.
- **Implementation** – details the server's device-specific implementation.
- **Diagnostics** – explains how to use logging, statistical items, data qualities, timestamps, and result codes to get the most efficiency from the user's system.
- **Limitations** – provides information on specific performance and operational limitations of the software.
- **Troubleshooting** – provides licensing, MatrikonOPC Support contact information, solutions for common problems that may be encountered, and answers to frequently asked questions.
- **OPC Compliance** – details supported interfaces with regard to installation, common interfaces, and data access.
- **Appendices**:
    - **A** - DCOM
    - **B** – Aliases
    - **C** – Standard Data Types
    - **D** – Microsoft Excel as a DDE Client
    - **E** – Configuring Dynamic Data Exchange (DDE)
    - **F** – Item Quality
    - **G**– Security
    - **H** – Offline Mode
    - **I** – Installation
    - **J** – Installed Files
    - **K** – Un-Installation
    - **L** – SCADA Servers Connection Logic
    - **M** – Features and Examples
    - **N** – OPC Items

References

This document references information found within the following documents/sites:

- **www.opcfoundation.org**
- **www.matrikonopc.com**
- **www.opcsupport.com**
- **www.modbus.org**
- *MODBUS Application Protocol Specification (ver. 1)*. The Modbus Organization, May 2002
- *OPC Overview 1.0*
- *OPC Common Definitions and Interfaces 1.0*
- *OPC Data Access Specification 2.05a*

- *OPC Data Access Specification 3.00*
- *OPC Alarms and Events Specification 1.10*

Document Terminology

The following terms are used interchangeably throughout this document:

- *screen* and *window*
- *tab* and *panel*

Table 1 provides a list of definitions for terms used throughout this document.

Term/Abbreviation	Description
A&E	OPC Alarms and Events. Provides access to process alarm and event data.
ADT	Asynchronous Data Transfer.
API	Application Programming Interface.
COM	Component Object Model. A method for organizing software, specifying how to build components that can be dynamically interchanged.
DA	OPC Data Access. Provides access to real-time process data.
DCOM	Distributed Component Object Model. An extension of COM that allows communication between COM components over a network.
DDE	Dynamic Data Exchange. Allows the transfer of data between two running applications.
HDA	OPC Historical Data Access. Provides access to historical process data.
HMI	Human Machine Interface. Device that allows interaction between the user and machine. Typically used in process control applications.
Matrikon	Matrikon International.
MatrikonOPC	Matrikon's brand name for its OPC servers and clients.
OPC	A communication standard. Refer to **www.opcfoundation.org** for more information.
PLC	Programmable Logic Controller.
SCADA	Supervisory Control And Data Acquisition. Software that is interfaced to hardware via a PLC or other API.

Table 1 - Terms and Definitions

Getting Started

This chapter contains important information about installing the server and how to contact the MatrikonOPC Support team.

The **System Requirements** section shows how to avoid future problems by ensuring that the system meets the minimum software and hardware requirements. Detailed step-by-step instructions in **Appendix I - Installation** walks you through the installation process. **Appendix J – Installed Files** lists the files that are installed during this process.

Once the software is installed, refer to the **Licensing** section for information on how to obtain the appropriate license. The *Licensing* section will refer you to the *Licensing Procedures* document that was installed along with the server and this *User's Manual*. If any problems are encountered during installation or licensing, refer to the **Contacting Support** section for information about how to contact the MatrikonOPC Support team for assistance.

System Requirements

The software has minimum **Software** and **Hardware** system requirements. These requirements must be met for the software to function properly.

Note: To install and configure a MatrikonOPC server, you must set up as an administrative user account rather than a restricted user account.

Software Requirements

The server requires the following software:

- Microsoft Windows Server 2012 (64-bit)
- Microsoft Windows Server 2008 R2 SP1 (64-bit)
- Microsoft Windows Server 2008 SP1 (32-bit and 64-bit)
- Microsoft Windows 7 SP0 (32-bit and 64-bit)
- Microsoft Windows 2003 R2 SP0 (32-bit and 64-bit)
- Microsoft Windows 2003 R1 SP0 (32-bit and 64-bit)
- Microsoft .NET 2.0 Framework (included with this install program). This framework is valid for 32-bit operating systems only. It has to be installed separately on a 64-bit operating system.

Your system must have the following additional software to use the corresponding functionality:

- TCP/IP or UDP/IP terminal server

Note: It is recommended that the most current service packs are installed.

Hardware Requirements

The server requires the following hardware:

- Intel CORE2 Duo Processor
- 2 GB RAM

- 80 GB 7200 RPM Hard Drive

The server requires the following additional hardware to make use of the following functionality:

- Serial communications port
- RS-232/422/485 cabling and/or converter
- Radio modem
- Hayes-compatible dial-up modem
- CDPD modem with support for the AT command set
- IP-compatible network

Configuration

Starting the MatrikonOPC Server for SCADA Modbus

To launch the MatrikonOPC Server for SCADA Modbus, choose the appropriate shortcut from the **Start** menu.

To start the MatrikonOPC Server for SCADA Modbus:

Click on the Windows **Start** button and select **Programs** -> **MatrikonOPC** -> **SCADA Modbus**, and choose **MatrikonOPC Server for SCADA Modbus**.

By default, the server is installed as a Windows service. When the server is installed as a service, the splash screen does not appear, nor does the logo appear in the **Tool Tray**. Instead, the main **Configuration** window is immediately displayed. If the server is installed as a Windows service, users can also start it using the **Service Control Manager** applet (under the Windows **Control Panel**).

To run MatrikonOPC Server for SCADA Modbus as an application:

1. In the Windows **Service** window, stop the Service of MatrikonOPC Server for SCADA Modbus.
2. To un-register the service, type the command below:
   C:\Program Files\Matrikon\OPC\SCADA Modbus\OPCScadaModbus.exe –unregserver
3. After un-registering, type the command below:
   C:\Program Files\Matrikon\OPC\SCADA Modbus\OPCScadaModbus.exe –regserver

If the server is not installed as a Windows service, the server starts and the MatrikonOPC logo appears in the **Tool Tray** (Figure 1)located by default in the system tray.

Figure 1 - Tool Tray

Tool Tray Menu

Note: If the server is installed as a Windows service, the **Tool Tray** and **Tool Tray Menu** are not available.

The **Tool Tray Menu** provides access to the **Control Panel**, the **Configuration** window, and the **About** screen. It also contains the commands to shut down the server.

To view the Tool Tray Menu:

1. Right-click your mouse on the **MatrikonOPC logo**.
2. The **Tool Tray Menu** appears (Figure 2).

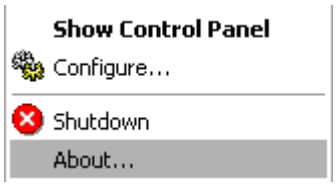


Figure 2 - Tool Tray Menu

Table 2 describes the commands in the **Tool Tray Menu**.

Command	Description
Show Control Panel	Displays the configuration **control panel**.
Configure	Displays the main **Configuration window**.
Shutdown	Shuts down the server.
About	Displays the server's **Welcome** screen, which includes information about the software.

Table 2 - Tool Tray Menu Commands

Control Panel

Note: If the server is installed as a Windows service, the **Control Panel** is not available.

The **Control Panel** displays the server's status, system time, and number of clients connected to the server. The **Control Panel** allows users to:

- Access the **Configuration** window.
- Access the **About** screen.
- Execute the command to shut down the server.

The **Control Panel** stays on top of every other window on the desktop to help monitor the status of the server.

To view the Control Panel:

1. Right-click your mouse on the **MatrikonOPC logo** .
2. The **Tool Tray Menu** appears.
3. From the **Tool Tray Menu**, select the **Show Control Panel** menu option.
4. The **Control Panel** (Figure 3) appears.

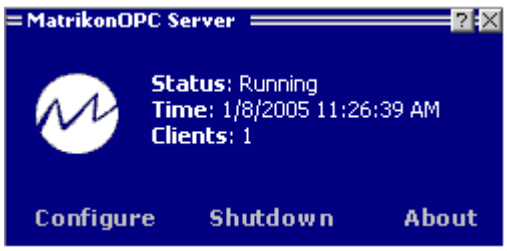


Figure 3 - Control Panel

5. To "window shade" the **Control Panel** so that it displays only the client count, double-click your mouse on the title bar.

Table 3 describes the commands in the **Tool Tray Menu**.

Command	Description
Status	Displays the current status of the server.
Time	Displays the current time.
Clients	Displays the total number of OPC clients presently connected to the server.
Configure	Displays the main **Configuration** window.
Shutdown	Shuts down the server.
About	Displays the server's **Welcome** screen, which includes information about the software.

Table 3 - Control Panel Commands

Configuration Window

Note: If the server is installed as a Windows service, the **Tool Tray Menu** and **Control Panel** are not available. In this case, the **Configuration** window is then accessed via the Windows **Start** menu.

The **Configuration** window displays a tree view of objects configured in the server. Select an object to display its current configuration, which appears on the right-hand side of the window.

If the server runs as a local executable, it will remain running even after the main configuration window is closed, even if no clients are connected. Displaying the control panel or main configuration window for the server will "lock" it so that it will not shut down while configuration is being performed.

If the server runs as an NT service, it will remain running so long as there are clients connected to it. Closing the main configuration window will shut down the configuration utility, but will not interfere with the normal operation of the server.

To view the Configuration window:

1. From the **Tool Tray Menu** or the **Control Panel**, select **Configure**.
2. The **Configuration** window (Figure 4) appears.

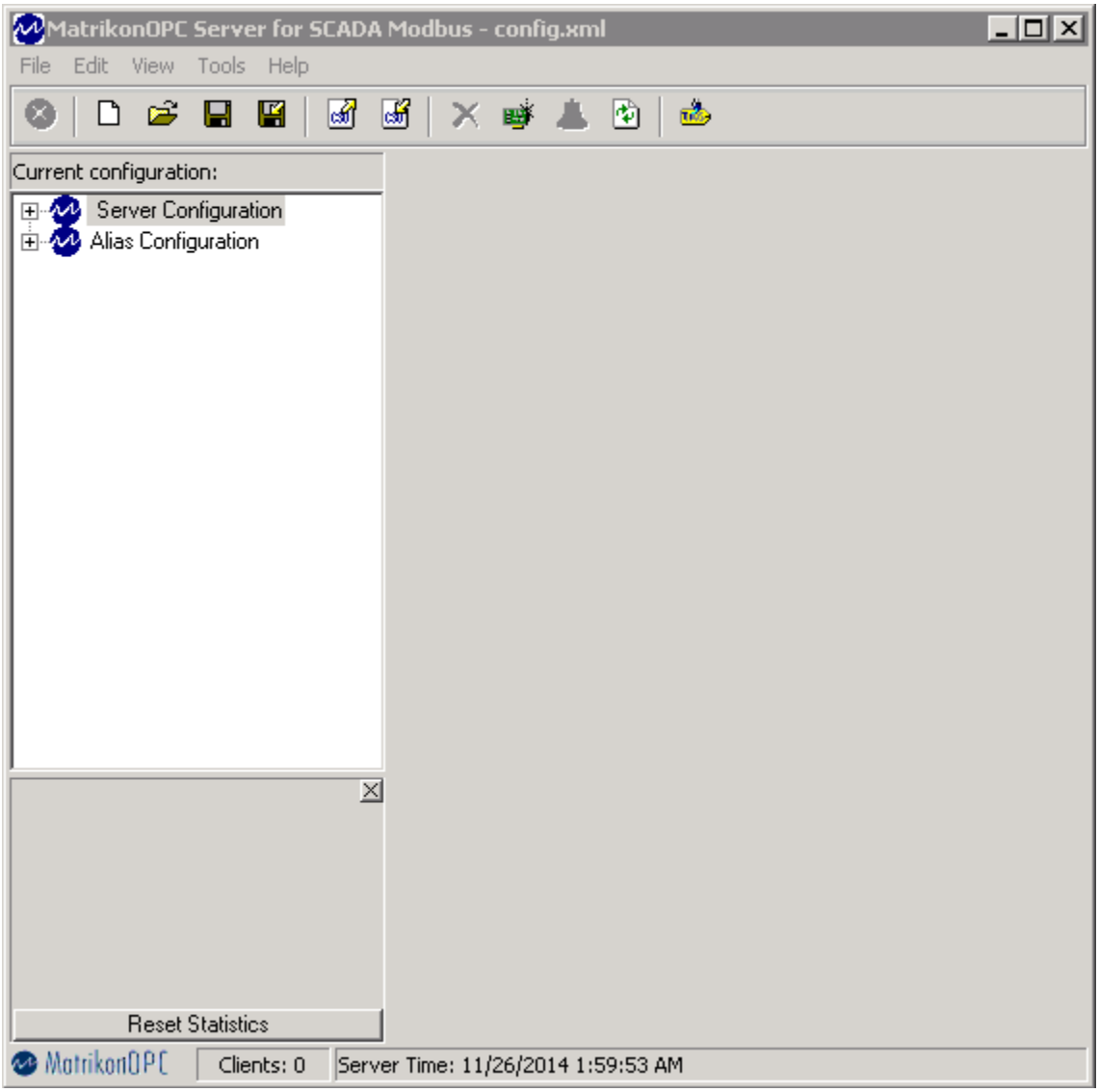


Figure 4 - Configuration Window

Table 4 describes the commands in the **Configuration** window.

Command	Description
Main Menu	Provides access to the **File**, **Edit**, **View**, **Tools**, and **Help** menus.
Main Toolbar	Provides shortcut buttons for commands in the **Main Menu**.
Current Configuration	Displays a tree of configuration objects currently defined in the server. Select an object to display its current settings in the right-hand panel.
Server Configuration	Objects under this node affect how the server communicates with the data source.
Alias Configuration	Define aliases and alias groups under this node.
Object Settings Panel	Displays the current settings of the object presently selected in the tree.
Statistics Panel	Displays available statistics for the object presently selected in the tree: ***Sent***, ***Received***, ***Timed Out***, ***Retried***, ***Failed***, ***Overrun***.

Command	Description
Reset Statistics	Resets statistics on the object presently selected in the configuration tree.
MatrikonOPC Logo	Opens a link to the MatrikonOPC web site.
Clients	Displays the total number of OPC clients presently connected to the server.
Server Time	Displays the current time.

Table 4 - Configuration Window Commands

The following sections describe the menus available from the **Configuration** window, and what they are used for.

File Menu

Table 5 describes the **File** menu commands.

Command	Description
New	Clears the current configuration and starts a new one. If OPC clients are still connected to the server, the user is prompted for confirmation before continuing.
Open	Clears the current configuration and loads a new one from an existing file. Displays the **Open Configuration** window to prompt for the file name. If OPC clients are still connected to the server, the user is prompted for confirmation before continuing.
Save	Saves the current configuration to an XML file. Displays the **Save Configuration** window to prompt for a new file name if the configuration is new and has not been saved before.
Save As	Saves the current configuration to an XML file. Displays the **Save Configuration** window to prompt for a new file name.
Export Aliases	Exports all currently configured aliases to a CSV file. Displays the **Export Aliases** window to prompt for a file name.
Import Aliases	Clears all aliases and imports new ones from a CSV file. Displays the **Import Aliases** window to prompt for an existing file name.
Shutdown Server	Closes the current configuration and shuts down the server after signalling the shutdown event to all connected clients. If OPC clients are still connected to the server, the user is prompted for confirmation before continuing.
Close	Hides the main **Configuration** window. The server shuts down after a short delay if no OPC clients are currently connected to the server.

Table 5 - File Menu Commands

Edit Menu

Table 6 describes the **Edit** menu commands that appear when the currently selected object appears under the **Server Configuration** node.

Command	Description
Can't Undo	Not implemented.
Delete	Removes the currently selected object from the server configuration.

Command	Description
	Prompts the user for confirmation before continuing to see if the OPC clients have reference to items associated with the selected object or one of its children. Does not remove such objects but simply disables them and sets them to "marked for deletion".
Enable Group Events	Allow a group or item to trigger events.
Define New	Displays the **Insert New Object** window used to create a new configuration object under the currently selected configuration object.

Table 6 - Edit Menu Commands (Server Configuration)

Table 7 describes the **Edit** menu commands that appear when the currently selected object appears under the **Alias Configuration** node.

Command	Description
Undo/Can't Undo	Not implemented.
Rename	Renames the currently selected alias group.
Delete Alias Group	Removes the currently selected alias group and all subordinate groups and aliases.
Delete Alias	Removes the currently selected aliases.
Insert Alias Group	Creates a new alias group under the currently selected alias group.
Insert New Alias	Displays the **Insert New Alias** window used to create a new alias under the currently selected alias group.

Table 7 - Edit Menu Commands (Alias Configuration)

View Menu

Table 8 describes the **View** menu commands.

Command	Description
Statistics	Displays/hides general communication statistics for the currently selected object.
Reset Statistics	Resets all statistics for the currently selected object to zero.
Defaults	Displays the **Server Defaults** window used to configure the default settings for the various types of communication objects available in the server.
Refresh	Rebuilds the tree display of the server configuration and fully expands all branches.
Options	Displays the **Options** window used to change general server options.

Table 8 - View Menu Commands

Tools Menu

Table 9 describes the **Tools** menu commands.

Command	Description
DCOM Configurator	Launches the **Microsoft Distributed COM Configuration Properties** utility. This utility can be used to edit DCOM settings for specific COM

Command	Description
	servers to allow client applications on remote computers to connect to the server.
View Tags	Launches MatrikonOPC Explorer, which is a general-purpose OPC client included with all MatrikonOPC servers that may be used to test the capabilities of the software. For more information, refer to the *MatrikonOPC Explorer User's Manual*.
Security Settings	Launches the **Enter server password** window where you are asked to enter the security password set during the installation of the server. The default password set during the installation of this server is ***MatrikonOPC***. The **MatrikonOPC Tag Security Configuration Utility** screen is then displayed which allows you to configure security settings. For more information, refer to **Appendix G – Security**.

Table 9 - Tools Menu Commands

Help Menu

Table 10 describes the **Help** menu commands.

Command	Description
MatrikonOPC on the Web	Opens a link to the MatrikonOPC web site.
About	Displays the **Welcome** screen, which includes information about the software version.

Table 10 - Help Menu Commands

Options Window

The **Options** window is used to edit the server's options and contains four tabs: **General**, **General Logging**, **Interface Logging**, and **Advanced**.

General Options

The **General** tab allows users to edit general server options (i.e., settings).

To view the General options:

1. On the **Configuration** window, from the **View** menu, select the **Options** menu item.
2. The **Options** window appears.
3. From the **Options** window, select the **General** tab (Figure 5).

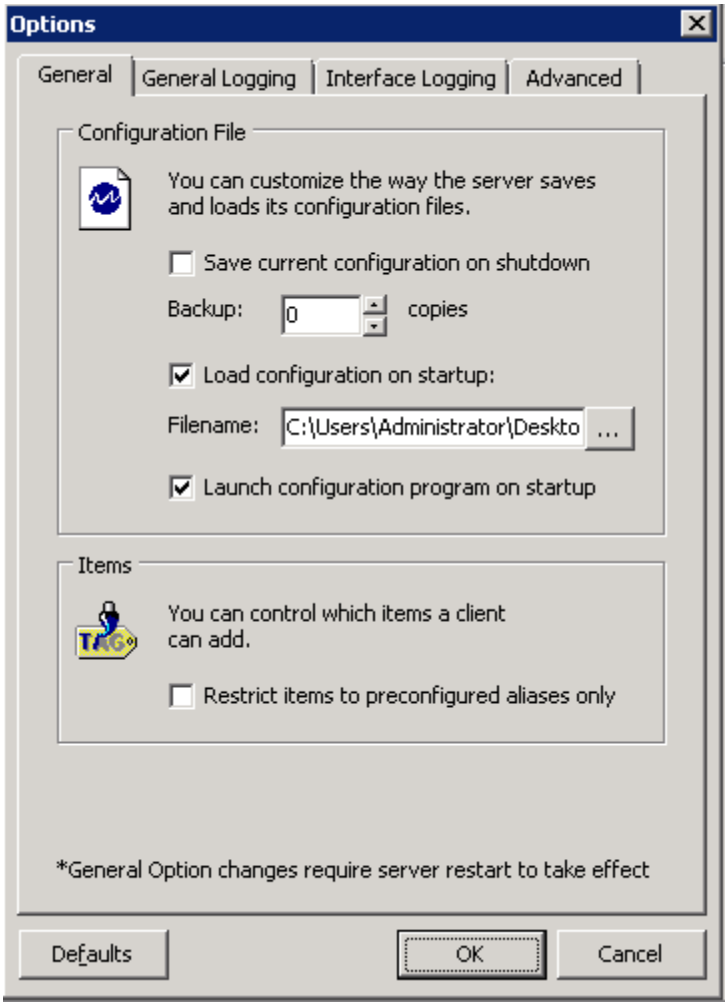


Figure 5 - General Options Tab

Table 11 describes the fields available on the **General** tab.

Field	Description
Save current configuration on shutdown	Automatically saves the current configuration to the specified file before shutting down, if selected.
Backup: X copies	Automatically keeps up to the specified number (i.e., X) of backup copies of the configuration file when saving the configuration.
Load configuration on setup	Automatically loads a configuration from the specified file when starting up, if selected.
Filename	Displays the full path of an existing file for load and save operations. Click on the ellipsis button (...) to display the **Open** window used to select an existing file name.
Launch configuration program on startup	Automatically launches the configuration utility when starting up as a local executable (rather than a Windows service), if selected.
Restrict items to preconfigured aliases	Prevents OPC clients from adding items other than configured aliases, if selected. Selecting this option will cause clients to see only aliases when

Field	Description
only	browsing this server. **Note:** As is the case with any changes to General options, if this option is changed you will need to restart the server for the change to take effect.

Table 11 - General Options Tab Components

Note: The server loads these settings from the Windows registry upon start-up. Changes to the **General** options settings take effect when the server is restarted.

General Logging Options

The **General Logging** tab allows users to edit general logging options.

To view the General Logging options:

1. On the **Configuration** window, from the **View** menu, select the **Options** menu item.
2. The **Options** window appears.
3. From the **Options** window, select the **General Logging** tab (Figure 6).

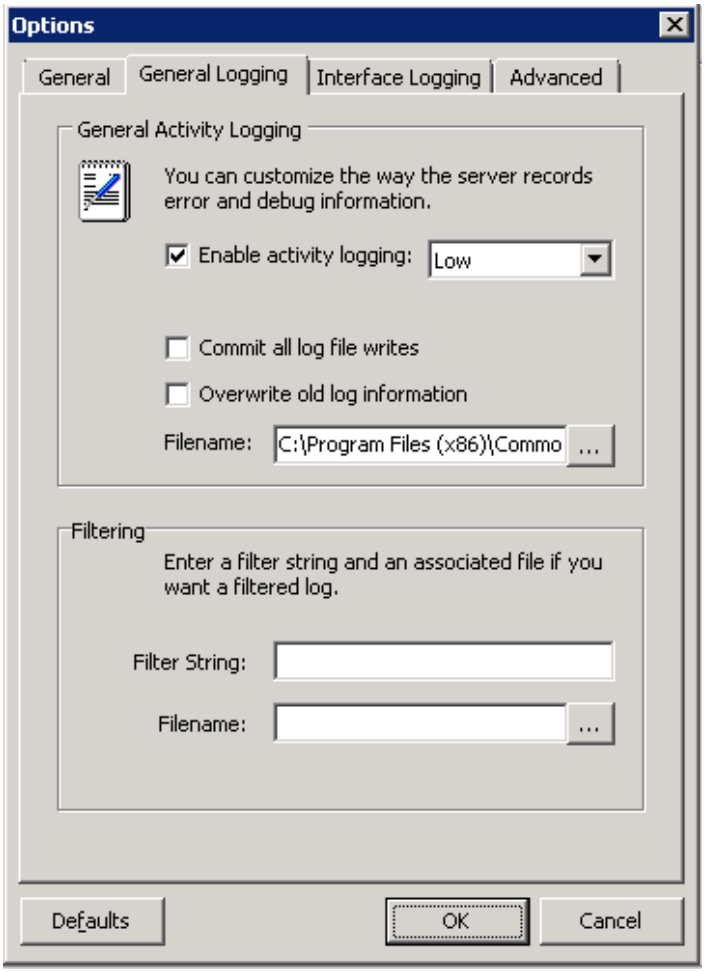


Figure 6 - General Logging Options Tab

Table 12 describes the fields available on the General Logging tab.

Field	Description
Enable activity logging	Enables/Disables general activity logging at the specified detail level (***High***, ***Medium***, ***Low***, or ***None***), if selected. **Note:** This option should **not** be selected unless users are debugging a problem as this will have a negative effect on performance of the OPC server. Always set this back to ***None*** when not in use.
Commit all log file writes	Purges the file buffer after each message is logged in the event of an unexpected server shut down.
Overwrite old log information	Overwrites the old log file each time the server starts up, if selected. Otherwise, rename the old log file with a (*.*bak*) suffix.
General Activity Logging Filename	Displays the full path for the general activity log file. Click on the ellipsis button () to display the **Open** window used to select an existing file name. For 64-bit, the default path is ***C:\Program Files(x86)\Common Files\MatrikonOPC\Common\PSTCFGMatrikon.OPC.SCADAModbus.1.LOG***. For 32-bit, the default path is ***C:\Program Files\Common Files\MatrikonOPC\Common\PSTCFGMatrikon.OPC.SCADAModbus.1.LOG***.
Filter String	Match string for filtered log is displayed.
Filtering Filename	Displays the full path for the filtered log file. Click on the ellipsis button () to display the **Open** window used to select an existing file name.

Table 12 - General Logging Options Tab Components

Note: General Logging options settings take effect immediately.

Interface Logging Options

The **Interface Logging** tab allows users to view and edit COM interface logging options.

To view the Interface Logging options:

1. On the **Configuration** window, from the **View** menu, select the **Options** menu item.
2. The **Options** window appears.
3. From the **Options** window, select the **Interface Logging** tab (Figure 7).

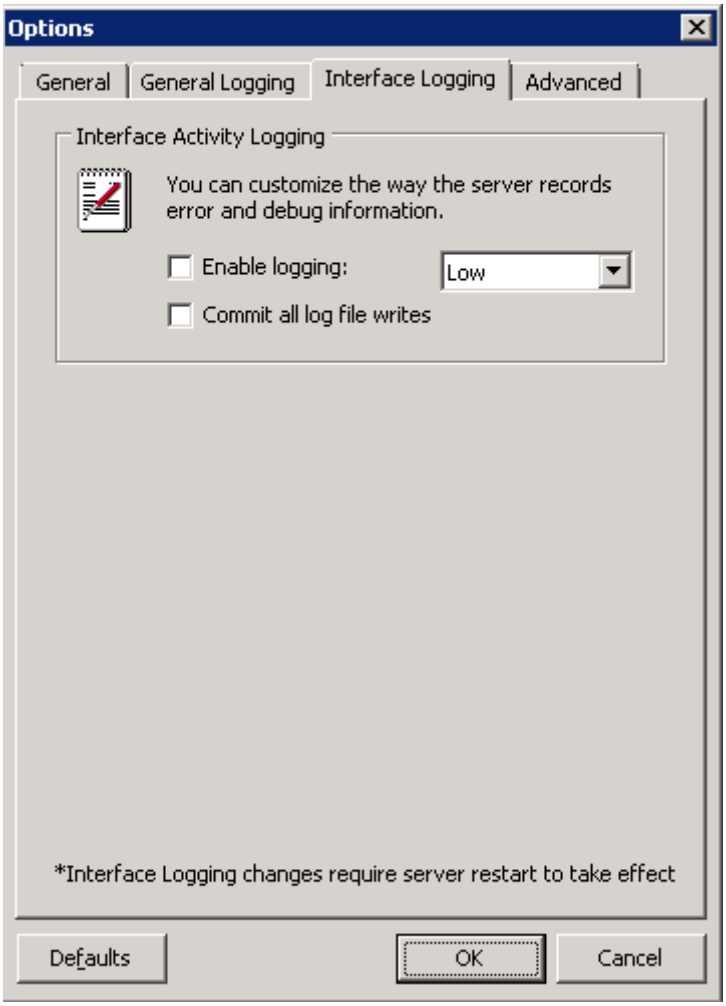


Figure 7 - Interface Logging Options Tab

Table 13 describes the fields available on the **Interface Logging** tab.

Field	Description
Enable logging	Enables/disables interface logging at the specified detail level (***High***, ***Medium***, ***Low***, or ***None***), if selected. Generates a separate log file for each client named *PST###.TMP* in the server installation directory. **Note:** This option should **not** be selected unless users are debugging a problem as this will have a negative effect on performance of the OPC server.
Commit all log file writes	Purges the file buffer after each message is logged in the event of an unexpected server shut down.

Table 13 - Interface Logging Options Tab Components

Note: Restart the server for **Interface Logging** options settings to take effect.

Advanced Options

The **Advanced** tab allows users to view and edit advanced server options.

To view the Advanced options:

1. On the **Configuration** window, from the **View** menu, select the **Options** menu item.
2. The **Options** window appears.
3. From the **Options** window, select the **Advanced** tab (Figure 8).

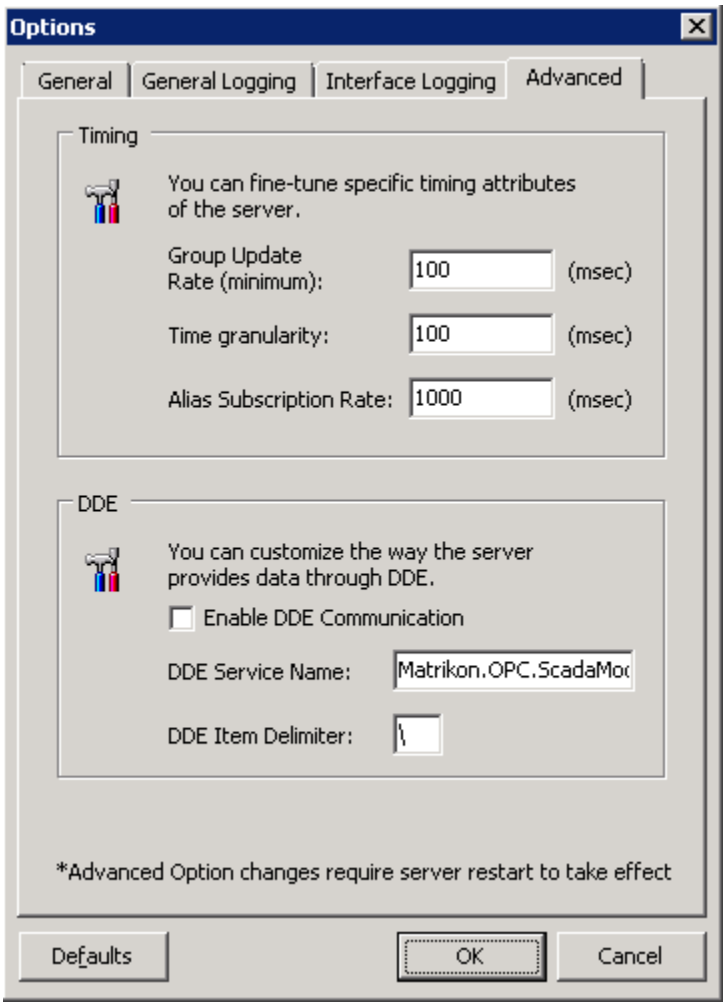


Figure 8 - Advanced Options Tab

Table 14 describes the fields available on the **Advanced** options tab.

Field	Description
OPC Group Update Rate (minimum)	Minimum update rate (milliseconds) allowed for an OPC group. Prevents OPC clients from requesting update rates that are impossible to achieve or that cause heavy CPU loads. The default value is 100 ms.
Time granularity	OPC server group scan thread interval (milliseconds). Determines the maximum resolution for data change detection. Time granularity should always be equal to or less than the minimum group update rate.
Alias Subscription Rate	This is the rate (in milliseconds) at which the Alias features subscribe to underlying data items. This rate is used by the Alias Events

Field	Description
	features and by the Expression and Calculation engines when other tags are referenced in the expressions. Default = ***1000***.
Enable DDE Communication	Selecting this checkbox this will allow the server to accept Dynamic Data Exchange (DDE) connections.
DDE Service Name	The DDE service name that should be used by clients to connect to this program.
DDE Item Delimiter	The character to be used to indicate the start of a new item in the DDE data request.

Table 14 - Advanced Options Tab Components

Note: The server loads these settings from the Windows registry on start-up. Changes to **Advanced** options settings take effect when the server is shut down and restarted.

Server Configuration

The server requires minimal configuration for it to run properly. You can customize the server's behaviour as required. This section shows you how to create and edit objects, using the **Configuration** window.

Creating Objects

The **Insert New Object** and **Create New** windows are used to create MatrikonOPC Server for SCADA Modbus device link objects.

Insert New Object Window

The **Insert New Object** window (Figure 9) displays a different set of options depending on the type of object selected and the types of objects already configured under it.

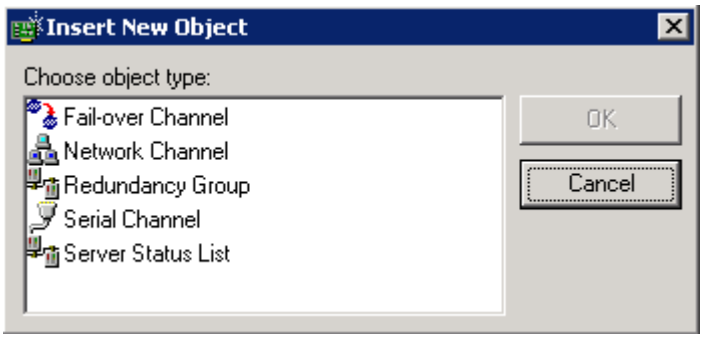


Figure 9 - Insert New Object Window

Table 15 describes the types of objects that are available for insertion as a child in the **Insert New Object** window, depending on which object is currently selected.

Selected Object	Insert New Object Window Options
Root Device Link (Server Configuration)	Fail-over Channel Network Channel Redundancy Group

Selected Object	Insert New Object Window Options
	Serial Channel Server Status List
Fail-over Channel	<None>
Network Channel	Network Host
Network Host	Dial-up Connection Modbus Unit
Dial-up Connection	Modbus Unit
Modbus Unit	<None>
Redundancy Group	<None>
Serial Channel	Dial-up Connection Modbus Unit Radio
Dial-up Connection	Modbus Unit
Modbus Unit	<None>
Radio	Modbus Unit
Server Status List	<None>

Table 15 - Insert New Object Window Options

To insert a new object:

1. On the **Configuration** window, select the **Server Configuration** node and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the  icon.
2. The **Insert New Object** window (Figure 9) appears.
3. From the displayed list, select the type of object to be created.

   Note: Different objects may be available depending on which object is currently selected in the tree view.
4. Click on the **OK** button.
5. The **Create New** window appears.

To create a newly inserted object:

1. From the **Create New** window, enter a name for the object. This name is part of the item ID that globally identifies data items from under the object. You may also choose to enter a description of the object.
2. Edit the configuration components as desired.

3. Click on the **OK** button. The object is created.

   Note: For the object to be used for communication, it must be enabled. This can be done at any time.

Create New Window

The **Create New** window (Figure 10) displays the settings available for a new object.

A device-specific window is displayed for each object:

- **Fail-over Channel**
- **Network Channel**
- **Redundancy Group**
- **Serial Channel**
- **Server Status List**

The following sub-sections contain information on the types of objects available and how to configure them.

Creating and Configuring a Fail-Over Channel Object

The **Create New** window (Figure 10) displays the settings available when creating a **Fail-over Channel** object.

Once a Fail-over Channel object has been created, and you select it in **Current configuration** pane, the settings panel is displayed and consists of two tabs:

- **Communication**
- **Load Distribution**

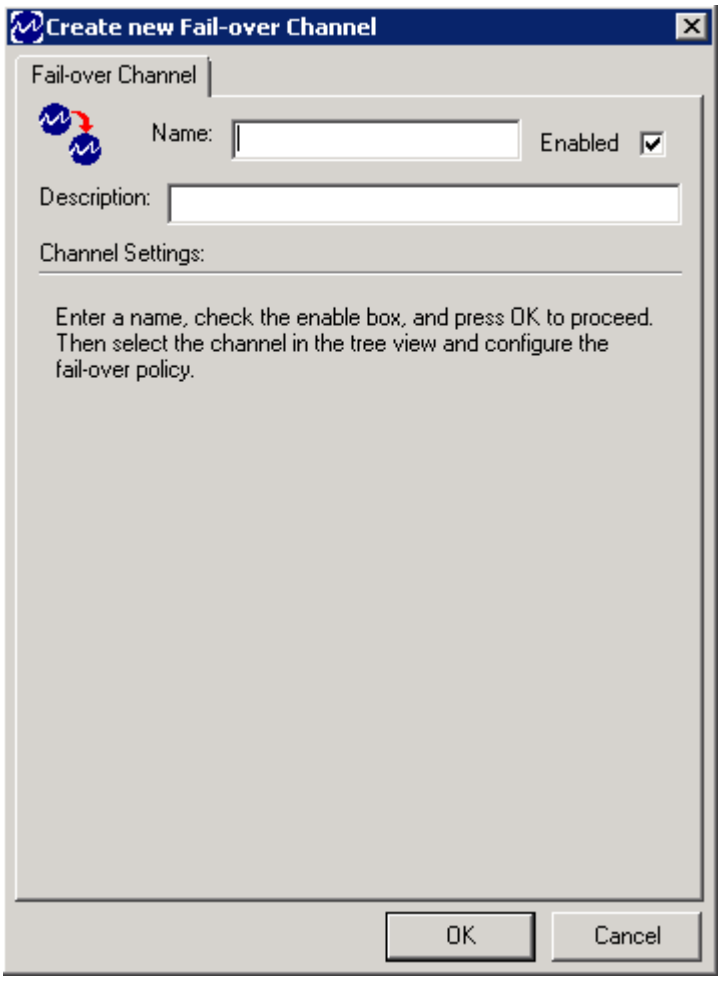


Figure 10 - Create New Fail-Over Channel Window

The **Communication** tab (Figure 11) is not available until you have already created the Fail-over Channel. To display this tab, select the Fail-over Channel in the **Current Configuration** list and it will appear in the **Fail-over Channel Settings** window.

Communication Tab

Figure 11 displays the **Fail-over Channel – Communication** tab.

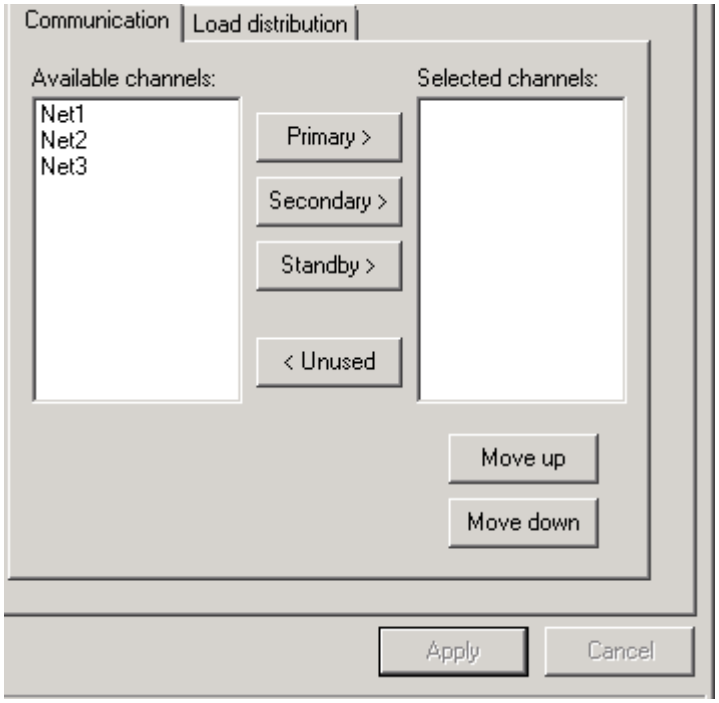


Figure 11 - Fail-Over Channel - Communication Tab

Table 16 describes the components of the **Create New Fail-Over Channel** - **Communication** tab.

Component	Description
Name	Name of the object, which becomes an element of the item ID for data items under the object. Automatically converted to title case for display purposes. **Note:** This information stays consistent across both tabs.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected. **Note:** This information stays consistent across both tabs.
Description	This field is optional, takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank. **Note:** This information stays consistent across both tabs.
Available channels	Lists those communication channels not currently used in the fail-over policy.
Selected channels	Describes the fail-over policy and the load distribution criteria for standby channels as an XML string. Default = ***<CFailoverDevlink/>)****.
Primary	Designates the channel as a Primary fail-over channel and transfers it to the **Selected channels** list.
Secondary	Designates the channel as a Secondary fail-over channel and transfers it to the **Selected channels** list.

Component	Description
Standby	Designates the channel as a Standby fail-over channel and transfers it to the **Selected channels** list.
Unused	Removes the channel from the fail-over policy and transfers it back to the Available channels list.
Move up	Increases the fail-over channels priority by one and moves it up one space in the **Selected channels** list.
Move down	Decreases the fail-over channels priority by one and moves it down one space in the **Selected channels** list.
OK	Select this button to save any changes and close the window. **Note:** This information stays consistent across both tabs.
Cancel	Select this button to close the window without saving any changes made. **Note:** This information stays consistent across both tabs.

Table 16 - Create New Fail-Over Channel Window - Communication Tab Components

Load Distribution Tab

Figure 12 displays the **Fail-over Channel – Load Distribution** tab.

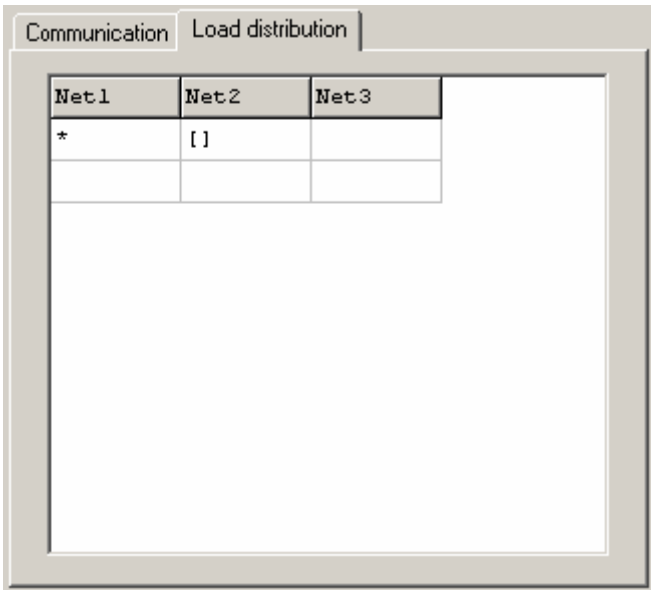


Figure 12 - Fail-Over Channel - Load Distribution Tab

Like the **Communication** tab, the **Load Distribution** tab (Figure 12) is only available after you have created the fail-over channel, in the **Create New Fail-over Channel** window.

The **Fail-over Channel** works on a tag by tag basis. If one tag is not available in Primary channel it will check the next connection in the list to attempt to get data for the tag. This also means that data can be collected from multiple different channels at the same time for **different** tags.

Notes:

- '*' indicates all items. [] indicates no items. A load distribution can be done by typing a part of the tag string under the desired Network channel in between 2 "*" symbols. For example if Net 1 is desired to browse all tags that belong to register 4X and Net 2 is desired to

browse all tags that belong to register 3X then the user will have to type *4000*(or 4:) under Net1 and *3000* (or 3:) under Net2.

- Trying to use a redundancy group as part of a Failover Channel will not yield any tags. It is recommended that the User keeps redundancy groups and Failover channels separate.

To create and configure a Fail-over Channel object:

1. On the **Configuration** window, select the **SCADA Modbus** configuration item and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the  icon.
2. The **Insert New Object** window (Figure 9) appears.
3. From the displayed list, select the **Fail-over Channel** object type.
4. Click on the **OK** button.
5. The **Create New** window appears (Figure 10).

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Fail-over Channel** is selected, then the **Create New Fail-over Channel** window is displayed.
6. From the **Create New** window, enter a name for the object.

   Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.
7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.
8. Click on the **OK** button. The object is created and appears as a child of the **SCADA Modbus** configuration item as shown in Figure 13.

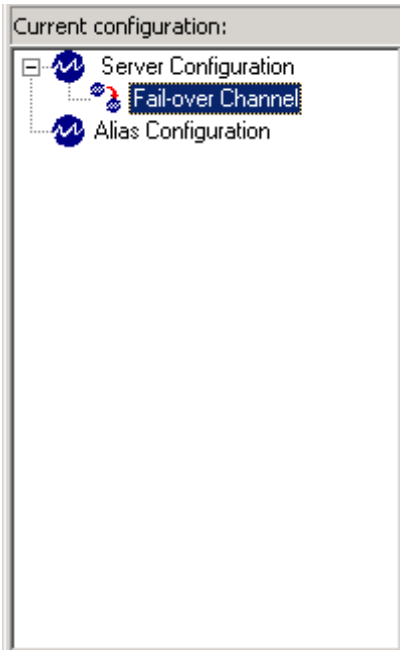


Figure 13 - New Fail-over Channel Object

Creating and Configuring a Network Channel Object

The **Create New Network Channel** window (Figure 14) displays the settings available for configuring a **Network Channel**.

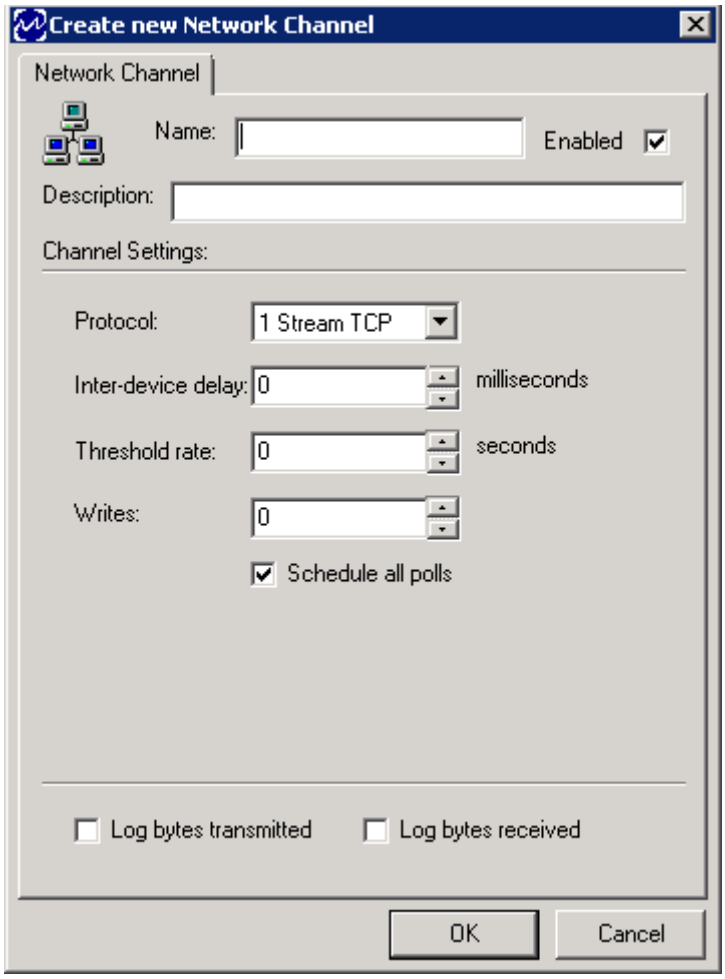


Figure 14 - Create New Network Channel Window

Table 17 describes the components of the **Create New Network Channel** window.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. The full path of the device link must be entered. Automatically converted to title case for display purposes.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank.
Protocol	Allows you to select an Internet Protocol (IP) type from the drop down list. Available options are: ***0 Datagram TCP***, ***1 Stream TCP***, ***2 Datagram UDP***. ***0 Datagram TCP*** and ***1 Stream TCP*** both use TCP/IP but ***0*** treats the protocol as a series of discrete packets, whereas ***1*** treats it as a continuous stream of bytes.

Component	Description
	Default = ***1 Stream TCP***. **Note:** The channel and all units must disconnect and reconnect if this parameter changes.
Inter-device delay	Allows you to enter a value specifying the amount of time (in milliseconds) that the communication channel will wait sending a new request to the next unit after the data has been received from the current unit on this channel. If this is set to 0 then the option is turned off. Default = ***0***.
Threshold rate	Allows you to enter a value specifying the maximum achievable scan interval (in seconds) for the circuit. The server polls items with faster update rates on a continuous basis rather than as scheduled. If this is set to 0 then the option is turned off. Default = ***0***. **Note:** All units on the channel must re-optimize if this parameter changes.
Writes	Allows you to enter or select a value specifying the maximum number of writes allowed before allowing a poll. Default = ***0***.
Schedule all polls	Selecting this checkbox indicates that the server must schedule items with update rates faster than the **Threshold rate** to poll at the **Threshold rate**. By default, this checkbox is selected. **Note:** All units on the channel must re-optimize if this parameter changes.
Log bytes transmitted	Enables (i.e., check box is selected) or disables (i.e., checkbox is cleared) the logging of bytes transmitted at Medium level. By default, this checkbox is cleared.
Log bytes received	Enables (i.e., check box is selected) or disables (i.e., checkbox is cleared) the logging of bytes received at Medium level. By default, this checkbox is cleared.
OK	Select this button to save any changes and close the window.
Cancel	Closes the window without saving any changes made.

Table 17 - Create New Network Channel Window Components

To create a Network Channel object:

1. On the **Configuration** window, select the **SCADA Modbus** configuration item and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the  icon.
2. The **Insert New Object** window (Figure 9) appears.
3. From the displayed list, select the **Network Channel** object type.
4. Click on the **OK** button.
5. The **Create New** window appears (Figure 14).

Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Network Channel** is selected, then the **Create New Network Channel** window is displayed.

6. From the **Create New** window, enter a name for the object.

   Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.

7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.

8. Click on the **OK** button. The object is created and appears as a child of the **Server Configuration** item as shown in Figure 15.

Figure 15 - New Network Channel Object

Creating and Configuring a Redundancy Group Object

Note: If a **Modbus Unit** is included in a Redundancy Group then it must be configured to **Stay Connected.**

The Redundancy Group contains a list of device links that can exist anywhere in the OPC server hierarchy. The Redundancy Group assigns one of the device links in the list as the active device link. Depending on the selected redundancy mode (see Table 18), the non-active device links are placed in communicating, standby or disabled states. As operations (e.g., adding items, reading/writing) are performed on the Redundancy Group, these actions are forwarded through to the active device link. If the active device link enters a failure condition (e.g., becomes disconnected), the Redundancy Group assigns a new active device link by iterating through its list until it can find a device link that is connected.

Multiple Redundancy Groups can exist, and there can be one or more device links contained within a Redundancy Group. A particular device link may only be a member of one redundancy group at a

time. The group tries to make one of the device links within itself, the active device link (i.e., the device link that the Redundancy Group is reading/writing data from/to).

Each device link within a Redundancy Group will have an assigned priority. When deciding which device link to make active, the Redundancy Group goes through the list of its device links in order of priority, from highest to lowest. When not in **Cold** redundancy mode, if a device link that has a higher priority than the active device links becomes connected, the Redundancy Group makes that particular device link active.

Items within the Redundancy Group are added with the same item ID as those in its redundant device links with the name of the redundant device link being replaced with the name of the Redundancy Group. Therefore, redundant device links should have identical hierarchy layout and naming conventions below their level. The success of a redundancy item addition depends on whether the Redundancy Group can add the mapped item to the active device link. OPC reads and writes performed on the redundancy item are forwarded to the mapped item in the active device link.

The **Create New Redundancy Group** window (Figure 16) displays the settings available for configuring a Redundancy Group object.

Mode	Description
Classic	All device links are connected. Items are added only to the active device link, but are not removed when a fail over occurs. As failovers occur and cause multiple device links to become active, the collection state becomes the same as **Hot.**
Hot	All device links are connected. When items are added they are added to all device links in the redundancy group. This means that all device links are actively collecting all the time.
Warm	All device links are connected. When items are added they are added to the active device link. When a failover occurs the items will be removed from the formerly active device link.
Cold	Only the active device link is connected. When items are added they are added to the active device link. When a failover occurs the items will be removed from the formerly active device link. **Note:** When a device link is a member of a **Cold** mode redundancy group and **not** the active device link, then **all** communication through that device link is disabled.

Table 18 - Redundancy Modes

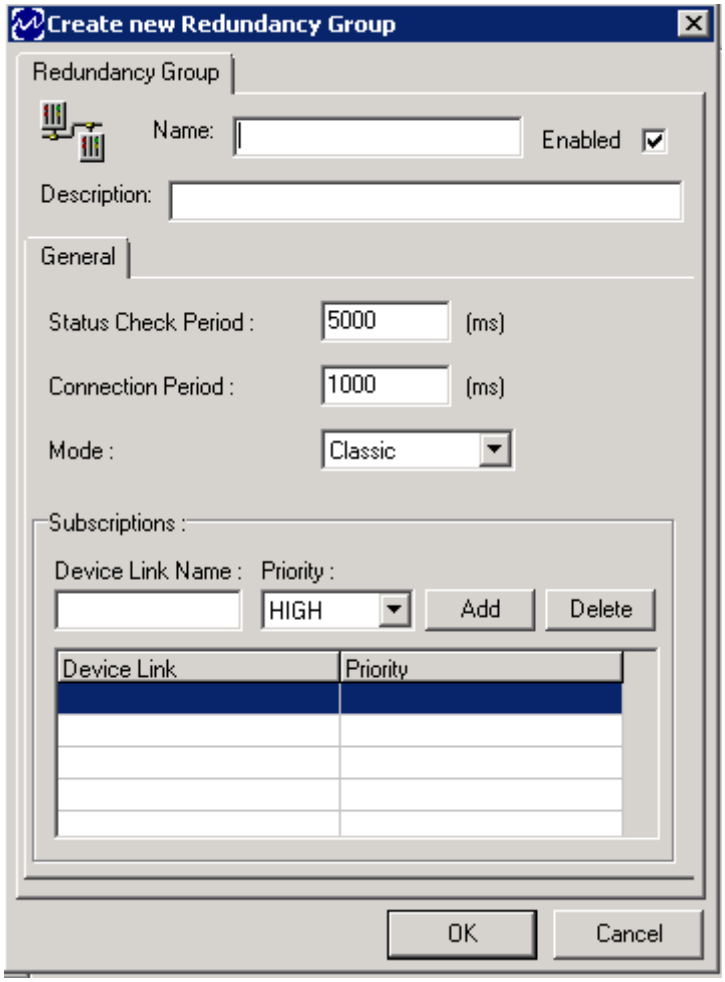


Figure 16 - Create New Redundancy Group Window

Table 19 describes the components of the **Create New Redundancy Group** window.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. The full path of the device link must be entered. Automatically converted to title case for display purposes.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank.
Status Check Period	Allows you to enter a value to define how often (in milliseconds) to check connected device links to make sure they are still actively communicating with a device. Default = ***5000*** ms.

Component	Description
Connection Period	Allows you to enter a value to define how often (in milliseconds) to check connections that have failed (i.e., communication is down) to determine if connections have been restored. Default = ***1000*** ms.
Mode	Allows you to select, from the drop-down list, the style (***Classic***, ***Hot***, ***Warm***, ***Cold***) of redundancy management that should be used by this group.
Subscriptions	This section of the **General** tab displays those device links defined as part of this particular redundant set, and the priorities of the device links. The **Subscriptions** section contains the following fields and buttons which are described below: **Device Link Name**, **Priority**, **Add**, **Delete**.
Device Link Name	Enter the name of the Device Link that you want to make redundant. The full path of the device link must be entered. For example, the device link name needs to be provided as <Network Channel>.<Network host> or <Network Channel>.<Network host>.<Device>.
Priority	Allows you to select a priority (***High***, ***Medium***, or ***Low***) for the current device link from the drop-down list. Device links of higher priority are preferred to lower priority device links.
Add	After entering a **Device Link Name** and selecting a **Priority**, use the **Add** button to add the device link to the list of redundant device links that appears at the bottom of the **General** tab. **Note:** Once you have added a device link, you can make changes to it or delete it (see **Delete** for more information). To update a device link, from the list, highlight the one you want to change. Notice that the **Add** button now reads **Update**. Make the necessary changes and click on the **Update** button.
Delete	From the list of redundant device links that appears at the bottom of the **General** tab, select the device link you want to remove and click on the **Delete** button to delete it from the list.
OK	Select this button to save any changes and close the window.
Cancel	Select this button to close the window without saving any changes made.

Table 19 - Create New Redundancy Group Window Components

To create a Redundancy Group object:

1. On the **Configuration** window, select the **Server Configuration** node and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the  icon.
2. The **Insert New Object** window (Figure 9) appears.
3. From the displayed list, select **Redundancy Group**.
4. Click on the **OK** button.

5. The **Create New** window appears (Figure 16).

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Redundancy Group** is selected, then the **Create New Redundancy Group** window is displayed.

6. From the **Create New** window, enter a name for the object.

   Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.

7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.

8. Click on the **OK** button. The object is created and appears as a child of the **Server Configuration** item as shown in Figure 17.

Figure 17 - New Redundancy Group Object

Creating and Configuring a Serial Channel Object

The **Create New** window (Figure 18) displays the settings available when creating a **Serial Channel** object.

The **Create New Serial Channel** window consists of two tabs:

- **Communication**
- **Data Acquisition**

Note: Changing the settings on this page (except for **Log bytes transmitted** and **Log bytes received**) requires that the communication channel be closed and re-opened.

Communication Tab

Figure 18 displays the **Create New Serial Channel – Communication** tab.

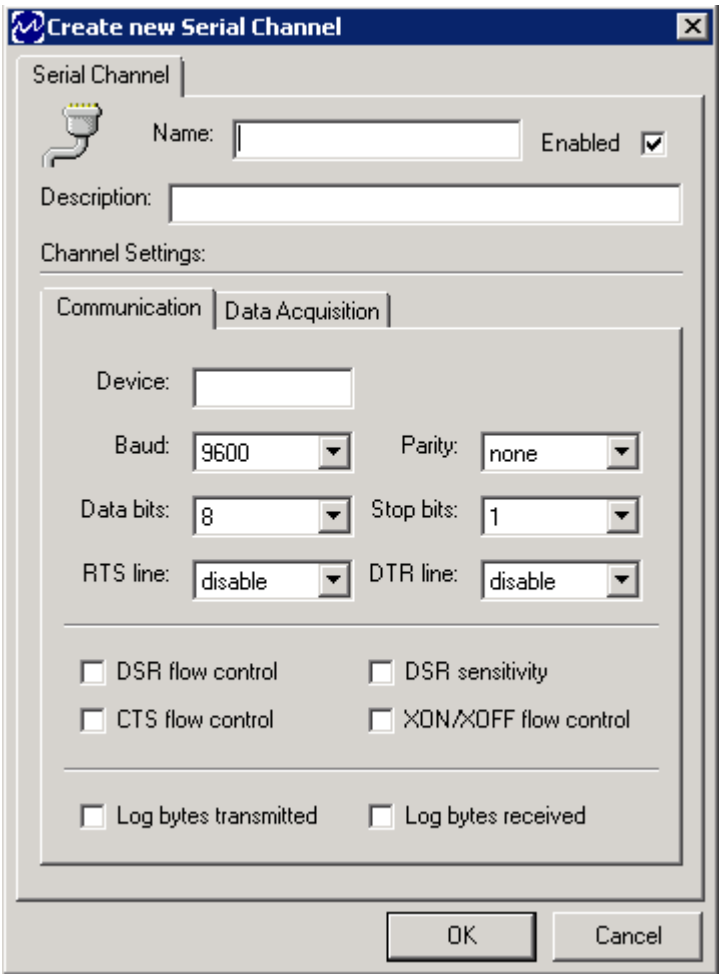


Figure 18 - Create New Serial Channel Window - Communication Tab

Table 20 describes the components of the **Create New Serial Channel – Communication** tab.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. The full path of the device link must be entered. Automatically converted to title case for display purposes. **Note:** This information stays consistent across both tabs.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected. **Note:** This information stays consistent across both tabs.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank. **Note:** This information stays consistent across both tabs.
Device	Allows you to enter a COM port number.

Component	Description
Baud	Define the communication rate by selecting a value from the drop-down list. Values range between ***110*** and ***256000***. Default = ***9600***
Parity	From the drop-down list, select the type of parity to be used. Options available are ***none***, ***odd***, ***even***, ***mark***, and ***space***. Default = ***none***
Data bits	Define the number of data bits by selecting a value from the drop-down list. Values range between ***4*** and ***8***. Default = ***8***
Stop bits	From the drop-down list, select the number of stop bits. Options available are ***1***, ***1.5***, and ***2***. Default = ***1***
RTS line	From the drop-down list, select a Request To Send (RTS) preset. Available options are ***disable***, ***enable***, ***handshake***, and ***toggle***. Default = ***disable***
DTR line	Allows you to select a Data Terminal Ready (DTR) preset from the drop-down list. Available options are ***disable***, ***enable***, and ***handshake***. Default = ***disable***
DSR flow control	This checkbox allows you to enable (i.e., select the checkbox) or disable (i.e., clear the checkbox) Data Set Ready (DSR) flow control. Default = checkbox is cleared.
DSR sensitivity	This checkbox allows you to enable (i.e., select the checkbox) or disable (i.e., clear the checkbox) Data Set Ready (DSR). Default = checkbox is cleared.
CTS flow control	This checkbox allows you to enable (i.e., select the checkbox) or disable (i.e., clear the checkbox) Clear To Send (CTS) flow control. Default = checkbox is cleared.
XON/XOFF flow control	This checkbox allows you to enable (i.e., select the checkbox) or disable (i.e., clear the checkbox) XON/XOFF software flow control. Default = checkbox is cleared.
Log bytes transmitted	Enables (i.e., check box is selected) or disables (i.e., checkbox is cleared) the logging of bytes transmitted at Medium level. Default = checkbox is cleared.
Log bytes received	Enables (i.e., check box is selected) or disables (i.e., checkbox is cleared) the logging of bytes received at Medium level. Default = checkbox is cleared.
OK	Select this button to save any changes and close the window. **Note:** This information stays consistent across both tabs.
Cancel	Closes the window without saving any changes made. **Note:** This information stays consistent across both tabs.

Table 20 - Create New Serial Channel Window - Communication Tab Components

Data Acquisition Tab

Figure 19 displays the settings available on the **Create New Serial Channel** - **Data Acquisition** tab.

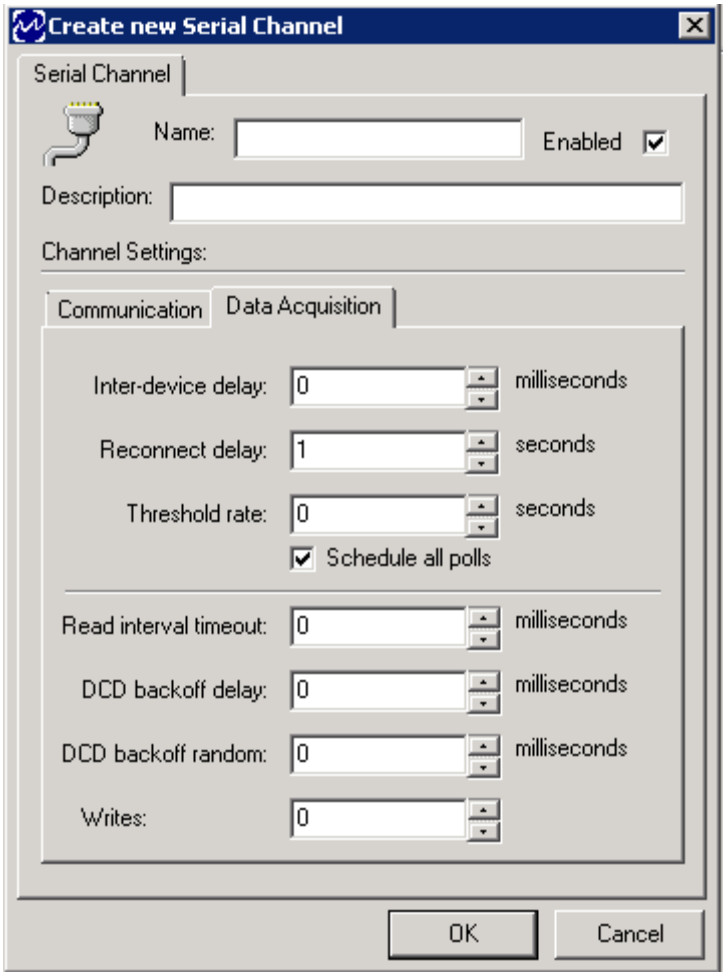


Figure 19 - Create New Serial Channel Window - Data Acquisition Tab

Table 21 describes the components of the **Create New Serial Channel – Data Acquisition** tab.

Component	Description
Inter-device delay	Allows you to enter or select a value specifying the minimum amount of time (in milliseconds) between the last request/response to/from one unit and the next request to another unit on this channel. If this option is set to 0 then it will be disabled. Default = ***0***
Reconnect delay	Allows you to enter or select a value specifying the minimum amount of time (in seconds) to wait before attempting to recover communications after the communication channel itself has failed. Default = ***1***
Threshold rate	Allows you to enter or select a value specifying the maximum achievable scan interval (in seconds) for the circuit. The server polls items with

Component	Description
	update rates faster than this on a continuous basis. Changing this option requires the server on this channel. If this is set to 0 then the option is turned off. Default = ***0*** **Note:** All units on the channel must re-optimize if this parameter changes.
Schedule all polls	When this checkbox is selected, rather than polling them continuously, the server schedules items with update rates faster than the threshold rate to poll at the threshold rate. Default = ***1 "checked"***
Read interval timeout	Allows you to enter or select a value specifying the maximum amount of time (in milliseconds) between the reception of one character and the arrival of the next. A value of ***0*** indicates no limit. Default = ***0***
DCD backoff delay	Allows you to enter or select a value specifying the minimum amount of time (in milliseconds) for DCD to be low before transmitting to avoid collisions. Set both **DCD backoff delay** and **DCD backoff random** fields to ***0*** to ignore DCD. Default = ***0***
DCD backoff random	Allows you to enter or select a value specifying the maximum amount of extra random time (in milliseconds) for DCD to be low before transmitting. Set both **DCD backoff delay** and **DCD backoff random** fields to ***0*** to ignore DCD. Default = ***0***
Writes	Allows you to enter or select a value specifying the maximum number of writes allowed before allowing a poll.

Table 21 - Create New Serial Channel Window - Data Acquisition Tab Components

To create a Serial Channel object:

1. On the **Configuration** window, select the **Server Configuration** node and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the  icon.
2. The **Insert New Object** window (Figure 9) appears.
3. From the displayed list, select the **Serial Channel** object type.
4. Click on the **OK** button.
5. The **Create New** window appears (Figure 18).

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Serial Channel** is selected, then the **Create New Serial Channel** window is displayed.
6. From the **Create New** window, enter a name for the object.

Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.

7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.

8. Click on the **OK** button. The object is created and appears as a child of the **Server Configuration** item as shown in Figure 20.

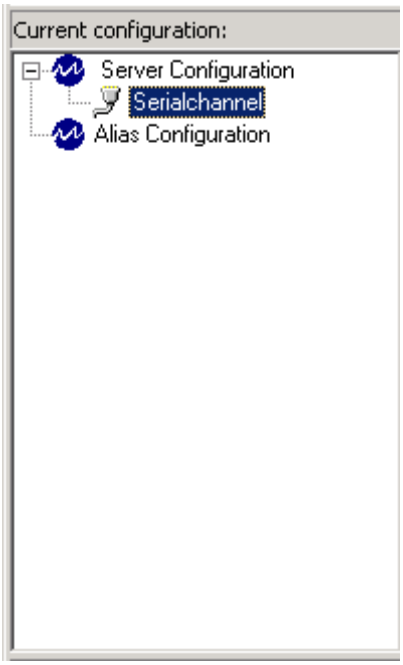


Figure 20 - New Serial Channel Object

Creating and Configuring a Network Host Object

Note: A **Network Host** can be created as a child of a **Network Channel**.

The **Create New Network Host** window (Figure 21) displays the settings available for configuring a **Network Host**.

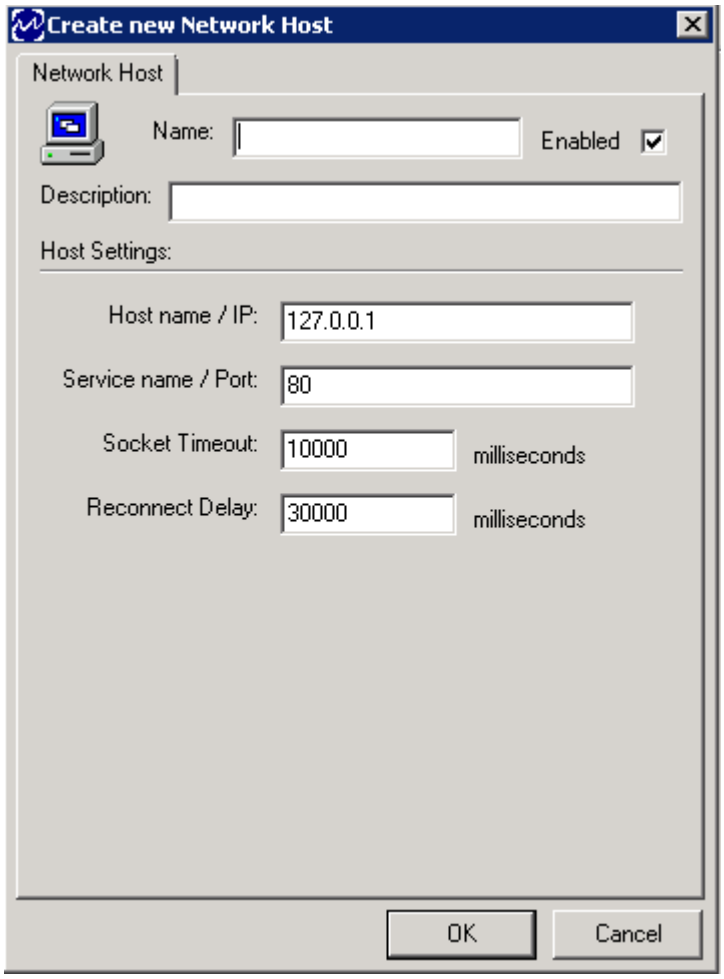


Figure 21 - Create New Network Host Window

Table 22 describes the components of the **Create New Network Host** window.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. The full path of the device link must be entered. Automatically converted to title case for display purposes.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank.
Host name/IP	Allows you to define the network name/IP address of the device. If the OPC server is to be a Modbus Master: • Enter the IP address of the Modbus unit or terminal server to which you wish to connect.

Component	Description
	If the OPC server is to be a Modbus Slave: • Leave this field blank if you have one network card. Or, • Enter the IP address of the network card you wish to have the OPC server listening on if you have more than one network card.
Service name/Port	Allows you to define the service name/port for the device. Default = ***80***.
Socket Timeout	Allows you to specify the time (in milliseconds) to wait for a response on the socket. Default = ***10000***.
Reconnect Delay	Allows you to specify the time (in milliseconds) between reconnection attempts. Default = ***30000***.
OK	Select this button to save any changes and close the window.
Cancel	Closes the window without saving any changes made.

Table 22 - Create New Network Host Window Components

To create a Network Host object:

1. On the **Configuration** window, select a previously-created **Network Channel** object (for assistance, refer to **Creating and Configuring a Network Channel Object**) and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the icon.
2. The **Insert New Object** window (Figure 22) appears.

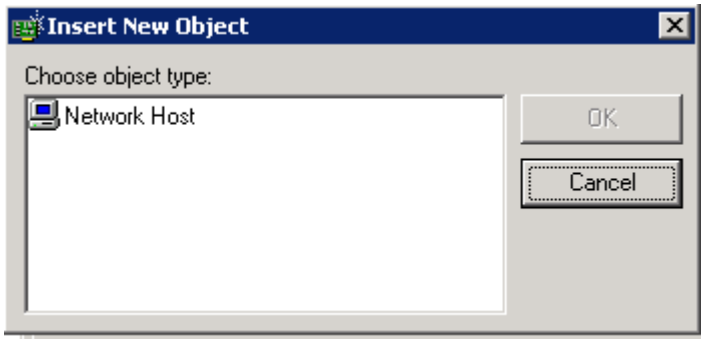


Figure 22 - Insert New Object

3. From the displayed list, select **Network Host**.
4. Click on the **OK** button.
5. The **Create New** window appears (Figure 21).

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Network Host** is selected, then the **Create New Network Host** window is displayed.

6. From the **Create New** window, enter a name for the object.

   Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.

7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.

8. Click on the **OK** button. The object is created and appears as a child of the **Network Channel** item as shown in Figure 23.

Figure 23 - New Network Host Object

Creating and Configuring a Server Status List Object

Some OPC clients use the OPC call, ***GetServerStatus()***, to monitor the state of an OPC server. MatrikonOPC's Server Status List object makes it possible to monitor an OPC server status in a customizable way.

The Server Status List object contains a list of subscribed device links that can exist anywhere in the OPC server hierarchy. The status of the server is checked every configured status check period by examining the status of the subscribed device links. Device links can be added to the Server Status List in one of two ways:

1. **Direct** – the device link is added directly to the subscription list.
2. **Parent** – the device link's sub-ordinates are added to the subscription list.

There are two options for the logic that will determine the server status. For the **Server Status** to be ***Running***, the first option requires that there is at least one subscribed device link that is running. In this case, if all subscribed device links enter a failure condition (e.g., become disconnected), the **Server Status** will be ***Failed***. The second option is stricter as it is required that all subscribed device links be running. In this case, if any subscribed device links enter a failure condition, the **Server Status** will be ***Failed***.

There can be one or more device links contained within a Server Status List.

The **Create New Server Status List** window(Figure 24) displays the settings available for configuring a Server Status List object.

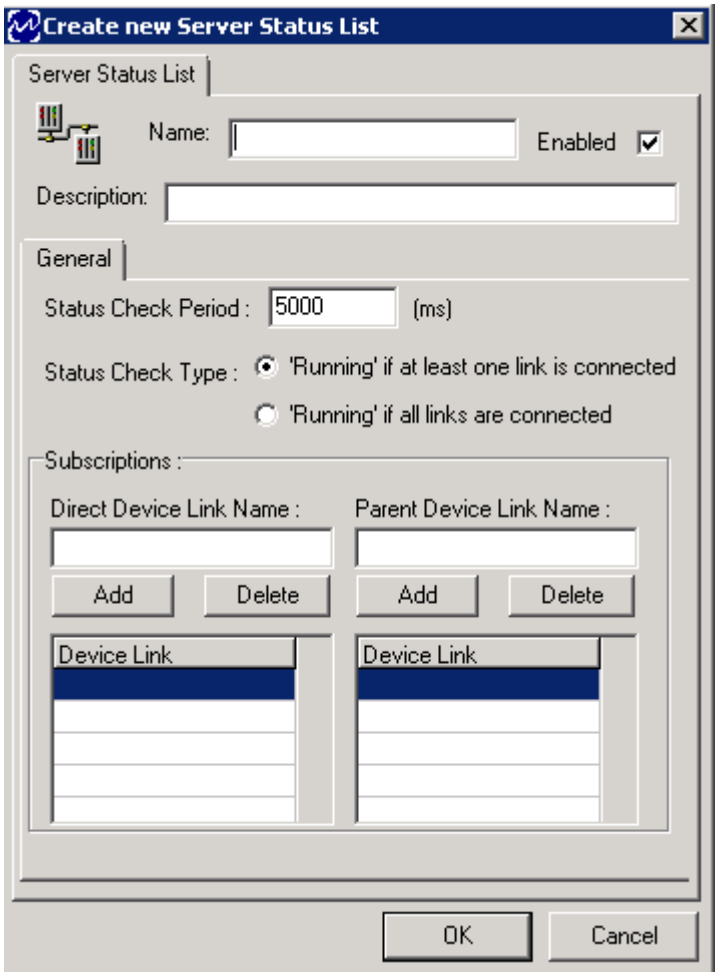


Figure 24 - Create New Server Status List Window

Table 23 describes the components of the **Create New Server Status List** window.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. Automatically converted to title case for display purposes.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank.
Status Check Period	Allows you to enter a value to define how often (in milliseconds) to check connected device links to make sure they are still actively

Component	Description
	communicating with a device. Default = ***5000*** ms.
Status Check Type	Allows you to select the type of logic used to determine **Server Status**. Selecting the first option, ***'Running' if at least one link is connected***, results in a Server Status of ***Running*** (rather than ***Failed***) if there are one or more device links with a status of ***Running***. Selecting the second option, ***'Running' if all links are connected***, results in a **Server Status** of ***Failed*** if there are one or more device links with a status of ***Failed***.
Subscriptions	This section of the **General** tab displays those device links defined as part of the server status list. The **Subscriptions** section contains the following fields and buttons which are described below: **Direct Device Link Name**, **Parent Device Link Name**, **Add**, **Delete**.
Direct Device Link Name	Allows you to enter the name of the device link that is going to be added to the list of Direct Device Links. These device links have their state monitored directly. The full path of the device link must be entered.
Parent Device Link Name	Allows you to enter the name of the device link that is going to be added to the list of Parent Device Links. The subordinates of these device links will have their state monitored directly. The full path of the parent device link must be entered. **Note:** Only immediate subordinates are monitored. If a parent device link named **Parent** has a subordinate **Child** that is the parent of another device link, **Grandchild**, only the **Child** device link is monitored. For **Grandchild** to be monitored, it may be added to the Direct Device Link list, or **Child** may be added to the Parent Device Link list.
Add	After entering a device link name into either the Direct Device Link Name field, or the Parent Device Link Name field, use the **Add** button for that particular list to add the device link to the list. **Note:** Once you have added a device link, you can make changes to it or delete it (see **Delete** for more information). To update a device link, from the list, highlight the one you want to change. Notice that the **Add** button now reads **Update**. Make the necessary changes and click on the **Update** button.
Delete	Select the device link you want to remove from a particular list (either a Direct Device Link Name, or a Parent Device Link Name) and click on the **Delete** button.
OK	Select this button to save any changes and close the window.
Cancel	Select this button to close the window without saving any changes made.

Table 23 - Create New Server Status List Window Components

To create a Server Status List object:

1. On the **Configuration** window, select the **Server Configuration** node and either:

- Right-click your mouse and select **Define New** from the menu that appears, or
- Select the **Define New** menu option from the **Edit** menu, or
- Click on the  icon.

2. The **Insert New Object** window (Figure 9) appears.
3. From the displayed list, select **Server Status List**.
4. Click on the **OK** button.
5. The **Create New** window appears

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Server Status List** is selected, then the **Create New Server Status List** window is displayed.

6. From the **Create New** window, enter a name for the object.

   Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.

7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.

8. Click on the **OK** button. The object is created and appears as a child of the **Server Configuration** item as shown in Figure 25.

Figure 25 - New Server Status List Object

Changing Objects

To change an object's configuration:

1. Once an object is created, select it in the tree view (i.e., **Current configuration**) panel.

2. In the configuration panel (i.e., pane on the right), make the applicable changes.
3. Click on the **Apply** button to accept the changes, or click on the **Cancel** button to discard them.

Removing Objects

Note: Users can easily remove any object that they have created, but a warning message (Figure 26) will appear when users remove an object that has data items to which one or more OPC clients hold reference.

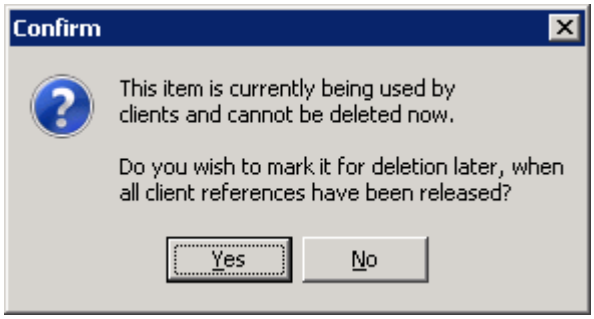


Figure 26 - Client Reference Warning Message

To remove an object:

1. In the **Configuration** window, under **Server Configuration**, select the object to be removed.
2. From the **Edit** menu, select the **Delete** menu option (or click on the ✕ button).

   Note: If one or more OPC clients hold references to data items somewhere under the selected object, a warning message (Figure 26) appears.
3. To mark the object for later deletion, click on the **Yes** button. To leave the object unmodified, click on the **No** button.

   Note: Items marked for deletion are disabled and a red X appears over the icon on the right side of the window when the object is selected.

Creating and Configuring a Dial-Up Connection Object

Note: A **Dial-up Connection** object can be created as a child of a **Network Host** or a **Serial Channel**. For more information, refer to **Creating and Configuring a Network Host Object** or **Creating and Configuring a Serial Channel Object**.

The **Create New** window (Figure 27) displays the settings available when creating a **Dial-up Connection** object.

The **Create New Dial-up Connection** window consists of three tabs:

- **Dial-up**
- **Modem**
- **Advanced**

Dial-Up Tab

Figure 27 displays the **Create New Dial-up Connection – Dial-up** tab.

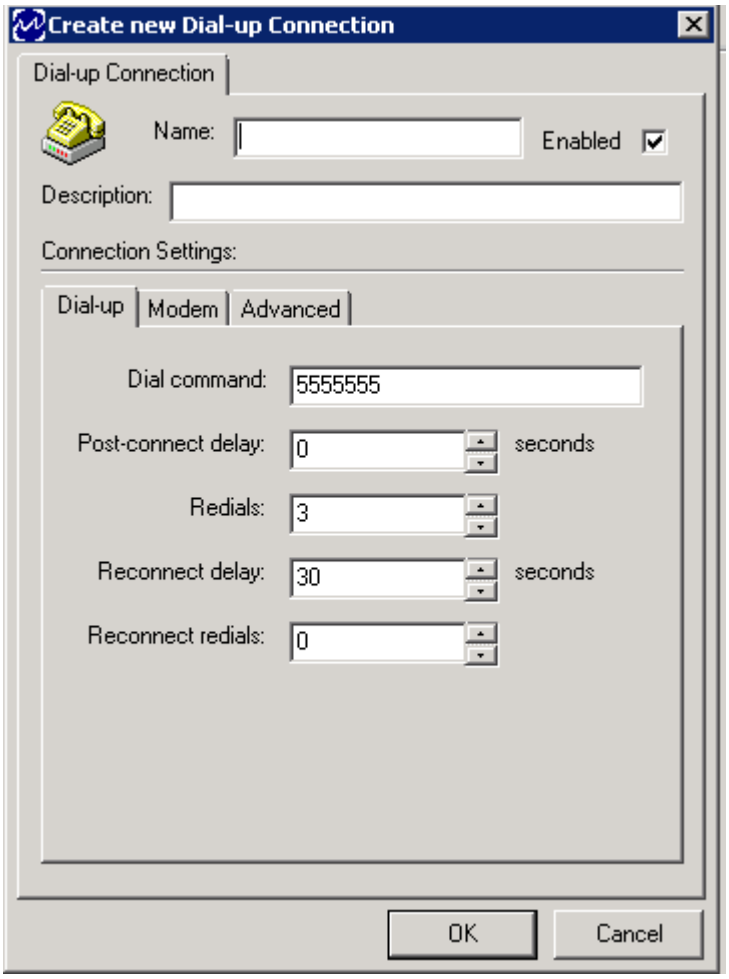


Figure 27 - Create New Dial-Up Connection Window - Dial-Up Tab

Table 24 describes the components of the **Create New Dial-up Connection – Dial-up** tab.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. The full path of the device link must be entered. Automatically converted to title case for display purposes. **Note:** This information stays consistent across all tabs.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected. **Note:** This information stays consistent across all tabs.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank. **Note:** This information stays consistent across all tabs.
Dial command	Allows you to enter any valid sequence of characters that can follow an

Component	Description
	ADT command.
Post-connect delay	Allows you to enter or select a value specifying the minimum amount of time (in seconds) to wait after establishing a dial-out connection before transmitting. Default = ***0***.
Redials	Allows you to enter or select a value defining the maximum number of times to redial a connection before concluding that communication has failed with the units on that connection. Default = ***3***.
Reconnect delay	Allows you to enter or select a value specifying the minimum amount of time (in seconds) to wait before attempting to recover communications after communication has failed with the connection. Default = ***30***.
Reconnect redials	Allows you to enter or select a value defining the maximum number of times to redial when trying to reconnect after communication has failed with a dial-up connection. Default = ***0***.
OK	Select this button to save any changes and close the window. **Note:** This information stays consistent across all tabs.
Cancel	Closes the window without saving any changes made. **Note:** This information stays consistent across all tabs.

Table 24 - Create New Dial-Up Connection Window - Dial-Up Tab Components

Modem Tab

Figure 28 displays the settings available on the **Create New Dial-up Connection** - **Modem** tab.

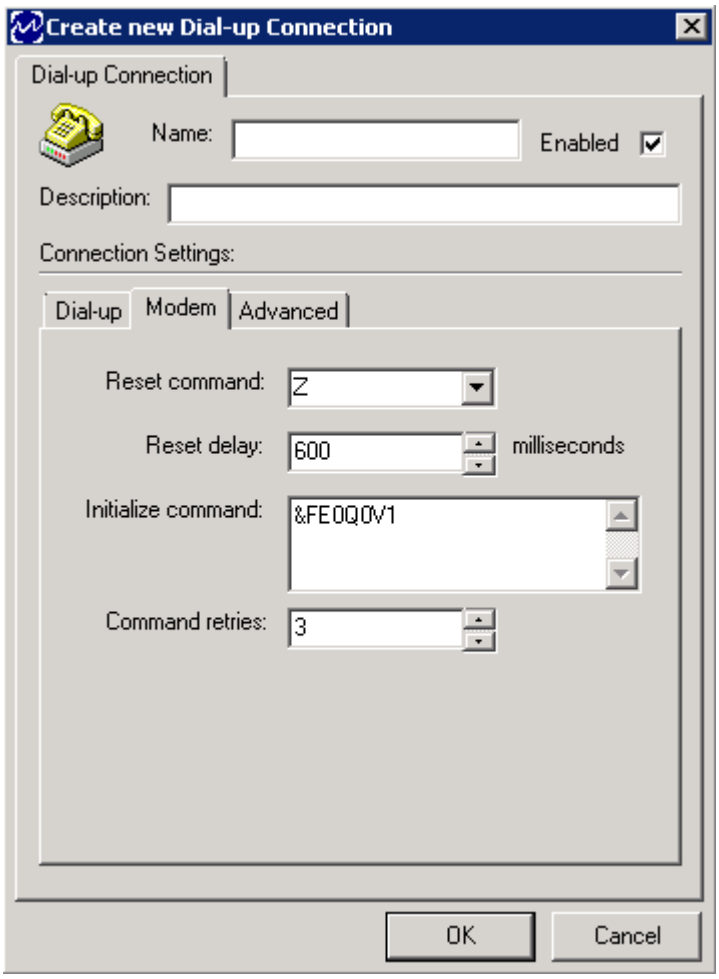


Figure 28 - Create New Dial-Up Connection Window - Modem Tab

Table 25 describes the components of the **Create New Dial-Up Connection – Modem** tab.

Component	Description
Reset command	Allows you to select, from the drop-down list, a modem reset command. This command is sent before initialization commands. Default = ***Z***.
Reset delay	Allows you to enter or select a value specifying the amount of time (in milliseconds) to wait after the reset command completes. Default = ***600***.
Initialize command	Allows you to enter modem initialization command or commands, issued before dialling out and after hanging up. The command should always end with ***E0Q0V1***.
Command retries	Allows you enter or select a value to define the maximum number of times to resend a command wait for a response before concluding that communication has failed with the modem. Default = ***3***.

Table 25 - Create New Dial-Up Connection Window - Modem Tab Components

Advanced Tab

Figure 29 displays the settings available on the **Create New Dial-up Connection – Advanced** tab.

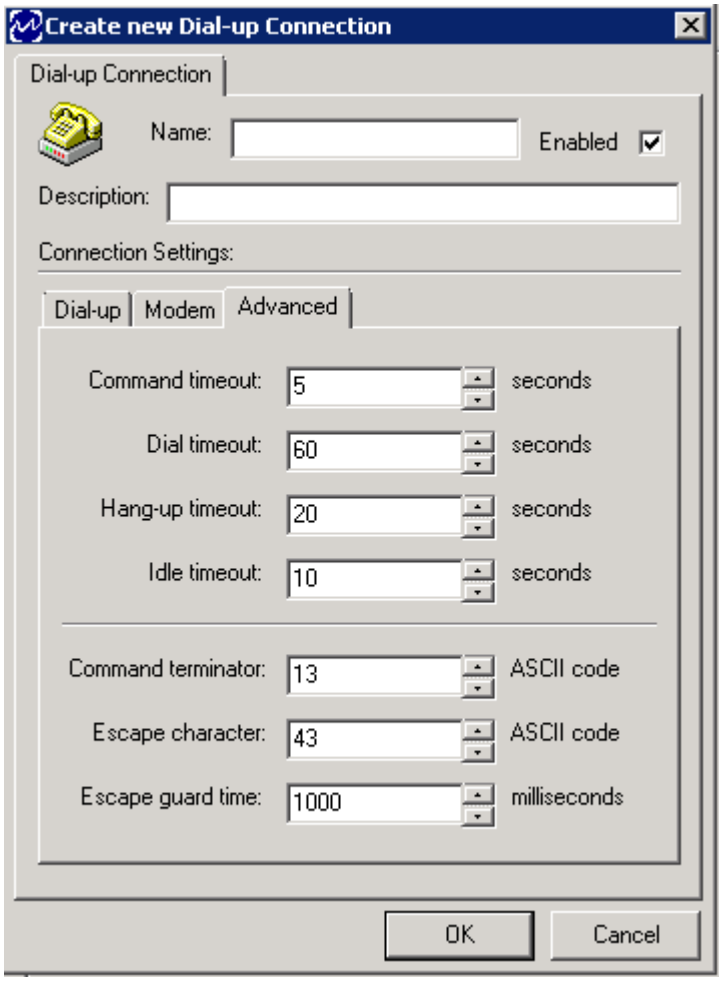


Figure 29 - Create New Dial-Up Connection Window - Advanced Tab

Table 26 describes the components of the **Create New Dial-up Connection** - **Advanced** tab.

Component	Description
Command timeout	Allows you to enter or select a value defining the maximum amount of time (in seconds) to wait for the modem to respond to AT commands with a valid result code. Default = ***5***.
Dial timeout	Allows you to enter or select a value defining the maximum amount of time (in seconds) to wait for the modem to receive a carrier from the dialled connection before aborting the call. Default = ***60***.
Hang-up timeout	Allows you to enter or select a value defining the maximum amount of time (in seconds) to wait for the modem to hang up after issuing the ATH0 command. Default = ***20***.

Component	Description
Idle timeout	Allows you to enter or select a value defining the maximum amount of time (in seconds) to allow an idle connection to continue (i.e., no communication) before hanging up. Default = ***10***.
Command terminator	Allows you to enter or select a value defining an ASCII character to be used to terminate commands and responses (usually a carriage return). Default = ***13***.
Escape character	Allows you to enter or select a value defining an ASCII character that is sent three times to escape into command mode (usually a plus sign). Default = ***43***.
Escape guard time	Allows you to enter or select a value defining the amount of time (in milliseconds) to wait before and after sending the escape sequence. Default = ***1000***.

Table 26 - Create New Dial-Up Connection Window - Advanced Tab Components

To create a Dial-up Connection object:

1. On the **Configuration** window, select a previously-created **Network Host** or a previously-created **Serial Channel** (for assistance, refer to **Creating and Configuring a Network Host Object** or **Creating and Configuring a Serial Channel Object**) and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the icon.
2. The **Insert New Object** window (Figure 30) appears.

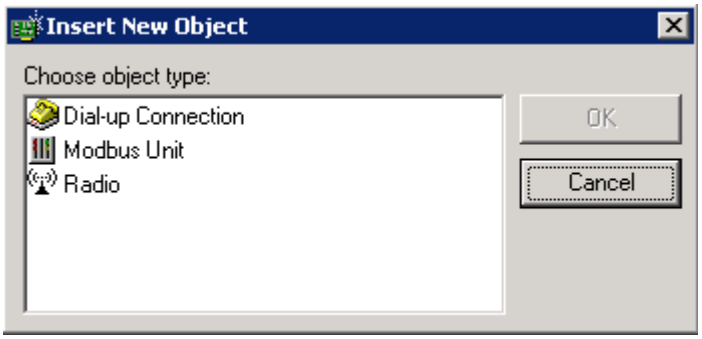


Figure 30 - Insert New Object Window

3. From the displayed list, select the **Dial-up Connection** object type.
4. Click on the **OK** button.
5. The **Create New** window appears (Figure 27).

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Dial-up Connection** is selected, then the **Create New Dial-up Connection** window is displayed.
6. From the **Create New** window, enter a name for the object.

Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.

7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.

8. Click on the **OK** button. The object is created and appears either as a child of the **Network Host** or **Serial Channel** item as shown in Figure 31 (depending on what the selected parent item was).

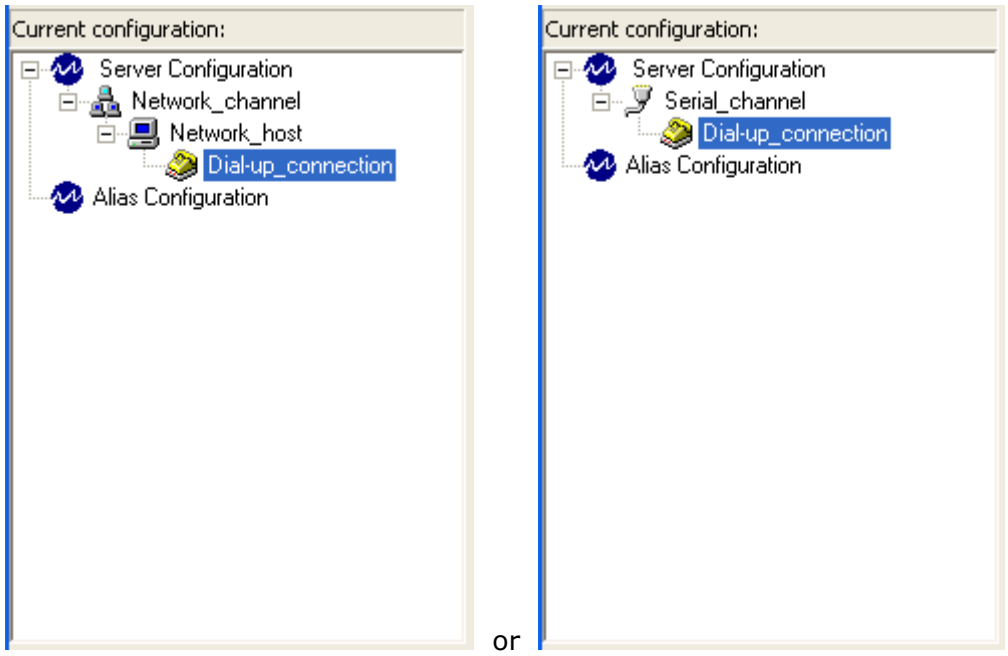


Figure 31 - New Dial-Up Connection Object

Creating and Configuring a Modbus Unit Link

Note: A **Modbus Unit** link can be created as a child of any of the following items:

- **Network Host** (**Creating and Configuring a Network Host Object**)
- **Dial-up Connection** (**Creating and Configuring a Dial-up Connection Object**)
- **Serial Channel** (**Creating and Configuring a Serial Channel Object**)
- **Radio Link** (**Creating and Configuring a Radio Link**)

The **Create New** window (Figure 32) displays the settings available when creating a **Modbus Unit** link.

The **Create New Modbus Unit** window consists of three tabs:

- **Communication**
- **Data Acquisition**
- **Optimization**

Communication Tab

Figure 32 displays the **Create New Modbus Unit – Communication** tab.

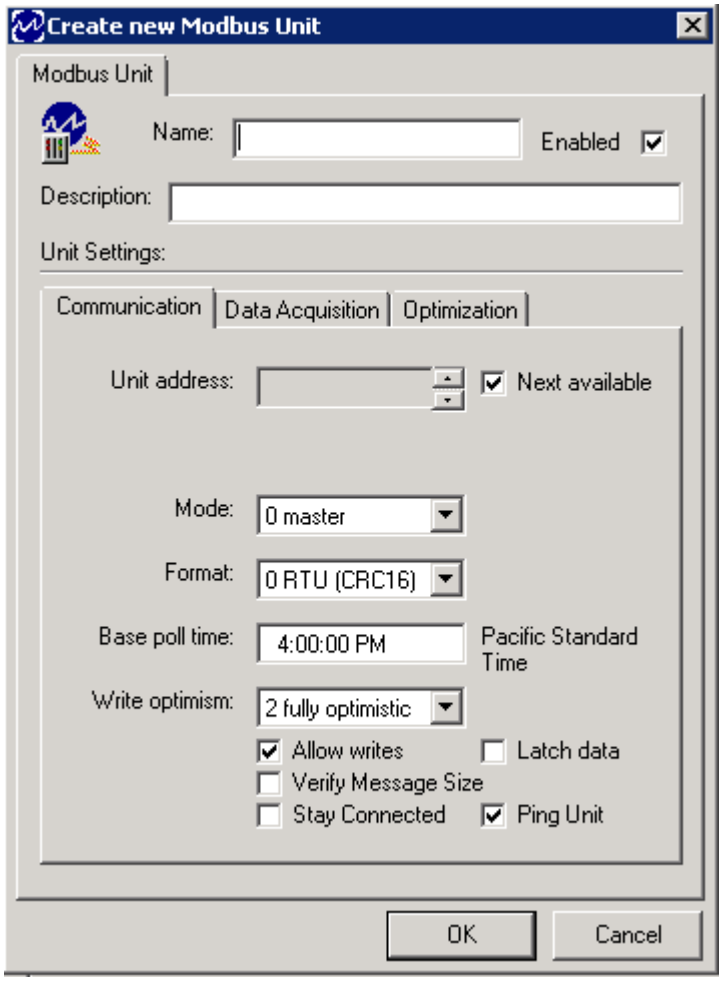


Figure 32 - Create New Modbus Unit Window - Communication Tab

Table 27 describes the components of the **Create New Modbus Unit – Communication** tab.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. The full path of the device link must be entered. Automatically converted to title case for display purposes. **Note:** This information stays consistent across all tabs.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected. **Note:** This information stays consistent across all tabs.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank.

Component	Description
	Note: This information stays consistent across all tabs.
Unit address	Allows you to enter or select a value defining a Modbus slave address (***1***-***255***). A value of ***0*** indicates a broadcast unit (write-only). Changing this option requires the server to reset communication with this unit. This field is not available (i.e., is greyed out) when the **Next available** checkbox is selected.
Next available	When this checkbox is selected, the unit defaults to the next available Modbus address on this channel. When this checkbox is selected, the **Unit address** field is disabled (i.e., greyed out).
Mode	Allows you to select, from the drop-down list, a mode indicating whether the server communicates with this unit as a master or by emulating a slave. Changing this option requires the server to reset communication with this unit. Default = ***0 master***. Refer to **Emulating Slave Mode** for more information about this field.
Format	Allows you to select, from the drop-down list, a message format to use for communication. Changing this option requires the server to reset communication with this unit. Default = ***0 RTU (CRC16)***.
Base poll time	Allows you to enter a normalization time for scheduling polls, stored as UTC. Changing this option requires the server to re-optimize communication with this unit.
Write optimism	Allows you to select, from the drop-down list, a level of "optimism" for writing control values to the unit: ***0 pessimistic***, ***1 semi-optimistic***, and ***2 fully optimistic***. If the unit is configured for either ***fully optimistic*** or ***semi-optimistic*** writes, then the server copies the written value to the item's cache with either a "good" or a "bad" quality (depending on the result of the write operation) after receiving the response from the device. However, if the unit is configured for ***fully optimistic*** writes, then the server immediately copies the written value to the item's cache with an "uncertain" quality before sending the write request. If the unit is configured for ***pessimistic*** writes, the server does not alter the item's cache during the write operation. The server updates that value for the item on the next read. Default = ***2 fully optimistic***.
Allow writes	Selecting this checkbox allows writes to the unit. Writes are not allowed when the checkbox is cleared. By default, this checkbox is selected.
Latch data	Select this checkbox to have items "freeze" at their last value/quality/timestamp when the unit is set off-scan. By default, this checkbox is cleared.
Verify Message Size	When this checkbox is selected, the OPC server verifies that the response message has the correct length before accepting it.

Component	Description
	By default, this checkbox is cleared.
Stay Connected	If that option is checked, the device link always stays connected even if items from that device link are not currently being scanned by an OPC client. **Note:** It is required that this option be selected if this unit is being used in a **Redundancy Group** configuration. By default, this checkbox is cleared.
Ping Unit	Select this checkbox to ping the unit on connection. By default, this checkbox is selected.
OK	Select this button to save any changes and close the window. **Note:** This information stays consistent across all tabs.
Cancel	Closes the window without saving any changes made. **Note:** This information stays consistent across all tabs.

Table 27 - Create New Modbus Unit Window - Communication Tab Components

Emulating Slave Mode

Some Modbus devices act as Master and the server needs to act as a slave to be able to communicate with them. This can be done by selecting ***1 slave*** from the drop-down list in the **Mode** field on the **Communication** tab (Figure 32). When this field is set to ***Slave*** mode, two extra fields appear (highlighted in Figure 33), both of which are described in Table 28: **Node Name**, **Use Name**.

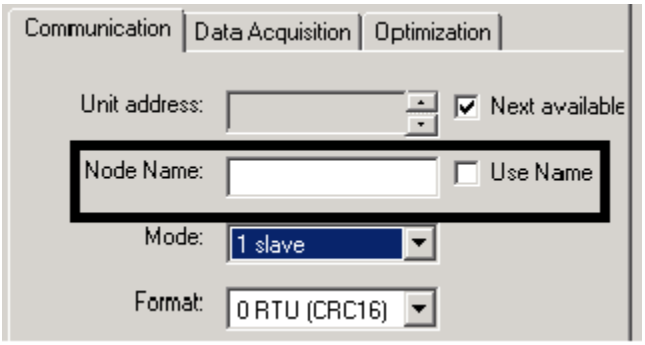


Figure 33 - Slave Mode Options

Component	Description
Node Name	This field is used to assign a unique address space for the Slave device link to store data. This option can be used when there are more than 255 Slave unit device links configured and a unique address is needed for every Slave device link. **Note:** If there are two unit Slave configurations in the OPC server that have the same Node Name, then they will share the same address space. This means that if one device does a write to this unit in the OPC server then any other Slave configured this way also displays the new value.
Use Name	This checkbox enables the ***Node Name*** to be edited.

Table 28 - Slave Mode Options Components

Note: The **Host name** field can be left blank in the **Network Host** device link while using the Slave emulation mode. The Slave address is made to be the Local IP Address of the PC for which the server is installed.

Data Acquisition Tab

Figure 34 displays the settings available on the **Create New Modbus Unit** – **Data Acquisition** tab.

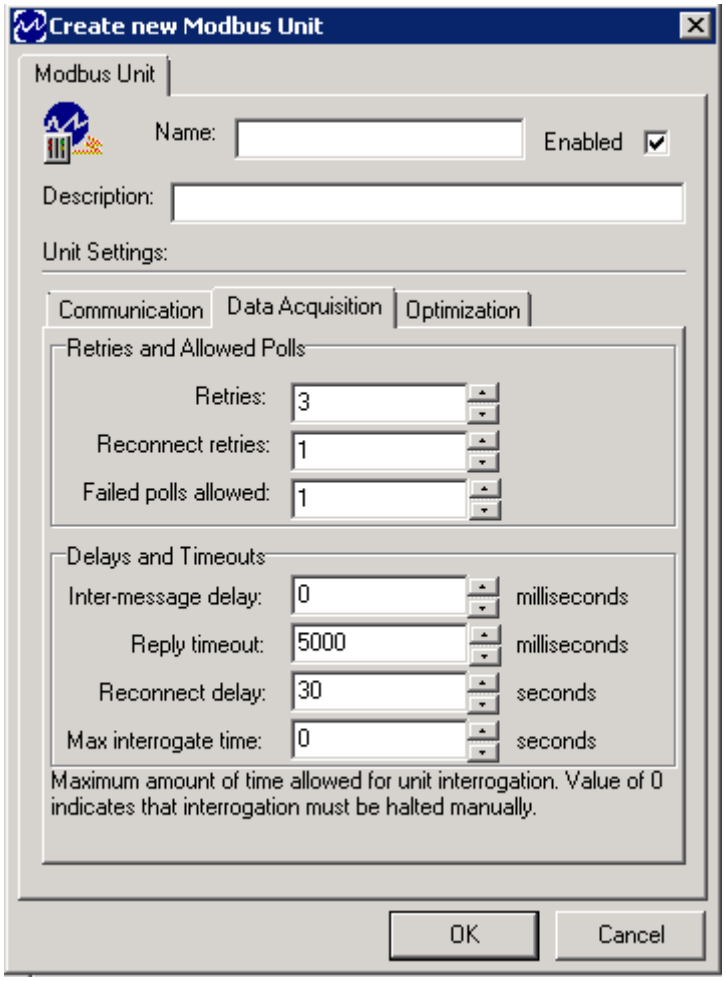


Figure 34 - Create New Modbus Unit Window - Data Acquisition Tab

Table 29 describes the components of the **Create New Modbus Unit – Data Acquisition** tab.

Component	Description
Retries	Allows you to enter or select a value defining the maximum number of times to resend a request and wait for a response before concluding that communication has failed with the unit. Default = ***3***.
Reconnect retries	Allows you to enter or select a value defining the maximum number of times to resend a request and wait for a response when trying to reconnect after communication has failed with a unit. Default = ***1***.

Component	Description
Failed polls allowed	Allows you to enter or a select a value defining the maximum number of polls allowed to elapse after the first communication failure before the quality goes to bad and considers the device link as having an ongoing communication failure. For more information, refer to **Appendix L – SCADA Servers Connection Logic**. Default = ***1***.
Inter-message delay	Allows you to enter or select a value defining the minimum amount of time (in milliseconds) between the last request/response to/from this unit and the next request to this same unit. **Note:** If this is set to ***0*** then the setting will be disabled. Default = ***0***.
Reply timeout	Allows you to enter or select a value defining the maximum amount of time (in milliseconds) to wait for the response to a request. **Note:** If this setting is too short, communication can become confused. Default = ***5000***.
Reconnect delay	Allows you to enter or select a value defining the minimum amount of time (in seconds) to wait before attempting to reconnect after communication has failed with a unit. Default = ***30***.
Max interrogate time	Allows you to enter or select a value defining the maximum amount of time (in seconds) to allow a unit to be interrogated. A value of ***0*** indicates that interrogation must be stopped manually. Default = ***0***.

Table 29 - Create New Modbus Unit Window - Data Acquisition Tab Components

Optimization Tab

Figure 35 displays the **Create New Modbus Unit – Optimization** tab.

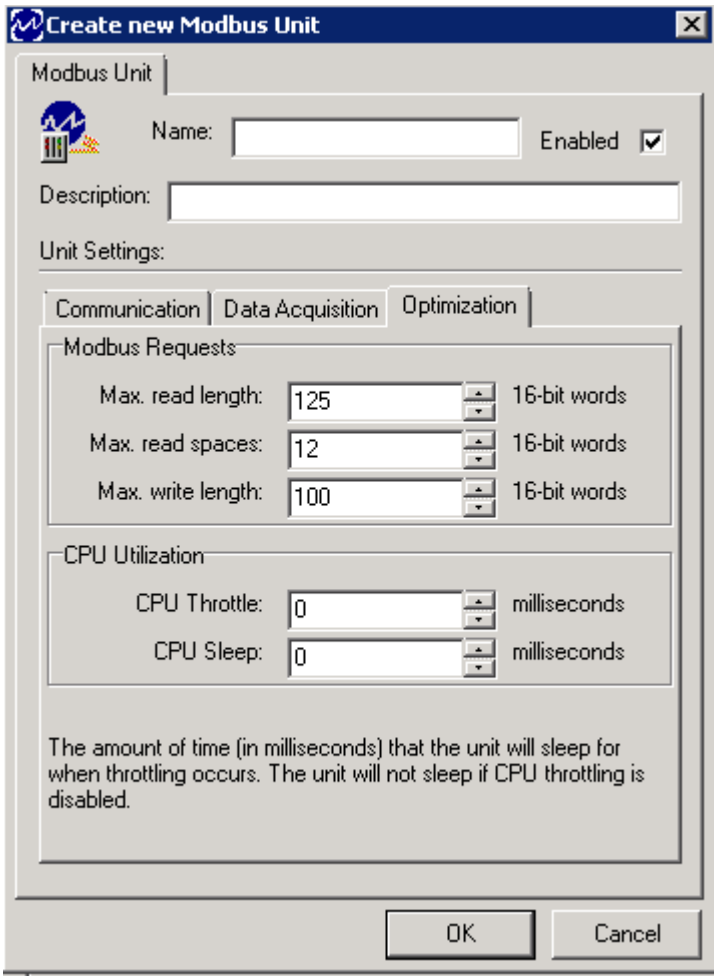


Figure 35 - Create New Modbus Unit Window - Optimization Tab

Table 30 describes the components of the **Create New Modbus Unit – Optimization** tab.

Component	Description
Max. read length	Allows you to enter or select a value defining the maximum number of registers (or blocks of 16 coils) per read request. **Note:** Changing this option requires the server to re-optimize communication with this unit. Default = ***125***.
Max. read spaces	Allows you to enter or select a value defining the maximum allowable number of consecutive inactive registers (or blocks of 16 coils) to include in a block read for optimization. **Note:** Changing this option requires the server to re-optimize communication with this unit. Default = ***12***.
Max. write length	Allows you to enter or select a value defining the maximum number of registers (or blocks of 16 coils) per write request. Set to ***1*** to force single writes. If set to ***1***, then **FC5** and **FC6** is used. **Note:** Changing this option requires the server to re-optimize

Component	Description
	communication with this unit. Default = ***100***.
CPU Throttle	CPU throttling is enabled if the unit is scanned faster than this time (in milliseconds). CPU throttling is disabled if this value is ***0***. In certain scenarios, CPU usage may be excessively high because scanning threads will use a majority of a machine's CPU time. This option will allow the Modbus Unit to sleep and reduce its CPU usage. A value of ***1000*** is typical in this situation. **Note:** Enabling CPU throttling may decrease performance Default = **0**.
CPU Sleep	The amount of time (in milliseconds) that the Modbus Unit will sleep for when throttling occurs. The unit will not sleep if CPU throttling is disabled. A value of ***10*** is typically sufficient. **Note:** A higher value will decrease CPU usage at the cost of performance Default = **0**.

Table 30 - Create New Modbus Unit Window - Optimization Tab Components

To create a Modbus Unit link:

1. On the **Configuration** window, select a previously-created **Network Host**, **Dial-up Connection**, **Serial Channel**, or **Radio** link, and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the  icon.
2. The **Insert New Object** window (Figure 30) appears.
3. From the displayed list, select the **Modbus Unit** object type.
4. Click on the **OK** button.
5. The **Create New** window appears (Figure 32).

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Modbus Unit** is selected, then the **Create New Modbus Unit** window is displayed.
6. From the **Create New** window, enter a name for the object.

   Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link is not created. The **Description** field is optional.
7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.
8. Click on the **OK** button. The object is created and appears either as a child of the selected item.

Creating and Configuring a Radio Link

Note: A **Radio** link can be created as a child of a **Serial Channel**.

The **Create New Radio** window (Figure 36) displays the settings available for configuring a **Radio** link.

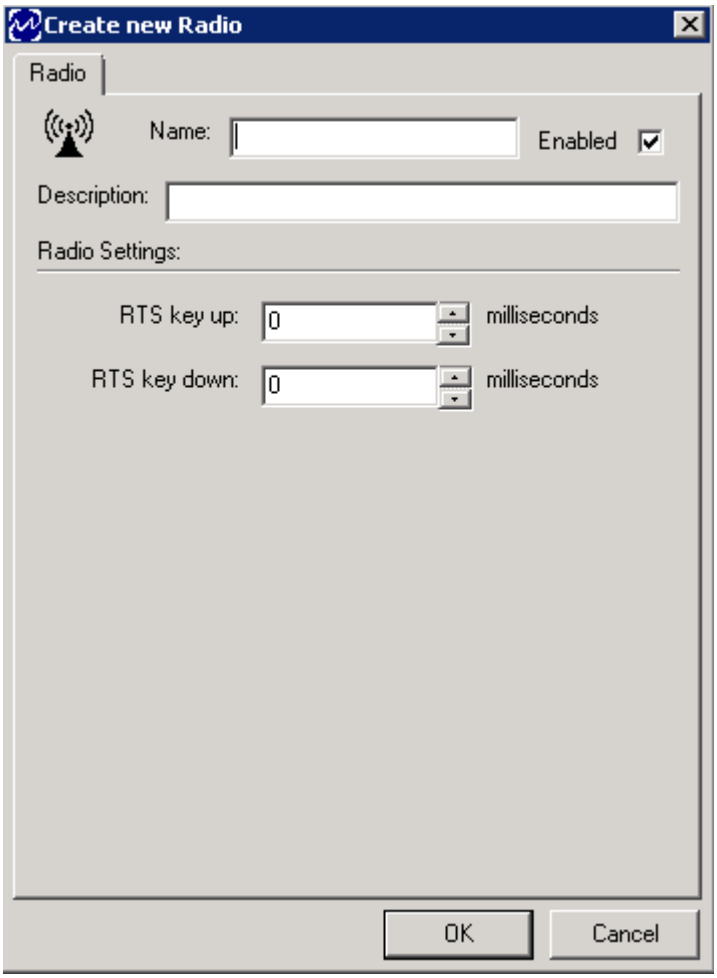


Figure 36 - Create New Radio Window

Table 31 describes the components of the **Create New Radio** window.

Component	Description
Name	Allows you to enter the name of the device link, which becomes an element of the item ID for data items. The full path of the device link must be entered. Automatically converted to title case for display purposes.
Enabled	Use this checkbox to enable (i.e., checkbox is selected) or disable (i.e., checkbox is cleared) communication for the object. By default, the checkbox is selected.
Description	This field is optional and takes any user-defined text (64-character maximum) for documentation purposes. By default, this field is blank.

Component	Description
RTS key up	Allows you to enter or select a value defining the minimum amount of time (in milliseconds) to wait after setting the RTS signal to allow the transceiver to "warm up". Default = ***0***.
RTS key down	Allows you to enter or select a value defining the minimum amount of time (in milliseconds) to wait before clearing the RTS signal to allow the transceiver to "cool down". Default = ***0***.
OK	Select this button to save any changes and close the window.
Cancel	Closes the window without saving any changes made.

Table 31 - Create New Radio Window Components

To create a Radio link:

1. On the **Configuration** window, select a previously-created **Serial Channel** object (for assistance, refer to **Creating and Configuring a Serial Channel Object**) and either:
   - Right-click your mouse and select **Define New** from the menu that appears, or
   - Select the **Define New** menu option from the **Edit** menu, or
   - Click on the  icon.
2. The **Insert New Object** window (Figure 30) appears.
3. From the displayed list, select **Radio**.
4. Click on the **OK** button.
5. The **Create New** window appears (Figure 36).

   Note: The item selected in the **Insert New Object** window determines which **Create New** window is displayed. For example, when **Radio** is selected, then the **Create New Radio** window is displayed.
6. From the **Create New** window, enter a name for the object.

   Note: An error message is displayed when a valid name is not entered. If the entered name is a duplicate of an existing device link, the new device link will not be created. The **Description** field is optional.
7. Edit the configuration components as desired.

   Note: For the object to be used for communication, it must be enabled (i.e., the **Enabled** checkbox is selected). This can be done at any time.
8. Click on the **OK** button.
9. The object is created and appears as a child of the **Serial Channel** item as show in Figure 37.

Figure 37 - New Radio Link

Changing Objects

To change an object's configuration:

1. Once an object is created, select it in the tree view (i.e., **Current configuration**) panel.
2. In the configuration panel (i.e., pane on the right), make the applicable changes.
3. Click on the **Apply** button to accept the changes, or click on the **Cancel** button to discard them.

Removing Objects

Note: Users can easily remove any object that they have created, but a warning message (Figure 38) will appear when users remove an object that has data items to which one or more OPC clients hold reference.

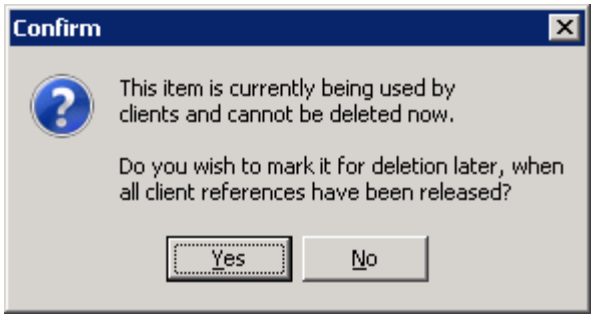


Figure 38 - Client Reference Warning Message

To remove an object:

1. In the **Configuration** window, under **Server Configuration**, select the object to be removed.
2. From the **Edit** menu, select the **Delete** menu option (or click on the ✕ button).

Note: If one or more OPC clients hold references to data items somewhere under the selected object, a warning message (Figure 38) appears.

3. To mark the object for later deletion, click on the **Yes** button. To leave the object unmodified, click on the **No** button.

   Note: Items marked for deletion are disabled and a red X appears over the icon on the right side of the window when the object is selected.

Alias Configuration

MatrikonOPC servers, including this one, provide the ability to create user-defined aliases that can be used in place of regular OPC items. This feature is particularly useful when the item path for a given server is very complex or difficult to remember, for example: *Com1.Radio1.Unit1.41.4.123*.

Servers can also be configured so that client applications have access to configured aliases only, rather than every available item.

This section of the manual describes how to create and edit aliases using the **Configuration** window.

Note: It is not necessary to define aliases before accessing server data items from an OPC client.

Inserting Alias Groups

Note: Alias Groups are used to contain one or more configured aliases.

To insert an Alias Group:

1. On the **Configuration** window, select **Alias Configuration** or a previously configured **Alias Group**.
2. From the **Edit** menu, choose **Insert Alias Group** or click on the button.
3. A new **Alias Group** folder appears.
4. Enter a new name for the **Alias Group**, and press **Enter**.
5. The **Alias Group** is renamed.

Renaming Alias Groups

Note: Alias Groups can be renamed after they have been created.

To rename an Alias Group:

1. On the **Configuration** window, select an **Alias Group**.
2. From the **Edit** menu, choose **Rename** or click on the button.
3. The **Alias Group** is renamed.

Inserting New Aliases

Use the **Insert New Alias** window (Figure 39) to insert a new alias into an Alias Group, or insert a new alias on its own directly under the Alias Configuration node.

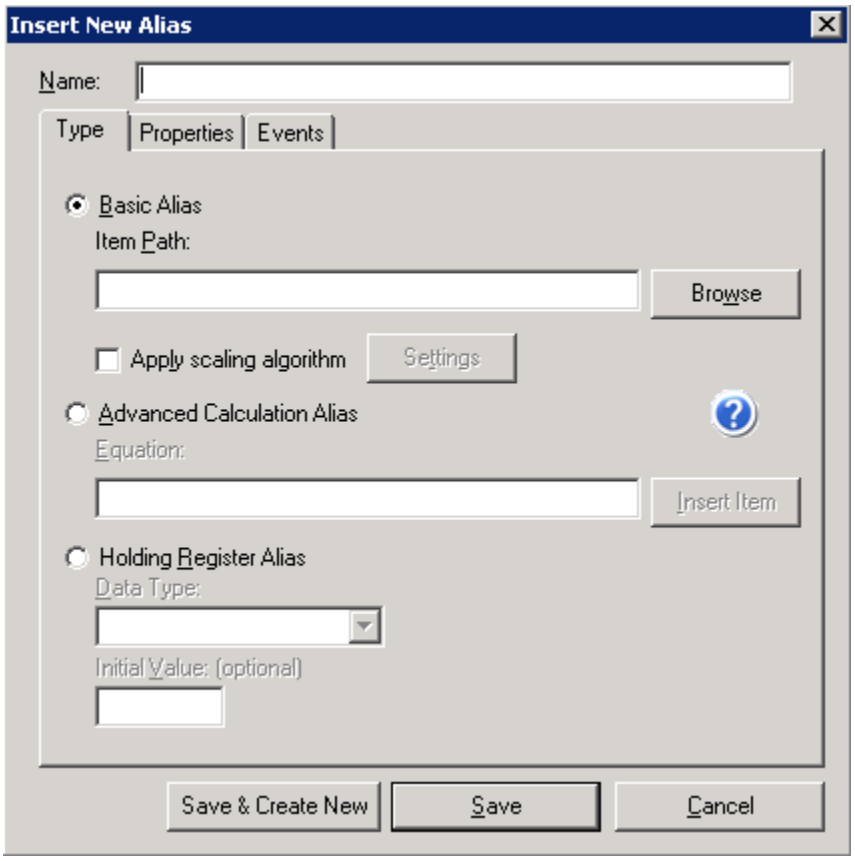


Figure 39 - Insert New Alias Window – Alias Type

Table 32 describes the components of the **Insert New Alias** window.

Component	Description
Name	Allows you to enter a name for the alias (which must be unique within the alias group).
Basic Alias	A basic alias is an alias that renames a data item in the server and optionally allows the configuration of simple scaling. This Alias type is typically used when trying to simplify a server's address space.
Item Path	Fully-qualified item ID for the OPC item to which the alias refers. Click on the **Browse** button to display the **Item Path Browser** window used to select an existing OPC item on the server. **Note:** If ***Events*** are enabled, the alias creates a data subscription to the configured item at the **Alias Subscription Rate**. This rate is configured in the **Advanced Options** window.
Apply Scaling Algorithm	Basic Aliases can be configured to have scaling calculations applied to the value of the mapped item ID configured under **Item Path.** To enable the use of a scaling algorithm check the **Apply scaling algorithm** checkbox. Click on the **Settings** button to display the **Configure Scaling** window used to configured the scaling algorithm used.
Advanced Calculation Alias	An Advanced Calculation Alias allows the user to configuration a data point whose value is the result of a mathematical equation that can

Component	Description
	involve values from multiple data items in the server. See **Appendix B – Aliases – Advanced Calculations** or click on the Help icon for information on how to write equations and the functions available.
Equation	The equation to apply to calculate the data value of the alias. Use the **Insert Item** button to display the **Item Path Browser** window used to select an existing OPC item on the server and insert a reference into the equation.
Holding Register Alias	A Holding Register Alias is a data item that is not associated with any data items in the server. These registers can be read from and written to like any other data item. Typically Holding Register Aliases are used as temporary data locations or for configuration testing during system deployment.
Data Type	The canonical data type of the holding register. All values written to this alias will be converted to this data type. By default, clients will be sent data as this data type.
Initial Value	By default, a holding register will initially contain a value of 0 or the closest value to zero that applies to the data type. If this value requires a different initial value it can be entered here. **Note:** Not all data types support the configuration of an initial value. This field may be disabled depending on the data type chosen for the holding register.
Save	Save changes to the alias and close the window.
Save & Create New	Save changes to the alias and begin configuring a new one.
Cancel	Discards changes to (abandon creation of) the alias and closes the window.

Table 32 - Insert New Alias Window – Alias Type Options

Component	Description
Data Type	Allows you to select, from the drop-down list, the value of the OPC item to the specified type unless ***Default*** is selected. Also serves as the "canonical" data type for the alias. Default = ***(Default)***.
Update Rate	Allows you to enter a value defining the update rate (in milliseconds) to use with the **Poll when inactive** option. Default = ***0***.
Read only	When this checkbox is selected, it prevents OPC clients from writing values to the alias. **Note:** This option is not available for ***Calculation*** aliases, as they are read-only. By default, this checkbox is cleared.
Poll when inactive	If selected, continues updating the value for the alias at the specified update rate even if there are no OPC clients currently accessing the alias.

Table 33 - Insert New Alias Window – Alias Properties Options

To insert a new alias:

Note: It is recommended that you create a new alias under an **Alias Group** rather than directly under the **Alias Configuration** node.

1. On the **Configuration** window, select **Alias Configuration** or a previously configured **Alias Group**.
2. The **Contents** table for that **Alias Group** is displayed on the right side of the **Configuration** window (Figure 40), listing the aliases it contains.

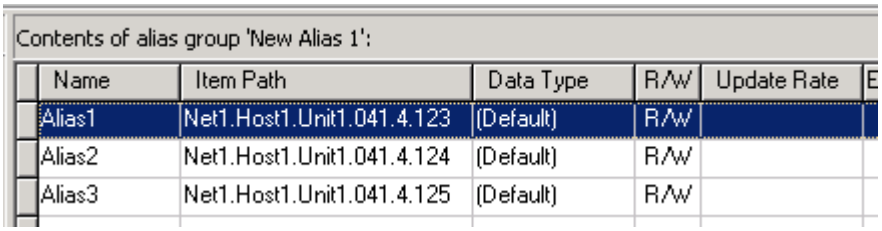


Figure 40 - Contents Table

3. From the **Edit** menu, choose **Insert New Alias**.
4. The **Insert New Alias** window (Figure 39) appears.
5. Enter a name for the alias in the **Name** field.
6. Select **Basic Alias.**
7. Enter the item **Path** and **ID** for the item to which the alias refers, or click on the **Browse** button to navigate to the item.
8. If scaling is desired, check the **Apply scaling algorithm** checkbox and click on the **Settings** button to **Configure Scaling**.
9. From the **Insert New Alias** window, click on the **Save** button to save the alias.

Editing Aliases

Note: Aliases can be edited after they have been created using the **Edit Alias** window. This is virtually the same as the **Insert New Alias** window (Figure 39) with the exception that it appears after the alias has been created.

To edit an alias:

1. On the **Configuration** window, select the alias you want to change, and then double-click or press **Enter**.
2. The **Edit Alias** window appears.
3. Make the changes as required.

Removing Aliases

Note: Any alias that was created can be removed.

To remove an alias:

1. On the **Configuration Window**, select one or more aliases that you want to remove.
2. From the **Edit** menu, choose the **Delete Alias** menu option.
3. The alias is removed.

Removing Alias Groups

Note: Any Alias Group that was created can be removed.

To remove an Alias Group:

1. On the **Configuration** window, select the Alias Group you want to remove.
2. From the **Edit** menu, select the **Delete Alias Group** menu option or click on the  button.
3. The **Alias Group** is removed.

Exporting Aliases

Note: Use the **Export Aliases** window to export all currently configured aliases to a CSV file.

To export all currently configured aliases:

1. From the **File** menu, select the **Export Aliases** menu option or click on the  button.
2. The **Export Aliases** window appears.
3. Enter a file name for the CSV export file.
4. Click **Save**.
5. The aliases are exported.

Importing Aliases

Note: Use the **Import Aliases** window to clear all aliases and import new ones from a CSV file. The CSV file format does not support event generation settings. All imported aliases will be configured to **not** generate events.

To import aliases from a CSV file:

1. From the **File** menu, select the **Import Aliases** menu option or click on the  button.
2. The **Import Aliases** window appears.
3. Browse to the CSV file name.
4. Click **Open**.
5. The old aliases are cleared and the new ones are imported.

Configure Alias Scaling

An alias can be configured to apply a scaling algorithm to the data. Below is a description of how to configure the scaling algorithms.

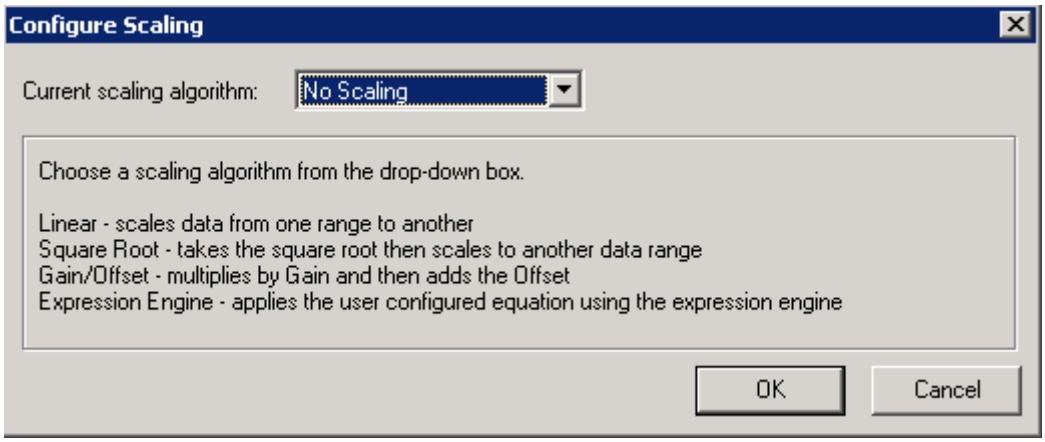


Figure 41 - Configure Scaling Window

Table 34 describes the **Scaling Options** for the **Configure Scaling** window.

Selection	Component	Description
Linear and Square Root	**Raw High**	Maximum expected raw value of the OPC item.
	Raw Low	Minimum expected raw value of the OPC item.
	Scaled High	Maximum desired scaled value for the alias (corresponds to **Raw High**).
	Scaled Low	Minimum desired scaled value for the alias (corresponds to **Raw Low**).
	Clamp High	Clamp the scaled value at the high limit to prevent it from going out of range.
	Clamp Low	Clamp the scaled value at the low limit to prevent it from going out of range.
Gain/Offset	**Gain**	Scaling factor. (No limits are assumed.)
	Offset	Scaling offset.
Expression	**Input**	Simple text equation to apply to incoming values (reads). Click on the ellipsis button (…) to display the **Expression Wizard** screen used to help build a formula. For more information, refer to the **Expression Wizard** section in this manual.
	Output	Simple text equation to apply to outgoing values (writes). Click on the ellipsis button (…) to display the **Expression Wizard** screen used to help build a formula. For more information, refer to the **Expression Wizard** section in this manual.

Table 34 - Configure Scaling Window Options

Notes:

- The data type for scaled values is double-precision real unless the alias data type is specified.
- Using invalid settings, such as ***zero-zero*** for the high-low range in **Linear** scaling, may be fatal to the server, OPC clients, and communications.
- The OPC item values must be numeric for **Linear**, **Square Root**, and **Gain/Offset** scaling to work.
- The OPC item values must be non-negative for **Square Root** scaling to work. Otherwise, the scaled value will be ***-1*** with a bad quality.
- The server may be unable to load configuration files containing invalid scaling expressions, and may fail to start up properly. If this occurs, rename the configuration file so that the server will not load it upon start-up. Alternatively, edit the XML file to correct the invalid alias settings.

Configure Alias Events

Note: Alias Events needs to be licensed separately, and by default, is delivered with only a 30-day demo. To purchase a full Alias Events license, please contact your MatrikonOPC Account Manager.

Use the **Generate Events** (Figure 42) window to configure whether or not aliases generate OPC A&E events when their value changes. Access the **Generate Events** tab through the **Insert New Aliases** window (Figure 42).

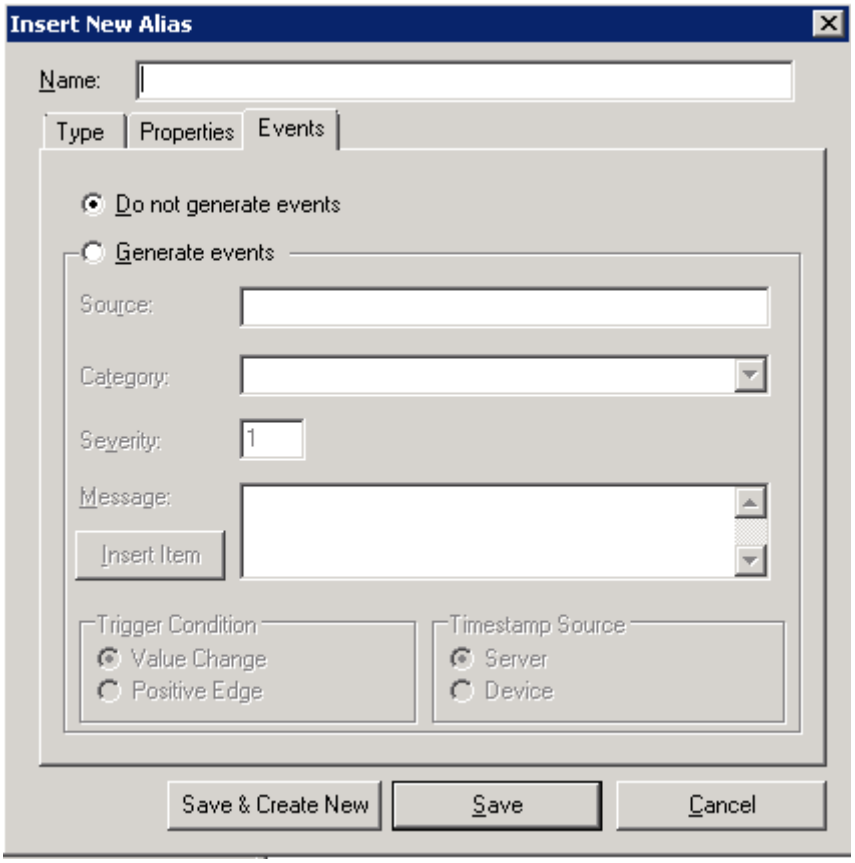


Figure 42 - Generate Events Options

Table 35 describes the **Events** tab fields.

Component	Description
Source	Allows you to enter the name of the source of the event. By default, this field is the Alias Group name, or if there is no Alias Group then the Alias name.
Category	Allows you to select from the drop-down list, the event category under which the event is to be registered. Select either the ***New Category*** or ***Edit Category*** option to configure the registered categories in the server. For more information, refer to the **Configure Categories** section for more information.
Severity	Allows you to enter a value defining the severity of the alarm. The allowable range is ***1*** through ***1000***.
Message	Allows you to enter a message that is to appear in the posted event. This message can contain references to data item in the server. These references are replaced with the values of those items at the time the event was generated. Click the **Insert Item** button to display the **Item Path Browser** screen used to add item paths to this field. For more information, refer to the **Item Path Browser** section in this manual. **Note:** If items are referenced, the alias creates a data subscription to the referenced items at the **Alias Subscription Rate**. This rate is

Component	Description
	configured in the **Advanced Options** window.

Table 35 - Generate Events Window Options

Configure Categories

The **New Category** window (Figure 43) is presented when you choose to add a new category to the server from the **Generate Events** window (Figure 42).

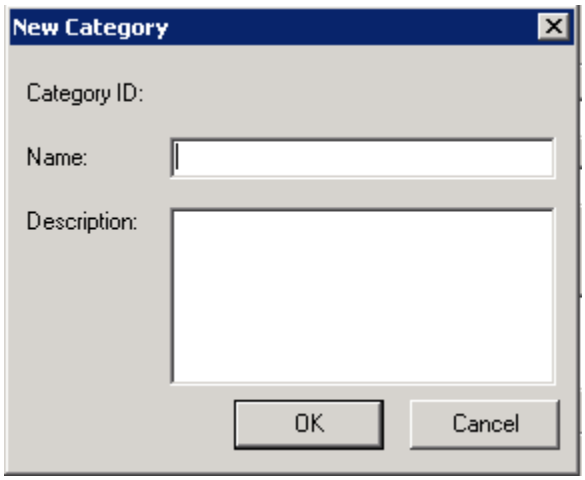


Figure 43 - New Category Window

Table 36 describes the **New Category** window fields.

Component	Description
Category ID	Displays the ID number that appears in the event that is registered under this category. This numbered field is blank until the category is created. Once the category is created the server assigns a unique ID.
Name	Allows you to enter the name that is to be used to refer to the category.
Description	Allows you to enter a description of what the category represents in the system. The name and the description appear together as the category description in OPC clients that support retrieving category information.

Table 36 - New Category Window Options

The **Edit Categories** window is displayed when you choose to edit categories from the **Generate Events** window (Figure 42).

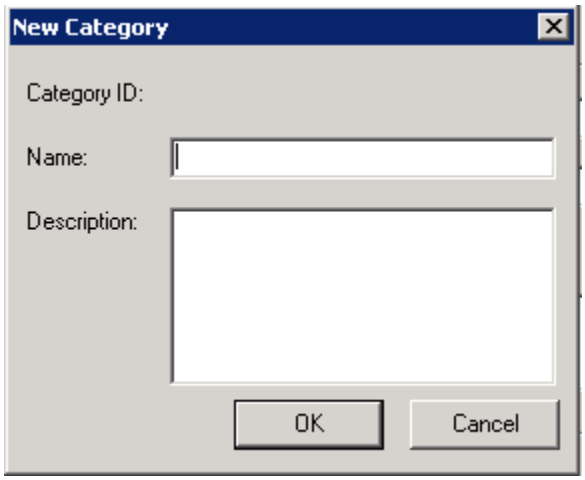


Figure 44 - Edit Categories Window

When presented with the **Edit Categories** window, select the category to edit and click the **Edit** button. This causes the **Edit Category** window to appear which is virtually identical to the **New Category** window.

Note: Some categories cannot be edited. These categories are marked as *Read Only* and can be viewed, but not edited. From the **Edit Categories** window, if you wish to view existing category details, select the required category. Click on the **View** button. The **View Category** window is displayed, which again, is virtually identical to the **New Category** window.

Expression Wizard

The **Expression** form of scaling uses an **Expression Wizard** to build formulas. The created formula is a simple text equation that is applied to either incoming values (i.e., reads), or outgoing values (i.e., writes).

Examples

Example 1:

The following turns any number into a Boolean value. Any non-zero value shows up as a ***1***.

IF(INPUT = 0,0,1)

OUTPUT

Example 2:

The following takes the text out of ***SAMPLE*** and converts it to ***ON***. All other values show as ***OFF***. If ***OUTPUT*** is specified in the **Output** field, then the value is transferred to the end OPC item. Using this alias, you can write ***SAMPLE*** or any other value to the alias to get the display to changed from ***OFF*** to ***ON***.

String comparisons are case-sensitive.

IF(INPUT = 'SAMPLE','ON','OFF')

OUTPUT

Table 37 describes the **Expression Wizard** fields.

Component	Description
Input/Output Expression	Displays the expression as it is created. This field is free-form which allows you to manually enter or edit information. This field reads **Input Expression** when creating an input or "read" expression. It reads **Output Expression** when creating an output or "write" expression.
Database field	This button is reserved for future enhancements.
Function	Select this button to display the **Select function** window which allows you to select the necessary function from those available.
Variable	Select this button to display the **Select variable** window which allows you to select the necessary variable from those available.
Operators	Select the applicable button to add the corresponding operator or operators to the expression.
Clear	Select this button to clear the **Input/Output Expression** field.
Validate	Select this button to validate the expression that appears in the **Input/Output Expression** field. If the expression is invalid, an error message is displayed. If the expression is valid, a message is displayed indicating there are no errors and also stating what the result is.
OK	Select this button to accept changes to (or the creation of) the expression.
Cancel	Select this button to discard changes to (or abandon the creation of) the expression.

Table 37 - Expression Wizard Options

Notes:

- The **Expression Wizard** appears and functions the same for an input expression as it does for an output expression. The differences being the expression field label: **Input Expression**, **Output Expression**, and that one is for reads and the other for writes.
- If you are expecting to read a particular alias, create an **Input Expression**. If you are expecting to write to a particular alias, create an **Output Expression**.

To create an Input or Output Expression:

1. On the **Configuration** window, select **Alias Configuration** or a previously configured **Alias Group**.
2. The **Contents** table for that alias group is displayed on the right side of the **Configuration** window, listing the aliases it contains.
3. Either create a new alias (for instructions, refer to **Inserting New Aliases**), or

   Double-click your mouse on an existing alias, or

   Click on the **Edit the selected alias item** ( ) button.
4. The **Edit Alias** window appears.
5. Check the **Apply scaling algorithm** checkbox and then click the **Settings** button.
6. Select the **Expression Engine** option (Figure 45).

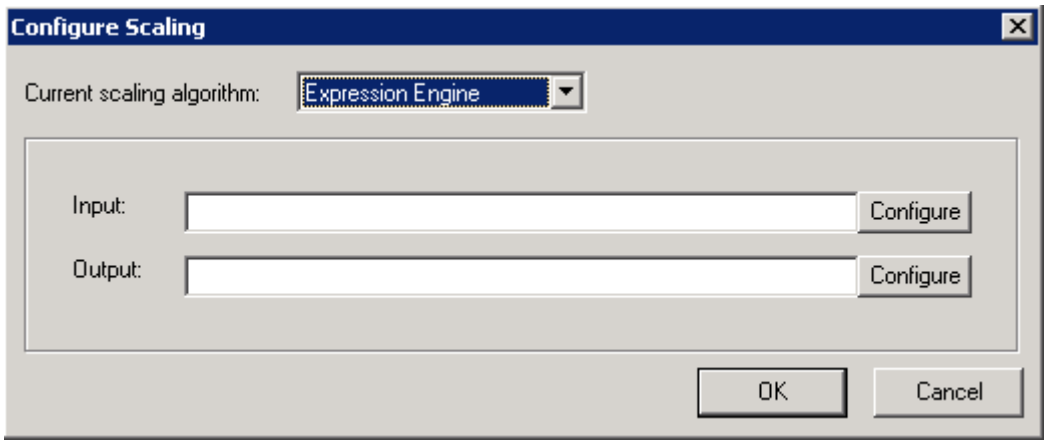


Figure 45 - Expression Engine Configuration

7. Click on the **Configure** button adjacent to the **Input** field (or **Output** field).
8. The **Expression Wizard** (Figure 46) is displayed.

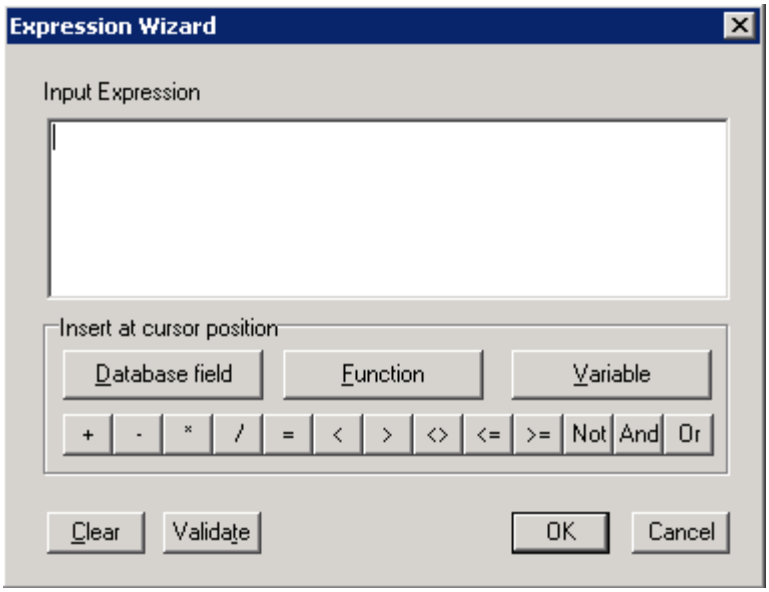


Figure 46 - Expression Wizard

9. Manually enter your new expression in the **Input/Output Expression** field and proceed to step 14.

   Or,

   Perform steps 9 through 13 using a combination of the **Function**, **Variable**, and **Operators** buttons to create your expression, which will be displayed in the **Input/Output Expression** field as you build it.

10. Click on the **Function** button to go to the **Select function** (Figure 47) window. After selecting the required function, click on the **Continue** button to close the **Select function** window and return to the **Expression Wizard**.

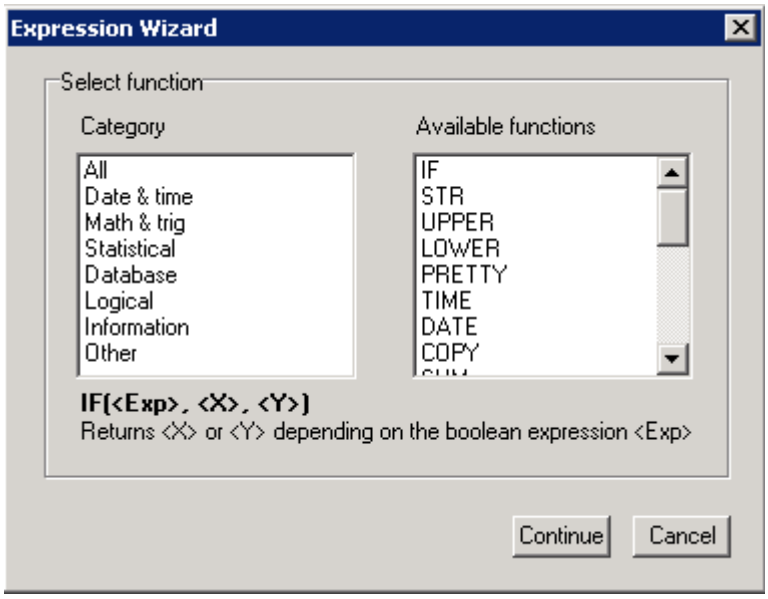


Figure 47 - Select Function Window

11. Click on the **Variable** button to go to the **Select variable** (Figure 48) window. After selecting the required variable, click on the **OK** button to close the **Select variable** window and return to the **Expression Wizard**.

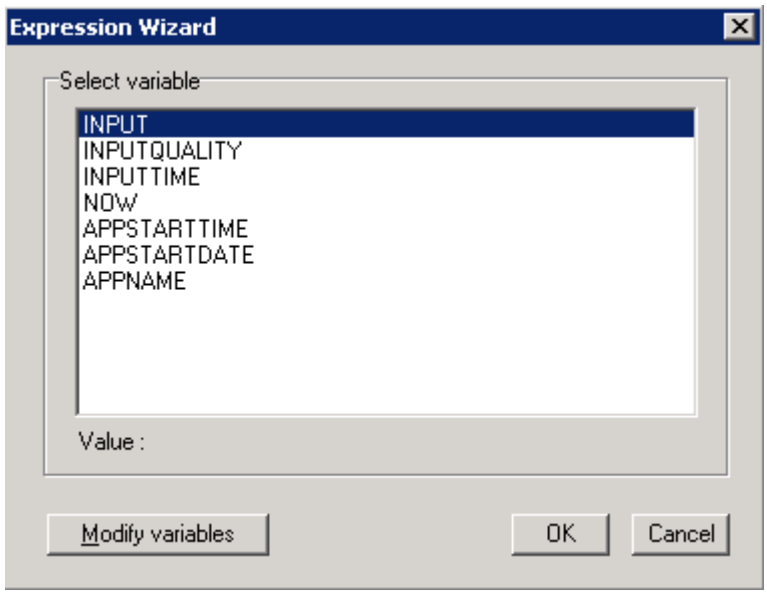


Figure 48 - Select Variable Window

12. If required, from the **Select variable** window, click on the **Modify variables** button to access the **User Defined Functions Editor** (Figure 49) which allows you to add, change, or delete variables.

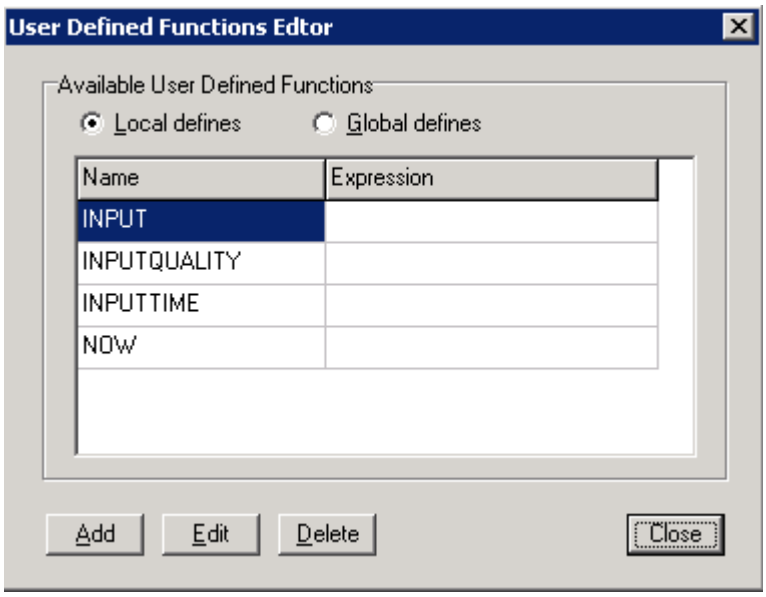


Figure 49 - User Defined Functions Editor

13. The **Operators** buttons (Figure 50) allow you to enter an operation descriptor based on the selected button.

Figure 50 - Operators Buttons

14. Click on the **Validate** button to determine whether or not the expression is valid.
15. If the expression is valid (i.e., a confirmation message is displayed), click on the **OK** button to close the message window and return to the **Expression Wizard** screen.
16. Click on **OK** button on the **Expression Wizard** screen to close the wizard and display the expression in either the **Input** or **Output** field, depending on which expression you defined.
17. From the **Edit Alias** window, click on the **Save** button.
18. The **Edit Alias** window closes and you are returned to the **Configuration** window.

Item Path Browser

The **Basic Alias** and **Advanced Calculation Alias** configuration uses the **Item Path Browser** to add item paths.

Table 38 describes the **Item Path Browser** fields.

Component	Description
Branch Hierarchy	Displays the branch hierarchy of the OPC server when browsing actions are being performed. The **+/-** symbols can be used to display/hide the branches below the specified branch. Left-clicking a branch displays its leaves in the **Leaf List** component.
Leaf List	Displays the list of leaves belonging to a selected branch from the **Branch Hierarchy**. Left-clicking a leaf enables the **OK** button.
OK	Select this button to add the selected item from the **Leaf List** to the **Input/Output** component of the **Insert New Alias** form.

Component	Description
Cancel	Select this button to exit the **Item Path Browser** without adding any item paths.
Refresh	Select this button to refresh the browsing hierarchy in the **Branch Hierarchy** component.

Table 38 – Item Path Browser Options

Saving a Configuration

Note: When the server configuration is complete, use the **Save Configuration** window to save the configuration to an XML file.

To save a configuration to an XML file:

1. From the **File** menu, select the **Save As** menu option or click on the button (alternatively, users can choose **Save** or click).
2. The **Save Configuration** window appears.
3. Enter a file name if saving the file for the first time.
4. Click **Save**.
5. The configuration is saved.

Clearing a Configuration

Note: Starting a new configuration will clear the existing one.

To clear a configuration:

1. From the **File** menu, choose the **New** menu option or click on the button.
2. If OPC clients are still connected to the server, a warning message appears.
3. Click **OK**.
4. The configuration is cleared.

Loading an Existing Configuration

Note: If you have previously saved a configuration, you can clear the current configuration and then load the saved configuration file.

To load an existing configuration file:

1. From the **File** menu, select the **Open** menu option or click on the button.
2. The **Open Configuration** window appears.
3. Browse for the configuration file that you want to load.
4. Click **Open**.
5. If OPC clients are still connected to the server, a warning message appears.

   Notes:

   - At this stage, the connection between the server and the client needs to be broken.
   - When the new configuration is completely loaded, the changes will take effect in the OPC client.

6. Click **OK**.
7. The old configuration is cleared and the new configuration is loaded from the file.

Shutting Down the Server

Note: Ordinarily, the server shuts down automatically shortly after the last OPC client disconnects from it. However, if the **Configuration** window is displayed, the configuration utility locks the server so that it will not shut down.

To shut down the server:

1. On either the **Control Panel** or the **Tool Tray Menu**, select **Shutdown**. If the server is running as a local executable, choose **Shutdown Server** from the **File** menu or click on the button.
2. If OPC clients are still connected to the server, a warning message appears.
3. Click **OK**.
4. The server is shut down.

Note: The shutdown command is not available for the server if it runs as a Windows service. Services can be shut down only from the **Service Control Manager** applet in the Windows **Control Panel**.

OPC Data Items

This section describes the OPC data items used in the MatrikonOPC Server for SCADA Modbus.

SCADA Modbus Items

The OPC item ID syntax of this OPC server is as follows with field descriptions defined in Table 39:

{PORT.}Device.X:YYYYY{DU{P}|F{P}|U{P}|A{P}|{D}{P|/{X}ZZ}}{R\S\T}

Or

{PORT.}Device.XYYYYY{DU{P}|F{P}|U{P}|A{P}|{D}{P|/{X}ZZ}}{R\S\T}

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>I/O:</td><td>The following fields are for computer-to-device I/O:</td></tr>
<tr><td>PORT</td><td>Name of the SA85 Card, Serial Port, or UDP Port under which the Modbus device appears. Not required for Modbus Ethernet or MTL 8000 Serial devices.</td></tr>
<tr><td>DEVICE</td><td>Name of the Modbus device. For Modbus Ethernet this may be a path to the device such as MyNetwork.ThisMachine.ModbusSim.</td></tr>
<tr><td>LOCATION:</td><td>The following fields describe the location of the data point in the device:</td></tr>
<tr><td>X</td><td>Modbus register file number. Not all devices support all file types.
<table>
<tr><th>File Type</th><th>Description</th><th>Size</th><th>R</th><th>W</th></tr>
<tr><td>0</td><td>Digital Output Coil</td><td>1 bit</td><td>✓</td><td>✓</td></tr>
<tr><td>1</td><td>Digital Input Coil</td><td>1 bit</td><td>✓</td><td></td></tr>
<tr><td>3</td><td>Analog Input Register</td><td>16 bits</td><td>✓</td><td></td></tr>
<tr><td>4</td><td>Analog Output Register</td><td>16 bits</td><td>✓</td><td>✓</td></tr>
<tr><td>3W</td><td>Non-Standard Analog Input Register**</td><td>32 bits</td><td>✓</td><td></td></tr>
<tr><td>4W</td><td>Non-Standard Analog Output Register**</td><td>32 bits</td><td>✓</td><td>✓</td></tr>
<tr><td>7</td><td>Analog Input Register**</td><td>32 bits</td><td>✓</td><td></td></tr>
<tr><td>8</td><td>Analog Output Register**</td><td>32 bits</td><td>✓</td><td>✓</td></tr>
</table></td></tr>
<tr><td>YYYYY</td><td>Modbus register address. Possible range is from 1 to 65535. The actual range depends on the Modbus device configuration for the particular register file type.</td></tr>
<tr><td>DATA TYPE</td><td>The following options control how the data is correlated to a computer data type. Only one data type may be selected, except for U, which may only be paired with D.</td></tr>
<tr><td>D</td><td>Append a D to the end of an item ID to indicate that the 3, 4, 3W, or 4W register pairs up with the next register to form a 32-bit integer. For example, 4:00001D reads the least significant word from 4:00001 and the most significant word from 4:00002.
Note: The D option is legal for register types 3W and 4W, but does nothing as this register type is already 32-bits wide.</td></tr>
<tr><td>F</td><td>Append an F to the end of an item ID to indicate that the 32-bit segment starting</td></tr>
</table>

Field	Description
	with this register forms a 32-bit IEEE floating-point real. If the register is of type 3 or 4, then it will be paired up with the subsequent register. If the register is of type 3W or 4W, it will be treated as an IEEE value.
P	Append a ***P*** to the end of an item ID to indicate that the 64-bit segment started with this register forms a 64-bit floating-point real. If the register is of type 3 or 4, then it will be paired up with the subsequent 3 registers. If the register is of type 3W or 4W, it will be treated as an IEEE value.
A	Append an ***A*** to the end of an item ID to indicate that the item is to be treated as a string.
U	Append a ***U*** to the end of an item ID to indicate that the 3, 4, 3W, or 4W register should be treated as an unsigned rather than a signed integer. **Note: *U*** can only be combined with data type ***D***, or with no data type modifier.
MISC	**The following options control how the data is presented to the OPC client and/or device.**
{X}ZZ	Append a forward slash followed by an integer to the end of an item ID to access an individual bit in a 3X or 4X register. The valid range for the bit index is ***0*** to ***15***, or ***0*** to ***31*** if the ***D*** modifier is specified. When ***X*** is pre-pended, it uses function code 16Hex (22) to write. **Note:** Writing individual bits without function code 22 will cause the entire register to be written to, based on the value received during the last read from the device. It is advised that if the read rate of the item is slow, that a device read be performed before writing the individual bit. Performing this read operation will help to prevent old data values being written to the device if the data on the device has changed. For more information, refer to the **Limitations** section in this manual.
S	Append an ***S*** to the end of an item ID to swap the word order from the default (least significant word in lowest address) to reverse format (most significant word in lowest address).
R	Append an ***R*** to the end of an item ID, to swap the byte order from the default (least significant bytes in lowest address) to reverse format (most significant byte in lowest address).
T	Append an ***T*** to the end of an item ID, to swap the byte order inside each word from the default (least significant bytes in lowest address of the word) to reverse format (most significant byte in lowest address of the word).
G	Append ***G*** to the end of any item ID to indicate that you want to write to a 4X register using Function number 21. ***G*** must be the last option in the item syntax When ***G*** is added to an Item Syntax, All of the write requests for that item will include the function code "21" in the sent message

Table 39 - Item ID Field Descriptions

Notes:

- The syntax does not support items with combinations of ***D***, ***F***, and ***U*** modifiers. Although the **Tag Studio** allows these items to be added, the last letter in the combination is all that will be interpreted when reading values. For example, ***Port.Device.X:YYYYYDU*** will act as an unsigned 16-bit value as the ***D*** is ignored.

- Registers 3W and 4W are provided in case your unit does not conform to standard Modbus design. They should only be used if your Modbus unit uses the non-standard larger 32-bit registers. Since 32-bit register design is not part of the standard Modbus protocol, we cannot claim to successfully support this mode for any particular device.
- Registers 7 and 8 are not standard and may not be supported on some devices. The hint tag for these registers will be shown regardless of their availability.
- Writing to items on a failed Modbus unit forces reconnection to the failed unit. Overall performance of the server is affected if the failed unit is continually written to.

The MatrikonOPC Server Framework creates a number of standard data items which can be used for monitoring and controlling certain behaviour.

Table 40 describes the standard items available for most device links.

Field	Description
#Enabled	This is a Boolean OPC tag which indicates whether or not the Device Link to which it belongs is enabled. If the value is ***True*** or a value of ***True*** is written to this item then the Device Link is enabled.
#OfflineMode	Adding this item allows an OPC client to monitor the offline mode for that location using read operations and provide the ability to toggle offline mode for that location using write operations. For more information, refer to **Appendix H - Offline Mode**.
@Connected	This is a Boolean OPC tag which indicates whether or not the Device Link is connected to its resources. Typically ***True*** indicates that a connection of an end device has been established.

Table 40 - Standard Data Items

Table 41 describes the standard items available for at the root branch of the server browse tree.

Field	Description
#MonitorACLFile	This is a Boolean OPC tag that has a default value of ***True*** when the server starts which indicates that the tag security permissions are monitored and applied while the server is running. By adding this item to an OPC group and writing a value of ***False*** to this item, the server no longer applies tag security configuration changes while the server is running. The server would need to be stopped and restarted for the changes to take effect. For more information, refer to **Appendix G - Security**.
@Clients	This is a read-only string item which lists the name of every client currently connected to the OPC server.
@ClientCount	This is a read-only unsigned integer item which provides the count of OPC clients connected to the server. **Note**: This count includes open PSTCFG GUI.

Table 41 - Server Data Items

Table 42 describes the statistical and control items available for the Redundancy Group device link.

Field	Description
#Manual Device Link	Write to this control item to indicate which device link, of the device links to which the Redundancy Group is subscribed, is forced to be active, whether or not it is available. Clear the value of this control item in order to allow

Field	Description
	the OPC server to decide which of the subscribed device links should be active, according to their configured priorities and availabilities.
@Active Device Link	The current active device link, which will be the highest-priority of the available device links on the Redundancy Group device link's subscription list.
#Redundancy Group	A numbered list of device links to which the Redundancy Group device link is subscribed. They are ordered from highest to lowest priority. Writing a numbered list of device links will modify the subscribed device links for the Redundancy Group. **Note:** Writing to this item is not supported if the device link name contains colon "**:**" or bracket "**()**" characters. If device link names contain these characters, the Redundancy Group **must** be configured using the **PSTConfig tool**.

Table 42 - Redundancy Device Link OPC Items

Table 43 describes the statistical and control items available for the Server Status List device link.

Field	Description
@Monitor List	A numbered list of the device links being monitored by the Server Status List device link. The full path of the device link is provided, as entered in the Server Status List device link's subscription list.
@Server Status	Indicates the customized status of the server based on the settings for the Server Status List device link. The status options are: Running, Failed, or No Configuration (if the monitor list is empty). **Note:** Status of **@Server Status** tag will be **Running**, even when the server device link is disabled.

Table 43 – Server Status List Device Link OPC Items

Slave Device Syntax

The Modbus Serial Slave Device has the same syntax format as the general syntax format. There are, however, a few important differences:

- Since this is a simulated device, it is controlled entirely by the OPC client or clients. Because of this, **all items are read-write**, even those that are normally read-only, such as register 1 and 3 items, and bit position items.
- Because we cannot control what command or commands are sent from the Modbus master, option ***P*** is not supported and is invalid for a Serial Slave Device Link item.
- As this is a slave device link and not a master, broadcast is not checked for as broadcast reply does not exist in the Modbus protocol.

Notes:

- Writing bit position to the slave device is not supported at this time.
- 32-bit register (***Daniel Modbus***) replies (replies from registers 3W and 4W instead of 3 and 4) are not supported at this time.

Examples

The Sample Item IDs shown in Table 44 are examples.

Sample Item ID	Description
Card1.Device1.0:56789	DO coil 56789 on Modbus + device ***Device1*** (SA85 card ***Card1***).
Card1.Device2.1:43210	DI coil 43210 on Modbus + device ***Device2*** (SA85 card ***Card1***).
Card1.Device3.40001	AO register 1 on Modbus + device ***Device3*** (SA85 card ***Card1***).
Card1.Device3.40002G	Any write command sent to the AO register 2 on Modbus + device ***Device3*** (SA85 card ***Card1***) will have the function code ***21*** included.
Com1.Rtu1.3:1	AI register 1 on serial device ***Rtu1*** (port ***Com1***).
Com1.Rtu2.4:2	AO register 2 on serial device ***Rtu2*** (port ***Com1***).
Udp1.Device1.3:12345D	AI register 12345 on UDP device ***Device1*** (port ***Udp1***) as a 32-bit integer (paired with register 12346).
Udp1.Device2.4:06789F	AO register 6789 on UDP device ***Device2*** (port ***Udp1***) as a 32-bit IEEE floating-point real (paired with register 6790).
Mtl8000-1.3:3U	AI register 3 on device ***Mtl8000-1*** as an unsigned 16-bit integer.
Mtl8000-1.4:10DS	AO register 10 on device ***MTL8000-1*** as a 32-bit integer with the words swapped (register 11 contains the least significant word).
Ethernet1.3:7FS	AI register 7 on device ***Ethernet1*** as a 32-bit IEEE floating-point real with the words swapped (register 8 contains the least significant word).
Ethernet1.4:9/0	Bit 0 of AO register 9 (least significant).
Ethernet1.3:16D/16	Bit 16 of AI register 16 as 32-bit integer (actually bit 0 of register 17).
Ethernet1.4:20DS/31	Bit 32 of AO register 20 as 32-bit integer (actually bit 15 of register 20 since the words are reversed).
Ethernet1.4:20/X5	Bit 5 of AO register 20 (when writing function code, 16Hex will be used).

Table 44 - Sample Item ID Descriptions

SCADA Modbus Access Paths

This OPC server does support access paths to further define how the OPC item should communicate with the underlying system. If your OPC client does not support access paths, you can add them to the end of your item ID by specifying the **@** sign after your item ID. For example:

TCP.Host.Unit.4:1@10000

Access Path Descriptions

Table 45 shows the available access paths.

Field	Description
+	When the item is added, an immediate poll is triggered for that item. This allows the

Field	Description
	OPC server to reply with data immediately instead of waiting for the update rate on the first scan.
!	Indicates that the server must poll the item at a high priority (honouring the fastest scan rate). If the same item is requested several times, then the server will poll the item at high priority if any of the requested access paths contain this flag.
NNNN	This number specifies the update rate that should be used to the end OPC item. This update will be maintained no matter what update rate is requested from an OPC client.

Table 45 - Access Path Descriptions

Getting Data without Being Connected to a Device

The MatrikonOPC Server Framework supports "offline mode", which allows OPC clients to obtain changing good quality data without being connected to a device. This is useful if you wish to test an HMI or other OPC client without worrying about the connection to the device, or the quality of the data provided. For more information, refer to **Appendix H - Offline Mode**.

SCADA Communication

In general, most SCADA applications involve communication with large fields of remote units over a slow medium, such as radio or dial-up modem. As a consequence, it may take a relatively long time to collect/exchange all of the necessary data from/with the field. Such applications require the use of an efficient and orderly transaction exchange mechanism in order to transfer data reliably.

This section describes how the MatrikonOPC Server for SCADA Modbus implements reliable, optimized communication for supervisory control and data acquisition.

Channels

The MatrikonOPC Server for SCADA Modbus supports the channels of communication described in the following sub-sections.

Serial

The MatrikonOPC Server for SCADA Modbus supports direct serial communication via RS232, RS422, RS485, etc. In addition, the server:

- Supports a variety of settings for baud rate, data bits, parity, and stop bits.
- Supports RTS/CTS flow control as well as other common types of flow control (DSR/DTR, software XON/XOFF).
- Supports DCD transmission collision avoidance with random back-off time.
- Allows you to preset control line states (RTS, DTR) to accommodate line converters (RS232/RS485, RS232/RS485, etc.).

Radio

The MatrikonOPC Server for SCADA Modbus supports serial radio modems with the following additional requirement:

- The server supports key up/down delays for transceiver warm-up and cool-down, with different delays for one or more repeaters on the same channel.

Network

The MatrikonOPC Server for SCADA Modbus supports network communication via Internet Protocol (IP). In addition, the server:

- Establishes client connections to both TCP/IP and UDP/IP servers.
- Accepts incoming connections and responds to requests from TCP/IP or UDP/IP clients.
- Supports direct network communication with a single device or with multiple devices through a terminal server.
- Supports Point-to-Point Protocol (PPP) and Cellular Digital Packet Data (CDPD) via dial-up networking (externally pre-configured).

Dial-Up

The MatrikonOPC Server for SCADA Modbus supports Hayes-compatible dial-up modems via serial or network connection. In addition, the server:

- Supports any modem that responds to the standard AT command set.
- Tolerates any character case and white space in command responses.

- Issues reset and initialization commands before dialing out and after hanging up.
- Supports communication directly with Cellular Digital Packet Data (CDPD) modems using the Hayes standard AT command set. These modems must respond normally to the following commands: AT, ATZ, ATO, ATH0, and ATD followed by the user entry. **Note:** The server can be configured to avoid using the ATZ command if necessary.
- Supports auto-answer for dial-in connections even while using the same modem to dial out.

Poll Optimization

The MatrikonOPC Server for SCADA Modbus organizes poll requests as efficiently as possible.

The server provides values for data items in OPC groups at the requested update rate for those groups (fastest rate used when more than one group references the same items). If the requested update rate is too fast for the server to handle, it makes a "best effort" to update as fast as possible. Internally, the server tries to keep the data items at least as fresh as indicated by the update rate. The server will not, however, send updates to the OPC client any faster than the update rate in order to avoid overwhelming it.

Poll optimization involves the destruction and re-creation of communication packets for reading in the most efficient manner. During the optimization process, the server gathers the requested data items into bundles with similar scan rates, and then uses those bundles to generate packets. Optimization is most successful when it generates the fewest number of packets to contain all items with the same scan rate.

Once the server has completed the optimization process and has generated the necessary packets to read the requested data, it then determines whether to queue the packets for transmission. Queued packets are transmitted to the unit at the earliest opportunity. Once all packets have been sent, the queue is emptied and the unit relinquishes control of the circuit to the next unit.

Transactions

The MatrikonOPC Server for SCADA Modbus employs a robust transaction exchange mechanism for supervisory control and data acquisition. The server completes all same-priority transactions for a particular unit before moving on to the next unit, but interrupts lower-priority transactions in order to carry out higher-priority transactions, (after the current transaction has completed). Demand operations, such as writes and device reads, have the highest priority.

Continuous

The MatrikonOPC Server for SCADA Modbus polls all units continuously, in a cyclic, round-robin fashion (one after another) in order of the unit station numbers, when configured to do so. The server optimizes together low-priority items with scan times faster than a configured threshold rate and then polls them continually.

Any packet with a scan rate less than the threshold rate is queued immediately following optimization. Also, any packets with a scan rate less than the threshold rate is immediately re-queued when the queue is emptied.

Threshold Rate

In some cases, it may take minutes or even hours for the server to perform all of the communication operations necessary to finish one complete poll of all units on a channel. This has a significant impact on the optimization and scheduling of read operations. The Threshold Rate is used by the server to distinguish between scan rates that are attainable and those that are not.

The Threshold Rate indicates the fastest scan rate (in milliseconds) that the channel can realistically provide. This allows the polling mechanism to generate larger, more efficient packets for items with scan rates faster than the Threshold Rate, to be sent "as fast as possible". Items with scan rate equal to or slower than the Threshold Rate are optimized as normal and scheduled to send at their specified rate.

For example, if it takes approximately five minutes to collect all the desired information from a large field of units, then it is not possible for a particular item to receive an update every ten seconds without interrupting the round-robin polling sequence. In such a case, a Threshold Rate of 300,000 milliseconds would be appropriate, as it would allow the driver to optimize all items scanning faster than five-minute intervals into fewer packets.

Configuring the Threshold Rate for best performance may require some experimentation. The Poll Time and Overrun statistics may help to determine the best setting. Alternatively, choose a smaller Threshold Rate and configure all fast items to have scan rates below the Threshold Rate and all slow items to have scan rates well above the chosen rate.

Interrogation

Use the Interrogate control item to place a particular unit on a communication channel into interrogate mode. This has the effect of interleaving high-priority polls for that unit into the round-robin sequence. The interleaved polls only request items with scan rates less than the Threshold Rate; scheduled reads continue in the normal fashion (unaffected).

For example, in a field of ten units, if unit 3 is placed into interrogate mode, all of its continuous polls will be sent after unit 1 has finished sending, and again after unit 2 has finished sending, and again after unit 4, and so on. In other words, the polling sequence changes from 1...2...3...4...etc. to 1...3...2...3...4...3...etc.

This option might be used when activity on a specific unit is of particular interest for a period of time, and operators are not able to wait for the complete round-robin sequence to finish. Only one unit on a channel may be placed into interrogate mode at a time. Naturally, the interleaved polls increase the total amount of time required to poll all units on the channel.

Scheduled

The polling mechanism manages the schedule during idle processing. Whenever a packet becomes due, it is placed on a queue for transmission at the earliest opportunity. The packet's next scheduled time is then calculated, usually just by adding the scan interval to the last scheduled time.

When scheduling packets, the MatrikonOPC Server for SCADA Modbus calculates the absolute time at which that packet should be sent first (aside from the initial queuing). This first scheduled time is normalized to the base poll time configured for the unit. The server handles scheduled polls due for a unit first before handling continuous polls.

Scheduled packets with the initial poll option are queued immediately. The server always performs optimization when initially establishing or re-establishing communication with a unit. Therefore, the aforementioned packets are queued immediately at these times. The schedule all polls option transforms all continuous polls on a channel into scheduled polls with a scan rate equal to the Threshold Rate. This allows for finer control over precisely when these polls occur.

Priority

Certain types of operations may need to interrupt the regular round-robin polling sequence in order to carry out important tasks. To facilitate this, the "due" queue is sorted, first by the priority level of the packet, and then by the order in which they are queued. Thus, the packet at the front of the queue is also the highest-priority packet to be sent.

When a unit queries the channel to find out whether it is its turn to communicate, it also indicates the priority level of its next packet to be sent. If this level is higher than the level presently being used by any other unit, the requesting unit is granted priority override and takes over communicating on the channel. The unit will retain priority override as long as it continues sending packets at the same priority level, and no other unit requests to send at a higher level.

Once a unit has finished sending all the packets at a certain level, those packets are immediately removed from the queue, allowing them to become due again before lower-priority packets are finished, if necessary. The channel keeps track of the last unit sending at lower levels so that polling continues where it left off when a unit finishes priority override.

Overruns

Whenever it becomes a particular unit's turn to communicate, it services all of the messages in its own queue until they have all been sent. During this time, duplicate packets cannot be entered into the queue. If a packet becomes due again before it has been removed from the queue, then this constitutes an overrun providing this occurs while communication is still good. Overruns are a symptom that there are communication problems, or that the Threshold Rate needs to be tuned.

Instantaneous

Demand (i.e., user-initiated) operations, such as reading from a device or writing control values, bypass the schedule and priority queue and execute immediately, in the order indicated by the client application. These types of operations force the driver to attempt to re-establish communication with failed devices immediately, regardless of the configured reconnection delay.

Integrity

The MatrikonOPC Server for SCADA Modbus uses specialized communication algorithms to maintain communication integrity as much as possible. This section describes how the server deals with communication integrity with regard to Transmission, Reception, Failure and Recovery, and Redundancy and Fail-over.

Transmission

Before transmitting a packet, the server may impose a short delay, based on configured settings for the port and the unit. This delay can be used to help slower devices keep up with polling, to avoid overloading the circuit, and, sometimes, to allow unsolicited messages to be received from field devices.

Inter-message delay is configured on a per-unit basis, and indicates the amount of quiet time that must be allowed to pass between receiving a response from and transmitting the next request to the same unit. Inter-device delay is configured on a per-port basis, and indicates the delay that must pass between receiving a response from one unit and transmitting a request to another unit on the same circuit. It is also used whenever a broadcast occurs on the circuit.

Reception

The MatrikonOPC Server for SCADA Modbus employs a sliding window algorithm in order to filter out line noise on incoming messages without always having to wait for one entire timeout interval.

The server attempts to read sufficient bytes for a fixed-size response header. If the header is not valid, then the server discards the first byte of the header and repeats the process until the timeout interval passes. If the header is valid, then the server attempts to read sufficient bytes for the remainder of the message (length in header) including the CRC. If the CRC is not valid, then the server discards the first byte of the header and repeats the process until the timeout interval passes. If the CRC is valid, then the server considers the response valid and processes the content.

Failure and Recovery

The MatrikonOPC Server for SCADA Modbus uses the Reply Timeout and Retry settings in order to determine when communication has failed with a particular unit.

Whenever a request sent to a unit fails to receive a valid response within the timeout period, the polling mechanism repeats the request up to the specified number of Retries. If the proper response is still not received, the unit is considered to be in communication failure mode. The server also concludes that communication has failed with a particular unit when the number of consecutive failed demand operations is greater than the Retries setting for that unit.

The server polls units experiencing communication failure at a slower rate, based on the Reconnect Delay, using the configured number of Reconnect Retries for each attempt. However, the server attempts to re-establish communication with failed units immediately when it executes demand operations for that unit.

In addition, the server uses the reconnect delay for the channel when it cannot connect to the communication medium (for example, when it is unable to open the COMM port) to avoid pinning the CPU.

Redundancy and Fail-Over

The MatrikonOPC Server for SCADA Modbus supports redundant channels of communication to the same physical unit. The fail-over mechanism used by the server is flexible, making it easy to use regular fail-over scenarios.

Those operators who have used older versions of SCADA software are likely accustomed to configuring each RTU first and the channels of communication afterwards. With the MatrikonOPC Server for SCADA Modbus, it is the other way around.

You begin by configuring the path of communication from the host computer out to the field, exactly as you would if you were not using fail-over. The only difference is that a given RTU will appear more than once in the configuration, probably once for each communication channel. Each object in the configuration represents a link in the path of communication from the server to a source of real-time data.

For example, you might want to configure the scenario shown in Figure 51.

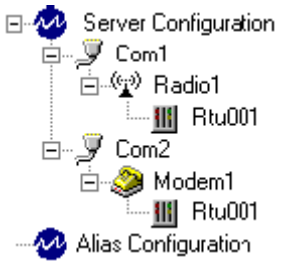


Figure 51 - Server Configuration Scenario

COM1.RADIO1.RTU001 and ***COM1.MODEM1.RTU001*** both reference the same physical unit, but using two different (independent) paths of communication.

If you want to use fail-over with this configuration, you need to add a fail-over object, called ***FOV1***, as shown in Figure 52.

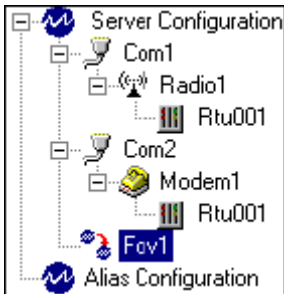


Figure 52 - F0V1 Server Configuration Scenario

In the configuration panel for FOV1 (Figure 53), a list of available communication channels that have already been configured is displayed, in this case ***COM1*** and ***COM2***. For a simple example, you could assign COM1 as a Primary fail-over channel and COM2 as either a Secondary or Standby fail-over channel (these are interchangeable when there is only one non-Primary fail-over channel).

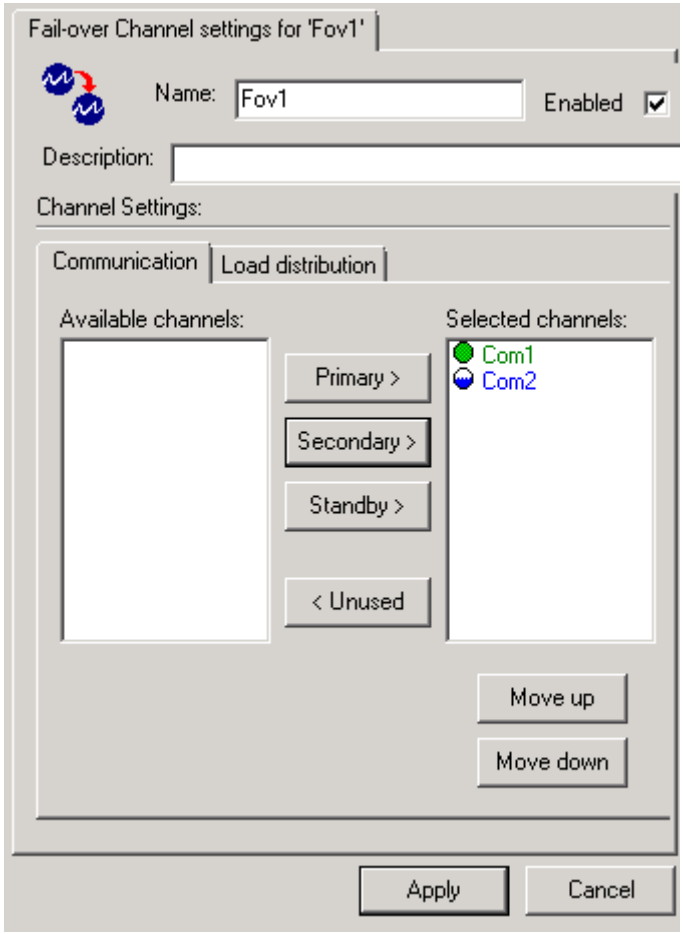


Figure 53 - Fail-Over Channel Settings for F0V1

This tells FOV1 to always continually poll for data using COM1, and only use COM2 if communications fail on COM1. When communications recover on COM1, FOV1 stops using COM2 and returns to using COM1.

Fail-over is employed on an item-by-item basis. If you want the server to fail-over data collection for a particular item from one communication channel to another, you must address that item via the fail-over object rather than one of the configured communication channels. You may still

address the object via the configured communication channels if you have other reasons for doing so, but the server will not use fail-over with that item.

Primary, Secondary, and Standby Channels

In order to address an item via the fail-over object, you must prefix the portion of the Item ID that is common to all communication channels with the name of the fail-over object (***FOV1*** in our example). So, ***FOV1.RTU001.1.M_SP.1***, using the settings in our example, would create and start scanning ***COM1.RADIO1.RTU001.1.M_SP.1*** (Primary) and then create and start scanning ***COM2.MODEM1.RTU001.1.M_SP.1*** (Standby or Secondary) if the Primary ever receives a bad quality (due to communication failure or other problems). The server does not change the qualities of fail-over items to bad until communication has failed with on all channels.

Secondary and Standby are only differentiated when you have more than two communication channels. Secondary is an auxiliary fail-over channel, whereas Standby is only used in an emergency as a last resort. When more than one Secondary channel exists, the channels have an order or preference: when communication is recovered with a higher-priority channel, it ceases communication with lower-priority channels. There is no order for Standby channels: if communication is recovered with just one, it ceases communication with all the others.

The MatrikonOPC Server for SCADA Modbus attempts to communicate with each unit on the Primary channel first (i.e. on start-up) before failing over to a Standby channel. The server fails over communications from the Primary channel to a Standby channel if communications with the unit fail on the Primary channel. The server attempts to re-establish communication with the unit on the Primary channel using the reconnect delay and reconnect retry settings, if configured to do so. Otherwise, the server does not attempt to restore communication with the unit on the Primary channel unless communications fail on the Standby channel (fail-back).

The MatrikonOPC Server for SCADA Modbus supports both "hot" and "cold" Standby channels. The server keeps all "hot" Standby channels polling at all times. The server keeps all "cold" Standby channels disconnected so long as they are not required for failed-over communication. The server determines which Standby channel to use for failed over communications based on the order assigned to the channels by the user. The server determines the Primary communication channel to be the first channel. The server allows the client to disable a specific communication channel in order to force the server to fail communications on that channel.

Another example would be if you add several other paths of communication, such as another modem on COM3, and a satellite uplink on COM4. The modem on COM3 is a slower model, so you would prefer to use the one on COM2 if possible. The uplink may be expensive and you may want to avoid using it except as a last resort. You would configure the fail-over policy as follows:

- COM1: *Primary*
- COM2: *Secondary* (at a **higher** priority than COM3)
- COM3: *Secondary* (at a **lower** priority than COM2)
- COM4: *Standby*

There is nothing to prevent users from adding multiple Primary channels, which would result in a fully redundant communication with the server simultaneously polling the same points on both channels. Sites with very stringent data integrity requirements may choose to do this, but would likely use load distribution instead.

Load Balancing

Load balancing involves configuring Primary - Secondary - Standby on an item-by-item basis using filter strings. The Load Distribution tab contains a string grid that allows you to specify which items are to be polled on the respective communication channels. Rather than specifying individual items,

however, you would enter pattern strings that allow the server to match the items to their appropriate channels. The server uses Visual Basic like syntax for these strings.

The syntax is described in detail at:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vblr7/html/vaoprLike.asp

For example, to have all single-point information objects polled on channel 1 and all analog measurements polled on channel 2, enter ***.M_SP.*** under the channel 1 column and ***.M_ME.*** under the channel 2 column.

The grid grows automatically as you add pattern strings. The pattern string applies to the portion of the Item ID following the fail-over channel name (so, for ***FOV1.RTU001.1.M_SP.1***, ***RTU001.1.M_SP.1*** matches the filter ***.M_SP.***).

You can prefix a pattern-matching string with an exclamation point (!) to indicate items that you would like to avoid scanning on a particular channel, if possible. The server will only poll those items on that channel as a last resort, such as when communications have failed on every other channel. The server would stop polling such items on the non-preferred channel as soon as communication is recovered on a preferred channel.

MatrikonOPC Server for SCADA Modbus Implementation

For information about MatrikonOPC Server for SCADA Modbus, refer to applicable documentation at **http://www.modbus.org**.

The following table details the supported features.

Function Code	Description	Supported?
01	Read Coil Status	YES
02	Read Input Status	YES
03	Read Holding Registers	YES
04	Read Input Registers	YES
05	Force Single Coil	YES
06	Present Single Register	YES
07	Read Exception Status	
11	Fetch Communications Event Counter	
12	Fetch Communications Event Log	
15	Force Multiple Coils	YES
16	Preset Multiple Registers	YES
17	Report Slave ID	
20	Read General Reference	
21	Write General Reference	
22	Mask Write 4X Register	
23	Read/Write 4X Registers	
24	Read FIFO Queue	

Table 46 - Supported Features

Data Types

The following table details what data types are supported, and how they map to the VARIANT types used by OPC.

Table 47 shows data type mappings for OPC and MatrikonOPC Server for SCADA Modbus.

System X Data Type	VARIANT Type	Description
BOOL	VT_BOOL	Boolean
I32	VT_I4	32-bit integer
U32	VT_UI4	32-bit unsigned integer
I8ARRAY	VT_ARRAY|VT_I1	8-bit integer array

System X Data Type	VARIANT Type	Description
DATETIME	VT_DATE	Date/time
STRING	VT_BSTR	Character string NULL-terminated
XML	VT_BSTR	XML data (byte sequence)

Table 47 - Data Types

Available Item Types

Table 48 shows the available item types.

Item ID	Data Type	R	W	Comments
Binary_Readonly	VT_BOOL	✓		Binary Read-only.
Short_ReadWrite	VT_UI4	✓	✓	Short Read/Write item.
Real_ReadWRite	VT_R8	✓	✓	Real Read/Write item.
Date_Item	VT_DATE	✓		Date and time.
Short_Array	VT_ARRAY|VT_UI1	✓		Storage object (three elements).
String_Item	VT_BSTR	✓	✓	String (maximum 255 characters).
Description_Item	VT_BSTR	✓		Description item.

Table 48 - Available Item Types

Diagnostics

The server supplies diagnostic information to assist operators and support personnel with troubleshooting communication problems and software faults.

Logging

All MatrikonOPC servers produce log files that record errors and debugging information. The log files can be extremely valuable for troubleshooting. As such, it is important to note that the default log level is set to low. The **General Logging** tab of the **Options** dialogue in the main configuration window contains settings to control server logging.

General Activity Logging records information about the internal workings of the OPC server. It is useful for troubleshooting problems with configuration and device communication. Interface Activity Logging records information about the client/server OPC communication. It is useful for troubleshooting compliance issues.

For General logging, the higher the log level, the more information that is recorded. However, server performance may decrease at higher log levels. The recommended operating level is ***Medium***. More log levels are available for both types of logging which record more detailed information, as well as the ability to output log statements to a console window at run time. Contact MatrikonOPC support for further instructions on how to enable the higher log levels and console logging.

For Interface logging, the log level can be considered very high and therefore should never be used during normal operation of the OPC server unless specifically working on problems related to interfacing.

MatrikonOPC Sniffer is a useful tool for logging OPC client/server communication. It enhances the Interface Activity Logging by recording the client-side transactions. It can also be used with other OPC servers. MatrikonOPC Sniffer is a utility used to troubleshoot OPC client/server interoperability issues. Contact the MatrikonOPC Sales department for further information about this product, or use the following link to access the **MatrikonOPC Sniffer Download**.

In general, the server logs all errors and other information of immediate importance to the user at the low detail level, all warnings and other information of moderate importance to the user at a medium detail level, and additional information concerning the normal functioning of the software at a high detail level. The server also logs further information of concern to support personnel at the debug log level.

Note: The higher the log level, the slower the performance of the server. It is recommended that the log level be left at the default, unless troubleshooting needs to be performed.

Limitations

MatrikonOPC SCADA Modbus has the following expected limitations:

- **Bit writes** - when performing bit writes without the use of function 22 (see **SCADA Modbus Items** for details) it is possible to overwrite data changes that have occurred in the register containing the bit to which is being written. To perform a bit write, the server will change the desired bit in the last read value from the device register and write that modified value back to the device register. If the device register has changed since the last read then all 32-bits of the register could be modified, not just the specific bit being written.

Workaround: To mitigate this scenario on systems that perform infrequent reads, it is recommended that a forced device read be performed to update the server to the latest value from the device before performing the bit write.

Example:

Without Workaround:

10:00:00.000 AM Server reads 0x0000 from the device.

10:00:01.123 AM Devices value changes to 0xFFF0.

10:00:02.000 AM OPC client attempts to set the least significant bit to 1.

10:00:02.001 AM OPC server write 0x0001 to the device.

With Workaround:

10:00:00.000 AM Server reads 0x0000 from the device.

10:00:01.123 AM Devices value changes to 0xFFF0.

10:00:01.999 AM OPC client performs device read and server reads 0xFFF0 from the device.

10:00:02.000 AM OPC client attempts to set the least significant bit to 1.

10:00:02.001 AM OPC server writes 0xFFF1 to the device.

- In scenarios where bit writes to the 65535 register with 65535/0 to 65535/15, the validation and write, will be successful. However, when we try to write 65535/16 to 65535/31, the validation will pass, but while writing few of the simulators will crash.

Refer to the *MatrikonOPC Server for SCADA Modbus Release Notes* for known issues.

Troubleshooting

This section is intended to assist you by providing **licensing** information and **MatrikonOPC Support** contact information. Also addressed here are some of the most common problems encountered, and questions asked, while using this OPC server. Please check the following **Problems/Solutions** and **Questions/Answers** sections before contacting the MatrikonOPC Support team.

Problems and Solutions

"*aprxdist*" error on install

Problem:	What is this "*aprxdist*" error on install?
Solution:	For our installations to work correctly, the install must be able to access the ***C:\WINDOWS\system32*** directory to add, use, and remove the file called *aprxdist.exe* and add the file *opcenum.exe*. Generally, we will see this error on Windows 2003 SP1 system where this is not possible. To resolve the issue, **download these files**, and place them in ***C:\WINDOWS\system32***, and rerun the install.

PSTCFG error

Problem:	After my OPC server died, I tried to access the **Configuration** screen and now I am experiencing access violations and PSTCFG errors.
Solution:	If your OPC server process stops and you leave the PSTCFG running, when you attempt to access the **Configuration** screen to do anything to your configuration items (add, change, delete, and so on), you will experience access violations. You will need to kill the configuration and restart the server.

Server stops updating client with item values after two hours

Problem:	The server stops updating the client with item values after two hours.
Solution:	The server may be licensed with a hardware key and the hardware key may not be properly detected, or the software license has expired. Alternatively, try running the **MatrikonOPC Licensing** utility from the shortcut menu. Click on the **Check Licenses** button to see the status of all MatrikonOPC product licenses. Check that the correct HASP hardware key is securely fastened to the parallel port. If it is, run the HINSTALL program that is located in the ***Program Files\Common Files\MatrikonOPC\Common*** directory.

Server stops updating client with item values after 30 days

Problem:	The server stops updating the client with item values after 30 days.
Solution:	The evaluation period for the license has expired. Please contact **MatrikonOPC Support**.

Server shuts down after five minutes when no clients connected

Problem:	The server automatically shuts down after five minutes when there are no clients

connected.

Solution: This is standard behaviour for COM servers. However, this behaviour can be overridden as shown in the following example:

Note: Microsoft highly recommends that you back up the registry before editing it or making changes to it.

1. From the **Run** command, enter ***REGEDIT***.
2. The **Registry Editor** (Figure 54) is launched.
3. In the **Registry Editor**, under **HKEY_CLASSES_ROOT**, look for the required program ID (e.g., *Matrikon.OPC.Simulation.1*).

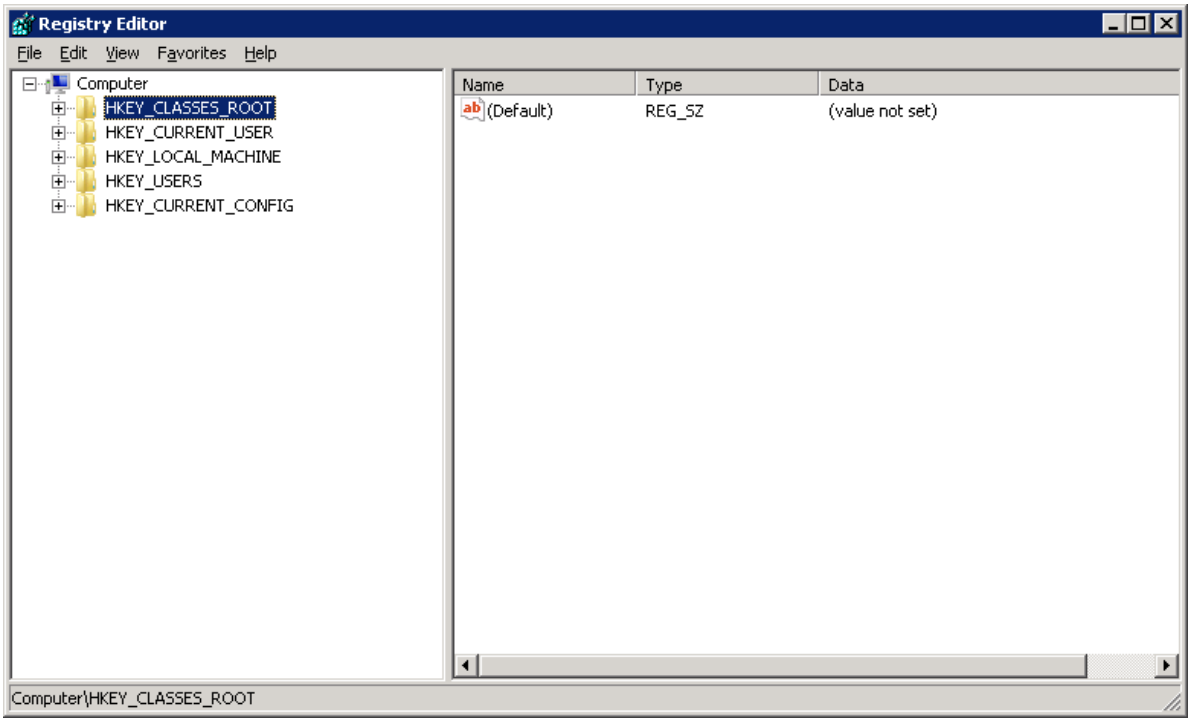


Figure 54 - Registry Editor

4. Expand the program ID branch.
5. Click on the CLSID branch.
6. Record the global universal ID (GUID).

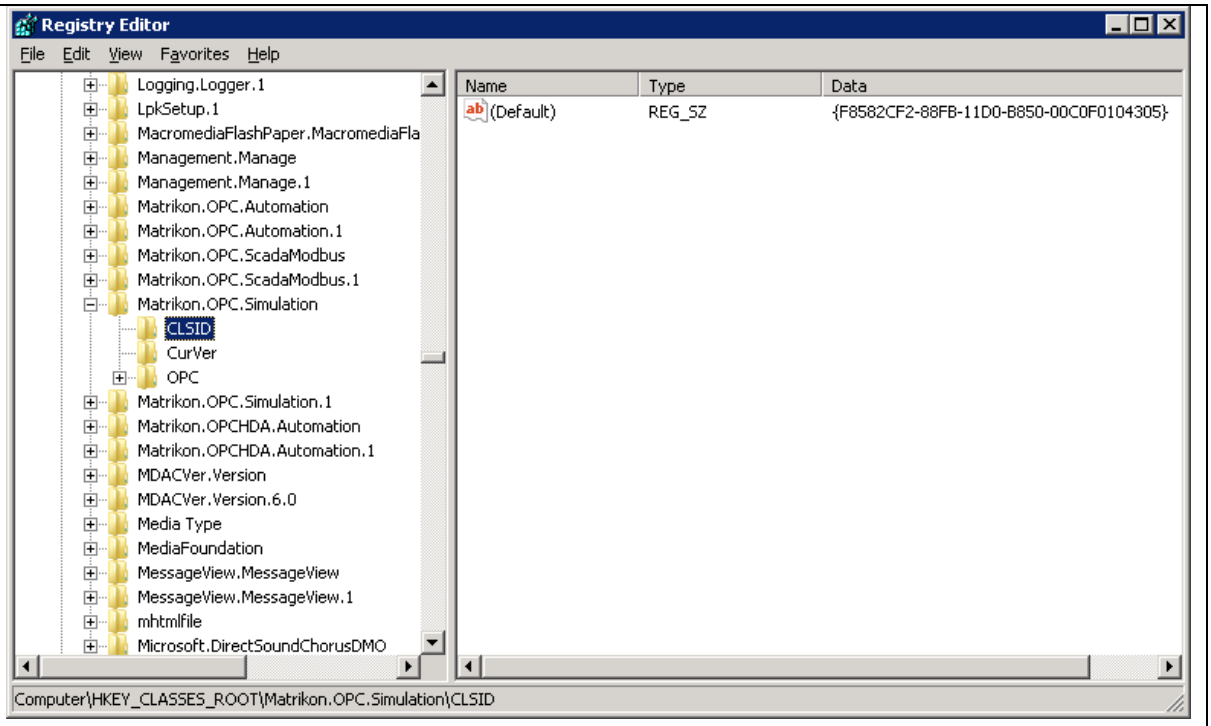


Figure 55 - Program ID Branch

7. Under the **HKEY_CLASSES_ROOT\CLSID** branch, search for the GUID retrieved in the previous step (e.g., ***HKEY_CLASSES_ROOT\CLSID\{024C7FFC-9CC3-476f-867A-BC3CAD79F4FE}***).
8. Create a string value called **LockServer** in the **Options** branch underneath the GUID (e.g., ***HKEY_CLASSES_ROOT\CLSID\{024C7FFC-9CC3-476f-867A-BC3CAD79F4FE}\Options***).
9. Create a new value using the context menu and choose **New String Value** from the menu.
10. Set the value of the **LockServer** entry to ***1***.

    Note: To edit the value, select the entry and select the **Modify** menu option from the **Edit** menu.
11. Shut down the server and restart it for the change to take effect.

Cannot configure client access paths, so cannot configure item options

Problem:	The OPC client does not provide a means to configure access paths, making it difficult to correctly specify the data acquisition options for the items.
Solution:	The server provides a global parameter for configuring an escape character, to allow users to append communication options to the end of an item ID which would accommodate clients that do not support access paths. Perform the following steps as shown (with example) below:

Note: Microsoft highly recommends that you back up the registry before editing it or making changes to it.

1. From the **Run** command, enter ***REGEDIT***.
2. The **Registry Editor** is launched.
3. In the **Registry Editor**, under **HKEY_CLASSES_ROOT**, look for the required program ID (e.g., Matrikon.OPC.Simulation.1).
4. Expand the program ID branch.
5. Click on the CLSID branch.
6. Record the global universal ID (GUID).
7. Under the **HKEY_CLASSES_ROOT\CLSID** branch, search for the GUID retrieved in the previous step (e.g., ***HKEY_CLASSES_ROOT\CLSID\{024C7FFC-9CC3-476f-867A-BC3CAD79F4FE}***).
8. Create a string value called **ItemIDEscapeCharacter** in the Options branch underneath the GUID (e.g., ***HKEY_CLASSES_ROOT\CLSID\{024C7FFC-9CC3-476f-867A-BC3CAD79F4FE}\Options***).

   Note: To create a new value, use the context menu and select the **New String Value** menu option.
9. Set the value of **ItemIDEscapeCharacter** to a single character, such as "|" (pipe).

   Note: To edit the value, select the entry and select the **Modify** menu option from the **Edit** menu.
10. Shut down the server and restart it for the change to take effect.
11. Configure the **ItemIDEscapeCharacter Item ID** in the OPC client with the access path appended to the end with the configured escape character (e.g., **ITEMID|ACCESSPATH**).

More than one HASP attached to parallel port, but only first one is recognized

Problem:	There is more than one HASP hardware key from MatrikonOPC attached to the parallel port, but the software recognizes only the first one.
Solution:	HASP keys from the same vendor cannot be chained together. Send an email to the MatrikonOPC Support team to obtain a HASP key containing the combined codes of all installed MatrikonOPC servers.

Server cannot be configured; no MatrikonOPC logo in Tool Tray

Problem:	The server cannot be configured because there is no MatrikonOPC logo in the **Tool Tray**.
Solution:	If the server is running as a service, the MatrikonOPC logo does not appear in the **Tool Tray**. To configure the server, start the configuration utility by using the shortcut to the OPC server in the Windows **Start** menu (this shortcut starts the **Configuration** utility

as well as the OPC server).

Note: If the server configuration utility generates an "access denied" error on launch, the DCOM permissions have not been configured properly.

OPC client unable to create or read items under configured object (device)

Problem:	The OPC client is unable to create or read items under a configured object (device).
Solution:	Check the object configuration (for more information, refer to **Server Configuration**). Ensure the **Enabled** checkbox is selected.

Server does not retain settings

Problem:	The server does not retain its previous settings and needs to be reconfigured each time it starts up.
Solution:	In the **Configuration** window, choose **Options** from the **View** menu. Under the **General** tab, ensure that the **Load configuration on start-up** checkbox is selected and that the correct file name appears in the edit box (see **General Options**).

Server cannot save configuration in XML format

Problem:	The OPC server cannot save the configuration in XML format.
Solution:	Microsoft Internet Explorer 4.01 (or later) must be installed to use the XML format.

Server does not show up in the list of locally available program IDs in OPC client application

Problem:	The server does not show up in a list of locally available program IDs in the OPC client application.
Solution:	Make sure that a user with local administrative rights installs the OPC server so that it can create the necessary entries in the registry. Ensure OPCEnum is registered as a service, DCOM permissions are configured properly, and the OPCEnum service is running. Also, especially for older OPC clients that do not use the OPC server browser, make sure the client is running as a user that has *read* access to the registry.

Server does not show up as remote program in OPC client application

Problem:	The OPC server does not show up in a list of remotely available program IDs in my OPC client application.
Solution:	Follow the steps found in **Appendix A – Distributed COM (DCOM)** to copy the program ID from the server machine to the client machine.

Access denied or time-out error when connecting to remove server via DCOM

Problem:	***Access Denied*** error message appears, or time out occurs when trying to connect to a remote OPC server via DCOM.
Solution:	Ensure that the access and launch permissions for the OPC server are set correctly on the server machine. For more information, refer to **Appendix A – Distributed COM**

	(DCOM).

Server does not start up or client cannot connect

Problem:	A client tries to connect to the server and fails, or the server does not start up.
Solution:	Ensure that the access and launch permissions for the OPC server are set correctly. For more information, refer to **Appendix A – Distributed COM (DCOM)**.

About screen and Configuration window do not show up

Problem:	The **About** screen and **Configuration** window do not show up when a remotely connecting client or a client that runs as a service (such as the *Aspen Technologies CIMIO OPC Manager*) tries to launch the OPC server.
Solution:	Either set the identity for the OPC server to the interactive user, or consider running the server as a service. For more information, refer to **Appendix A – Distributed COM (DCOM)**.

DA 2.05 asynchronous I/O does not work

Problem:	After installing an older OPC server, DA 2.05 asynchronous I/O no longer works.
Solution:	The old installation program installed and registered its own (older) version of the OPC proxy/stub DLL. Reinstall the standard OPC proxy/stub DLLs, using an up-to-date installation program or re-register them using the **REGSVR32** command-line utility (refer to the **OPC Compliance - Installation** section in this manual).

Server is using a large portion of CPU resources

Problem:	Server is using 99% of CPU resources.
Solution:	Check the server's Time Granularity settings in the servers **Advanced Options**. Ensure the granularity is not set to *0*. It is recommended that you not set the value lower than *100*.

Trouble communicating through a network

Problem:	Experiencing problems communicating to the product's device through a network.

Solution:	Make sure users can ping the device. Use the **Start** -> **Run** with command ***CMD*** to bring up a command window and type:
	Ping ipaddress `( e.g., C:\temp>ping 127.0.0.1`
	`Pinging 127.0.0.1 with 32 bytes of data:`
	`Reply from 127.0.0.1: bytes=32 time<10ms TTL=128`
	`Reply from 127.0.0.1: bytes=32 time<10ms TTL=128`
	`Reply from 127.0.0.1: bytes=32 time<10ms TTL=128`
	`Reply from 127.0.0.1: bytes=32 time<10ms TTL=128`
	`Ping statistics for 127.0.0.1:`
	`Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),`

```
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms)
```

Where ipaddress is the IP address of the user's <System X> device. If it says "no response", then there are likely network issues between user's computer and the product.

Note: Refer to the **DCOM Manual** to further diagnose any DCOM issues.

Events are not being generated

Problem:	Aliases have been configured to generate events, but a subscribed A&E client is not receiving events as expected.
Solution:	It is possible to disable event generation at the Alias Group level. Make sure that all of the Alias Groups are set to have events enabled.

Cannot validate/add Redundancy Group statistical items to the client configuration

Problem:	The OPC client does not validate the client configuration's Redundancy Group statistical items.
Solution:	Changing the **ItemIDEscapeCharacter** for the SCADA Modbus OPC server from "@" to another character such as "~" solves this issue. Perform the following steps (with example): **IMPORTANT:** Microsoft highly recommends that you back up the registry before editing it or making changes to it. 1. From the **Run** command, enter ***REGEDIT***. 2. The **Registry Editor** is launched. 3. In the **Registry Editor**, under **HKEY_CLASSES_ROOT**, look for the required program ID (e.g., *Matrikon.OPC.ScadaModbus.1*). 4. Expand the program ID branch. 5. Click on the CLSID branch. 6. Record the global universal ID (GUID). 7. Under the **HKEY_CLASSES_ROOT\CLSID** branch, search for the GUID retrieved in the previous step (e.g., ***HKEY_CLASSES_ROOT\CLSID\ {A61B22C3-0656-4F44-B76D-8913D4E9A567}***). 8. Change the string value called **ItemIDEscapeCharacter** in the **Options** branch underneath the GUID (e.g., ***HKEY_CLASSES_ROOT\CLSID\ {A61B22C3-0656-4F44-B76D-8913D4E9A567}\Options***). 9. Set the value of **ItemIDEscapeCharacter** to a single character, such as "~". **Note:** To edit the value, select the entry and select the **Modify** menu option from the **Edit** menu. 10. Shut down the server and restart it for the change to take effect. 11. Configure the **ItemIDEscapeCharacter Item ID** in the OPC client with the access path appended to the end with the configured escape character (e.g.,

ITEMID|ACCESSPATH).

Can I connect it to my computer's serial ports?

Problem:	I don't have any familiarity with RS-485/RS-422. Is it similar to RS-232? How can my computer "talk" to an RS-485/RS-422 device? Can I connect it to my computer's serial ports?
Solution:	Computers come standard with RS-232 serial ports. RS-485 serial ports must either be specially purchased, or an RS-232/RS-485 converter must be purchased. RS-485 and RS-422 allow you to serially chain all of the devices to a single communications port. The devices then take turns on the communication port. This allows reduced cost due to having to provide only a single communication port. This also results in reduced bandwidth since the total bandwidth provided to all devices cannot exceed the bandwidth of the communications port.

Minimizing communications to the devices

Problem:	I wish to minimize communications with my device.
Solution:	Consider which items change frequently and which do not. Having your OPC client or clients read different items at different rates (different OPC group update frequency) can result in reduced communications. If some items are not needed for a considerable period, consider having your OPC client either remove them or disable/deactivate them. MatrikonOPC Modbus will not scan items unless they are actively demanded by at least one OPC client.

Trouble scheduling poll cycles

Problem:	I need more options to schedule my polls at specific times or better manage the fairness of data access among devices.
Solution:	MatrikonOPC Server for SCADA Modbus allows significantly more configuration options and is preferable in SCADA environments.

Noisy communications

Problem:	I am having interference problems when communicating with my SCADA devices.
Solution:	MatrikonOPC Modbus has some head and tail squall squelch capabilities. However, you may consider using MatrikonOPC Server for SCADA Modbus for enhanced features.

Reads and writes time out

Problem:	My device does not seem to wish to communicate. All of my reads and writes time out.
Solution:	Check your configuration to make sure it matches the configuration of your Modbus device. In particular, make sure that the data/parity/stop settings match (serial), make sure you are using the correct node number or numbers, and make sure you have the correct IP address and port number (Ethernet Modbus). If your device is a Daniel (32-bit) device supporting Modbus-like protocol, you will need to use the 3W and 4W registers because this device will send and receive twice as much data per register as a normal Modbus device.

Serial and TCP connections

Problem:	Serial Device links do not work as fast as TCP Device links
Solution:	Serial communications are much slower than TCP communications. As the number of items increase, The time it will take to update all items will increase as well

Questions and Answers

Using PI OPC client

Problem:	How is the PI OPC client to be used with MatrikonOPC servers, if applicable?
Solution:	When using the PI OPC client with MatrikonOPC servers, please go into the PI OPC server **Advanced Options** setting and select the **Enable Mass Tag Adding** checkbox. This causes PI to validate and add all tags in one group at a time rather than one tag at a time, resulting in significant improvements in time and network bandwidth used when initializing. Although this is highly recommended with any MatrikonOPC server, the difference is most noticeable when using products such as MatrikonOPC SCADA servers and Tunneller where each OPC operation has a high latency.

How do I remove the server as a service?

Problem:	How do I remove the server as a service?
Solution:	Re-register the OPC server executable using the ***/REGSERVER*** command-line switch.

Does the server need to be started manually when installed as a service?

Problem:	Does the server need to be started manually when it is installed as a service?
Solution:	No, the first OPC client that tries to connect will start the OPC server.

How can I get OPC data into other applications?

Problem:	How can I get OPC data into *Microsoft Excel*, *Microsoft Word*, or another application with support for *Visual Basic*, *VBA*, or *VB Script*?
Solution:	Use the MatrikonOPC **Automation Component** to create a *Visual Basic* script that connects to a server, creates groups and items, and receives data change updates. Contact **Support** for examples of how to do this in *Excel*, *Visual Basic*, or an *HTML* document.

How can I access individual bits from an integer item?

Problem:	How can I access individual bits from an integer item?
Solution:	In order to do this, users will have to apply a mask to the value, which will null all of the bits in the integer value except the one that users want to keep. For example, if the user wanted to isolate the third bit, they would have to apply a mask that is in binary which will only have the third bit present. In binary, this is the number 4 (0100). To apply the mask, users must use the AND operator, which will leave a 1, if and only if, both the bits in that position are 1.

0110 1100

+ 0000 0100 = 0000 0100

Data bit we want to access

In decimal notation, this would be expressed as 108 AND 4 = 4.

To do this in an OPC server, users would configure an alias to use the IF operator. With the example above, where the user wishes to access the third bit in the integer, the statement would appear as:

IF (INPUT AND 4 = 4, 1, 0)

Therefore, if the bit in position 3 is a 1, since 1 AND 1 =1, the IF statement would be true and would return a 1. If the bit in position is 0, 0 AND 1 = 0, the IF statement would return a false value, which is a 0.

Using the 3W and 4W registers, or the 3 and 4 registers

Problem:	Should I be using the 3W and 4W registers, or the 3 and 4 registers?
Solution:	The difference between the 3W and 4W registers is the assumption as to the width of one register point. For example, in standard Modbus devices, register 4:1 will send and receive 16 bits of data. Daniel devices using a variant of the Modbus protocol (also called Daniel Modbus) use 32-bit registers, and send and receive 32 bits for each register requested or sent. The actual protocol is different, and they are mutually incompatible. A standard Modbus device will not correctly accept data fed to it using the 3W and 4W registers, and a Daniel Modbus device will not correctly accept data fed to it using the 3 and 4 registers. Your Modbus device *User's Manual* should indicate whether the device registers are standard (16-bit) or Daniel (32-bit). Generally, if one type does not work, try the other.

Search the *MatrikonOPC Support Knowledge Base* at **www.opcsupport.com** to find the answers to other commonly-asked MatrikonOPC Server for SCADA Modbus questions.

Licensing

Most MatrikonOPC products require that some form of licensing criteria be met for it to function correctly.

The MatrikonOPC Server for SCADA Modbus supports software **and** hardware licensing.

Note: Alias Events needs to be licensed separately, and by default, is delivered with only a 30-day demo. To purchase a full Alias Events license, please contact your MatrikonOPC Account Manager.

IMPORTANT TO NOTE:

The following licensing information is described in detail within the *Licensing Procedures* document which accompanies the MatrikonOPC Server for SCADA Modbus software and *User's Manual:*

- Hardware and software key licensing information.
- Information about the MatrikonOPC Licensing Utility that is used to license driver software, and the variety of ways in which licenses can be obtained (e.g., Internet Connection, Web Page, Email).
- Licensing Q&A and Troubleshooting.

Contacting Support

The MatrikonOPC Customer Services department (**www.opcsupport.com**) is available 24 hours a day, seven days a week.

Contact MatrikonOPC Support using the information below, or send an email (**support@MatrikonOPC.com**).

For Monday to Friday **daytime support** requests, contact MatrikonOPC Support using the regional phone numbers provided in Table 49.

Region	Office Hours	Contact Information
North America UTC/GMT -7 hours (MST)	8:00 am-5:00 pm	**+1-877-OPC-4-ALL**
South America UTC/GMT -3 hours (BRT)	9:00 am-5:00 pm	**+55 (11) 3475-1846**
Europe /Africa/Middle East UTC/GMT +1 hours (CET)	9:00 am-5:00 pm	**+49-221-969-77-10** **(Request OPC Support)**
Australia/Asia **UTC/GMT + 10 hours (AEST)**	9:00 am-5:00 pm	**+61-2-4908-2198**

Table 49 - MatrikonOPC Support Regional Contact Information

For **after-hours support** in all regions, please use the following number. There is no extra charge from MatrikonOPC for calling their after-hours support number.

Region	Contact Information
All*	+1-780-231-9480

Table 50 - After-Hours Support

*Available only to priority and premium support customers.

OPC Compliance

For more information on OPC, view the documents listed below (as well as other OPC Specifications) at **http://www.opcfoundation.org**. MatrikonOPC supports the following interfaces:

- *OPC Overview 1.0*
- *OPC Common Definitions and Interfaces 1.0*
- *OPC Data Access Specification 2.05a*
- *OPC Data Access Specification 3.00*
- *OPC Alarms and Events Specification 1.10*

Installation

The server is by default installed as a local out-of-process COM server and enters its registration information in the Windows registry when instructed to do so with the **/REGSERVER** command-line option. The server installs itself as a Windows service and enters its registration information in the Windows registry when instructed to do so with the **/SERVICE** command-line option.

- Version-independent program ID is **Matrikon.OPC.SCADAModbus**.
- Version-specific program ID is **Matrikon.OPC.SCADAModbus.1**.
- OPC sub-key and vendor information string.
- Globally unique application ID (e.g., **D7CA0556-C317-4512-8B8C-7543DD7F1626**).
- Globally unique class ID number (e.g., **C33ABAC0-15F9-44bb-B303-1424B2C13C7C**).
- Full path of 32-bit local server executable.
- OPC Data Access Server 2.05a and 3.00 implemented component categories.

The server removes its registration information from the Windows registry when commanded to do so with the **/UNREGSERVER** command-line option.

Common Interfaces

The server supports the mandatory functionality specified in *OPC Common Definitions and Interfaces*. The server supports the following locales for result code translation:

- US English (0x0409)
- System default (0x0800)
- User default (0x0400)
- Neutral (0x0000)

The server allows the client to set the client name for each connection. The server supports the shutdown event notification client-side interface.

Data Access

The server supports the mandatory functionality specified in the *OPC Data Access Custom Interface Standard*.

Groups

The server supports the addition, removal, and cloning of private groups. The server allows the client to manage group state.

- Name
- Active state
- Update rate (100 millisecond resolution)
- Time bias
- Dead-band (percent of analog item high-low range)

Items

The server supports the addition, removal, and validation of items in a group. The server allows the client to manage item configuration.

- Active state
- Requested data type
- Dead-band (percent of analog item high-low range)
- Sampling rate
- Buffer values (used only when sampling rate is enabled and is sampling faster than the group update rate).

Browsing and Item Properties

The server supports address space browsing.

- Hierarchical address space
- Simulate flat address space
- Branch/leaf name filter (pattern matching)
- Item data type filter
- Item access rights filter

The server supports the following mandatory item properties.

- Value, quality, and timestamp
- Canonical data type
- Access rights
- Scan rate

I/O Operations

The server supports synchronous I/O operations.

- Read active item values from cache.
- Read item values from device.
- Read items value from device based on how old the cache items value is.
- Read items without adding the item to an OPC group.

- Write control values to items (on device).
- Write control value, quality, and/or timestamp to items (if the server supports such features).
- Write control values without adding the item to an OPC group.

The server supports asynchronous I/O operations as defined for **DA 3.00**.

- Read items value from device based on how old the cache items value is.
- Write control value, quality, and/or timestamp (if the server supports such feature).

The server supports the data change event notification client-side interface as defined for **DA 3.00**.

- Keep-alive call-back indicating server is alive.

The server supports asynchronous I/O operations as defined for **DA 2.05a**.

- Read items value from device.
- Write control values to items (on device).
- Refresh active items from cache.
- Cancel outstanding operations.
- Enable/disable event notifications.

The server supports the data change event notification client-side interface as defined for **DA 2.05a**.

- Data change or dead-band exceeded.
- Asynchronous refresh operation complete.
- Asynchronous read operation complete.
- Asynchronous write operation complete.
- Asynchronous operation cancellation complete.

The server supports asynchronous I/O operations as defined for **DA 1.0**.

- Read items values from device.
- Read items values from cache.
- Write control values to items (on device).
- Refresh active items from cache.
- Cancel outstanding operations.

The server supports the data change event notification client-side interface as defined for **DA 1.0**.

- Data change or dead-band exceeded – data with timestamps.
- Data change or dead-band exceeded – data without timestamps.
- Asynchronous write operation complete.

Performance Report

MatrikonOPC Server for SCADA Modbus was tested on 20 channels with 30 devices/channel. Below is the Performance Matrix:

OS	Win 2008 R2 64 bit SP1
RAM	8 GB
HDD	320 GB
Type of Communication	TCP Connection
Number of Channels	20
Number of Hosts/Channel	30
Number of Devices/Host	1
Total Number of devices per instance of server	600
Number of OPC Clients	16
Total Number of Items (Read)	48000
Total Number of Items (Write)	400
Number of Groups / Client	4
Number of Items per group	800
Data change rate	2750 items/sec in each group approx

Table 51 – Performance Report

Appendix A Distributed COM (DCOM)

DCOM is an object protocol that enables COM components (such as OPC clients and servers) to communicate directly with each other across a network. A certain amount of configuration is required on the system where the OPC server is installed to allow remote clients to connect to it over the network.

Readers should be familiar with DCOM and with Windows 2000 security features and security administration. Information regarding Distributed COM and various links to related sites, white papers, specifications, and so on, can be found at **http://www.microsoft.com/com/default.mspx**.

Notes:

- The following steps are suggestions only. Ask your Windows Network Administrator for more information about the settings that you should use, especially between different domains.
- The steps provided in this appendix apply to Windows NT operating systems only. For information on how to configure DCOM settings for newer Windows operating systems, please refer to the MatrikonOPC Online Support page on **DCOM Settings**.

DCOM Configuration Utility

Start the DCOM configuration utility either from the server configuration utility or from the command-line (`DCOMCNFG`). Answer, **yes** to any message boxes that appear (allowing the utility to assign application ID entries to those servers that do not already have them).

The main window for ***DCOMCNFG*** allows the user to either configure default settings for all COM servers or else to configure settings for a specific server chosen from the list. The former will affect all servers configured to use the default settings. The latter will affect the selected server only.

Note: DCOM settings are stored in the registry and loaded by COM (and OPC) servers at start-up. Therefore, server processes must be shut down and re-started for these changes to take effect.

Default Properties

The **Default Properties** tab contains settings that affect all DCOM communication on the machine.

- First of all, ensure that the **Enable Distributed COM on this computer** is selected in so that the machine is available to others via DCOM.
- Select the **Enable COM Internet Services on this computer** to allow DCOM access to the machine from the Internet (check with your administrator).
- In general, the other settings do not need to be changed.

The **Authentication Level** specifies when COM should authenticate the identity of calling clients (each call, each packet, etc).

- Normally, it should be set to **Connect**, indicating that COM should authenticate clients when they first connect to a server. If it is set to **None**, then COM performs no authentication and ignores any access permission settings.

The **Impersonation Level** specifies whether servers can ascertain the identity of calling clients and whether they can then perform operations on the client's behalf (as if the server is the client).

- Normally, it should be set to **Identify**, allowing the server to identify the calling client to see if it is allowed access to a certain resource but not to actually access any of these resources as the client.
- Select the **Provide additional security for reference tracking** to make even the reference counting on COM objects secure. This setting is not generally required.

Security Permissions

The most important DCOM settings for an OPC server are the security permissions. There are two ways for you to set these:

1. Change the specific settings for the server (recommended).
2. Change the default settings (not recommended) and make sure that the OPC server will use these.

Either way, be certain that the access and launch permissions are correct for the server.

Setting Security Permissions

To set the security permissions for an OPC Server:

1. Open the DCOM configuration utility.
2. Select the OPC server, and then click **Properties**.
3. The **Distributed COM Configuration Properties** window (Figure 56) appears.

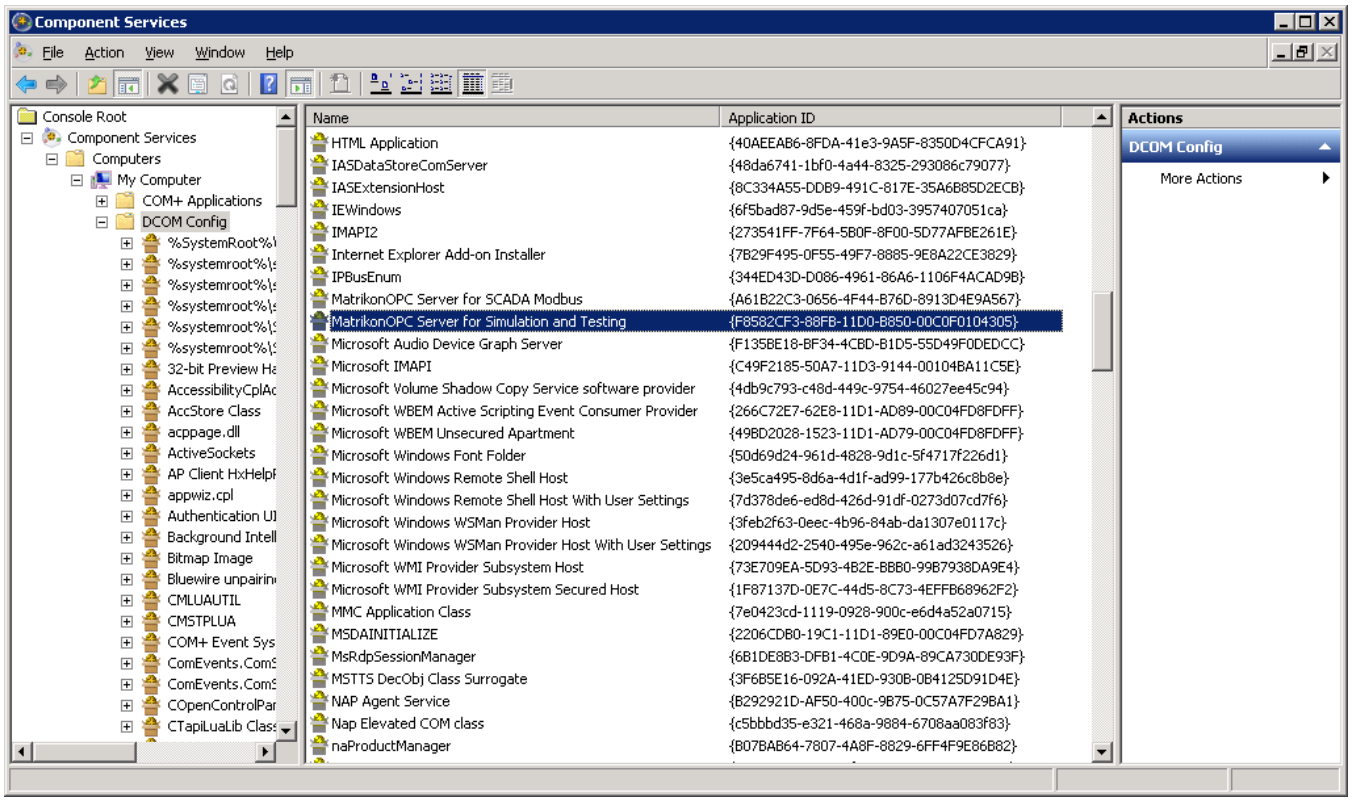


Figure 56 - Distributed COM Configuration Properties Window

4. Right click on the server and select **Properties**.
5. Click on the **Security** tab to set the security for the server.

6. The **Distributed COM Configuration Security** tab (Figure 57) appears.
7. The **Launch and Activation Permissions** contain an *Access Control List* of principals that are allowed to start up a server process or service.

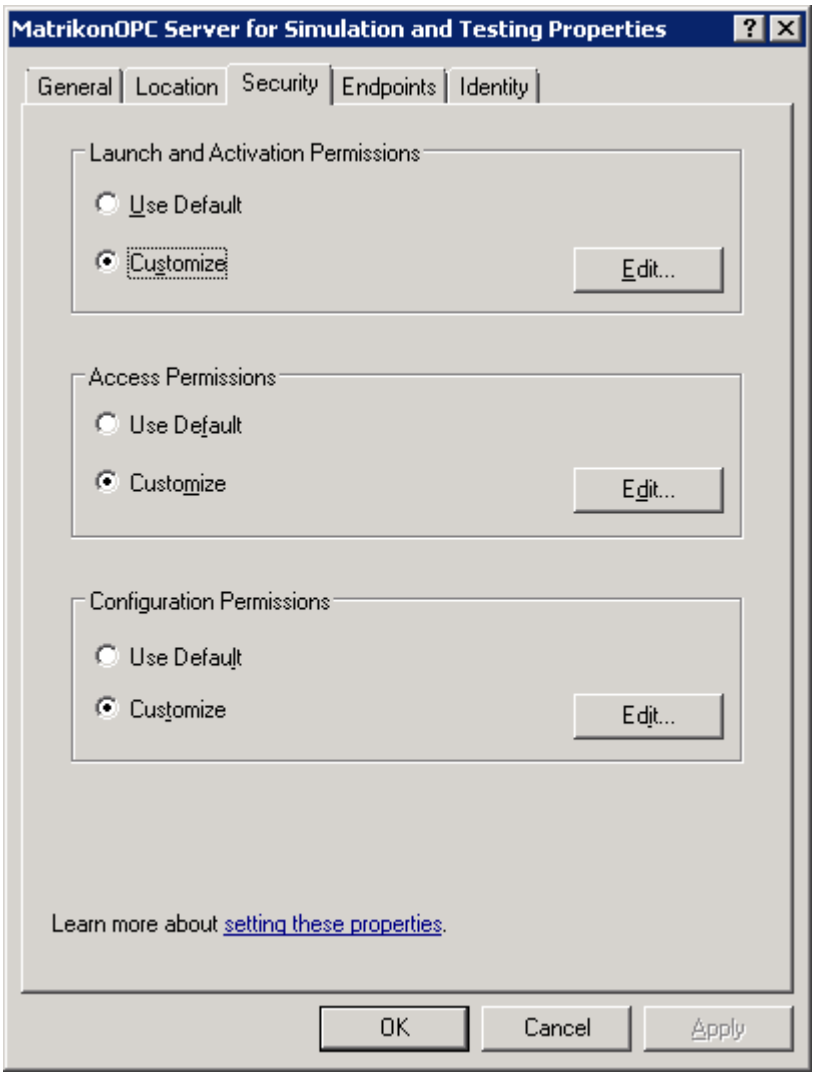


Figure 57 - Distributed COM Configuration Security Tab

Notes:

- The **Launch and Activation Permissions** contain an *Access Control List* of principals that are allowed to start up a server process or service.
- The **Access Permissions** contain an *Access Control List* of principals that are allowed to interact with objects supplied by a server.

8. Include the names of users or user groups from trusted domains that you wish to be able to use the OPC server on this machine. Include the **Everyone** group to allow access to all users on a particular domain.
9. To set the *Access* permissions, click **Use Customize Under Access Permissions** and then click **Edit**.
10. The **Access Permission** window (Figure 58) appears.

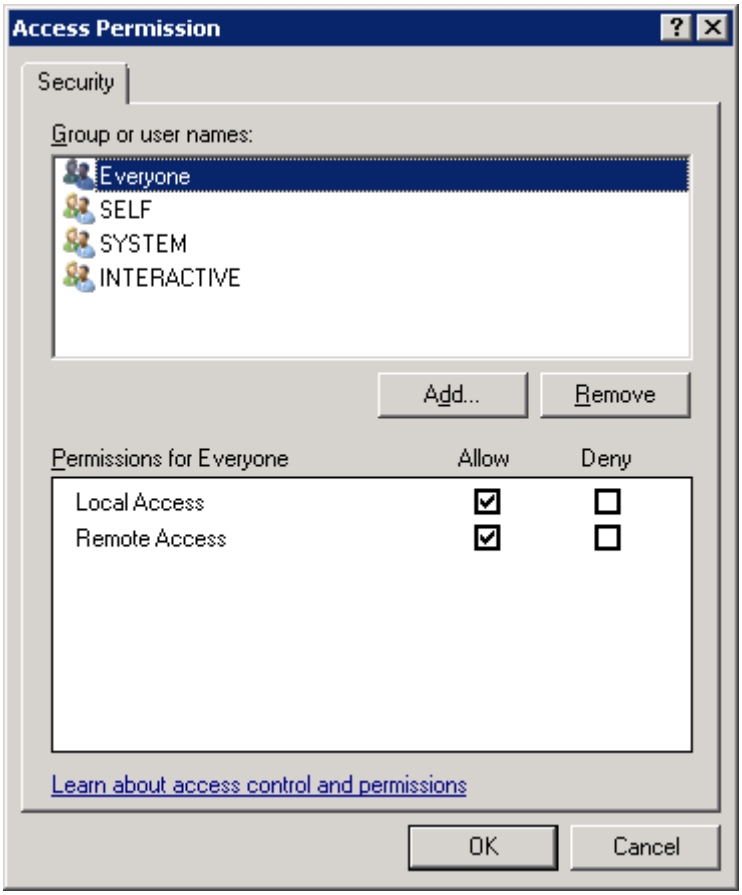


Figure 58 - Access Permissions Window

11. To add users to the list click **Add**.
12. The **Select Users or Groups** window (Figure 59) appears.

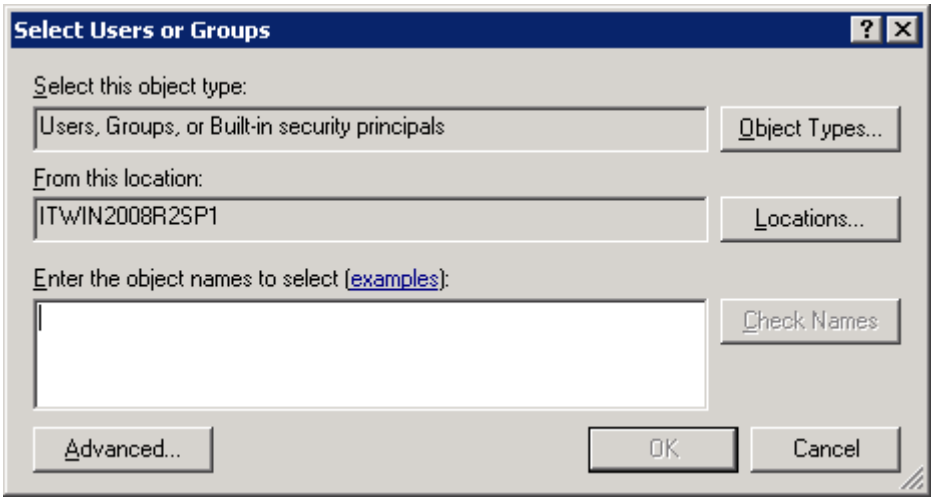


Figure 59 - Add Users and Groups Window

13. To add a user to the list, search and select the user, and then click **OK**. If the user you wish to add does not appear then click **Show Users**.
14. When you are done adding users, click **OK** to close the **Select Users or Groups** window. You can then choose to **Allow** or **Deny access** for each user.

Notes:

- The procedure to set the launch permissions is similar to the above.
- The Configuration Permissions contain an *Access Control List* of principals that are allowed to modify the configuration information for a server. In other words, it indicates who is allowed to alter the registry entries for installed servers as well as who is able to access the registry for the purposes of installing new servers. It is usually simpler to install and configure servers as a user with local administrative rights.

15. To connect to an OPC server from outside of the domain, create a local user account on both the server and the client machine with identical *username* and *password*. Then, add the local user on the OPC server to the DCOM permissions. Use the local account on the client machine to connect to the OPC server.

Server Identity

The **Identity** tab for a selected COM (or OPC) server specifies the user account that should be used when COM starts up the process. The available settings are different for servers that run as local executables as opposed to those that run as NT services.

Note: It is strongly recommended that OPC servers should be installed to run as NT services if they are going to be accessed by remote clients via DCOM. This ensures that the server can always be accessed even if no one is presently logged on to the machine, and only one server process ever starts up. It also adds a greater degree of security in terms of who is able to shut down the server process.

Servers that run as local executables have the option of running as the launching user (the calling client–this is the default), the interactive user (the one currently logged onto the machine), or a specified user. It is usually best to use the interactive user or a specified user. Otherwise, remote clients might start up multiple separate server processes that are not accessible to others.

Servers that run as NT services should generally run as the local **System** account. Alternatively, the server can be set to run as a specified user, although this is usually done from the **Service Control Manager** applet rather than DCOMCNFG. *Access* and *Launch* permissions are particularly important when installing a server to run as an NT service.

Default Protocols

The **Default Protocols** tab specifies the communication protocols available to DCOM. The order that protocols appear in the list indicates the priority in which they will be used (the topmost having the highest priority).

The more protocols that appear in the list, the better the chances of connecting to an OPC server on an unknown remote machine (such as at an OPC Interoperability Workshop). However, it may also take longer for DCOM to time out if a server machine is not present since it has to try each protocol in turn.

For most situations, it is best to remove all unused protocols from the list and only include those that are necessary for your network. For example, on a TCP/IP network, one would include the **Connection-oriented TCP/IP** protocol. Contact your IT personnel for more information about your network.

Note: Evidence indicates that there are problems with the datagram-oriented protocols (such as UDP/IP) that can cause memory leaks in DCOM. Therefore, it is strongly recommended that these protocols be removed from the list of default protocols. Datagram-oriented protocols are not supported under Windows 2000 at all (although the DCOM configuration utility still allows you to configure them).

Remote Program ID

Before the **OPC Server Browser** became available, OPC client applications had to search the registry to generate a list of available OPC servers. Therefore, some older OPC clients need to have a program ID in the local registry in order to connect to a particular OPC server.

The simplest solution is to install the OPC server software on the client machine even if it is not used. Alternatively, use the following steps to copy a program ID to the client machine.

Note: This method may not work for every OPC client. Please check the client documentation for more information.

WARNING: Any changes made to the registry must be made with **extreme caution**!

1. Back up your registry.
2. On the server machine, run **REGEDIT** as a user that has access rights to the local registry.
3. Expand the **HKEY_CLASSES_ROOT** key.
4. Find the program ID(s) for the desired OPC server(s).

   Note: In the case of MatrikonOPC servers, the ID has the form *Matrikon.OPC.Device*. If you quickly type the first few letters then **REGEDIT** should jump to the location of that key. Some servers may have both a version-specific as well as a version-independent program ID. In this case both IDs should be copied to the client machine.
5. For each program ID, select the key and choose **Export Registry File** from the **Registry** menu. Enter a file name, and then click **Save**. Be careful not to overwrite other export files that you are creating.
6. Copy the exported **REG** files to the client machine.
7. Merge the **REG** files into the registry of the client machine.

   Note: This should simply be a matter of double clicking on the file from the desktop of the client machine. Alternatively, run **REGEDIT** on the client machine and choose **Import Registry File** from the **Registry** menu, selecting each file in turn. This must be done as a user who has write access to the local registry.
8. Use **REGEDIT** to check that the program IDs have in fact, been copied.
9. Delete the **REG** files since they are no longer needed.

Appendix B Aliases

Scaling Expressions

Alias scaling expressions may use any of the expression constructs listed in Table 52.

Construct	Description	Examples
'text'	Text value (single quotation marks).	'hello' 'Joe's Alias'
+ - * /	Addition (or Append), Subtraction, Multiplication, and Division.	(3 * 7 + 1) / 8 – 2 'hi' + 'there'
= < > <> <= >=	Equal To, Less Than, Greater Than, Unequal To, Less or Equal To, Greater or Equal To.	3 < 4
Not And Or	Logical Not, Logical And, Logical Or. May also be used for binary math.	(7 >= 3) And (1 <> 0) (12 And 5) = 4
TIME	Current time as text.	'The time is' + TIME
DATE	Current date as text.	'Today is' + DATE
INT(x)	Returns the integer part of x.	INT(3.14) = 3
FRAC(x)	Returns the fractional part of x.	FRAC(3.14) = 0.14
SQRT(x)	Returns the square root of x.	SQRT(16) = 4
DIV(x, y)	Divides x by y.	DIV(42, 7) = 6
IF(expr, x, y)	Returns x if expression evaluates to TRUE, otherwise returns y.	IF(6 <= 12, 'OK', 'Bad') = 'OK'
TRUE FALSE	Constant Boolean true and false.	(5 > 6) = FALSE
TYPEOF(x)	Returns the data type of x as a text description ('STRING', 'INTEGER', 'FLOAT, 'BOOLEAN', 'ERROR').	TYPEOF('hello') = 'STRING'
STR(x)	Converts the numeric value x to text.	STR(3.14) = '3.14'
UPPER(x) LOWER(x) PRETTY(x)	Converts the text value x to all upper case or lower case letters, or the first letter upper case and the rest lower.	UPPER('alarm') = 'ALARM' LOWER('ALARM') = 'alarm' PRETTY('ALARM') = 'Alarm'
COPY(x, start, length)	Returns a sub-string of the x, copying length characters starting at start.	COPY('FIDDLE', 2, 4) = 'IDDL'
FORMATNUMERIC(mask, x)	Converts the numeric value x to text using the format mask.	FORMATNUMERIC('####.#', 123.4567) = '123.5'
INPUT	The incoming value read from OPC.	INPUT * 3.1415
OUTPUT	The outgoing value written to OPC.	OUTPUT / 3.1415

Construct	Description	Examples
'text'	Text value (single quotation marks).	'hello' 'Joe''s Alias'

Table 52 - Expressions Constructs for Alias Scaling

Advanced Calculations

Calculations may contain references to any OPC item in the server's syntax. However, they may not contain references to other alias items. There are no limits on how many items can be used in a calculation. The only limiting factor may be performance related. To reference another OPC item in the calculation, wrap the items full item path in square brackets. For example to add the items ***Com1.Rtu.Reg1*** and ***Com1.Rtu.Reg2*** together, the correct expression would be: ***[Com1.Rtu.Reg1] + [Com1.Rtu.Reg2]***

If the OPC item being referenced in a calculation is not found within the servers address space, the server attempts to add it. If successful, the OPC server uses the **Alias Subscription Rate** to update the OPC item. The **Alias Subscription Rate** is configured in the **Advanced Options** panel.

In the event that the server cannot create a subscription to the OPC item, the alias quality reports *Bad, bad configuration*.

Advanced alias calculations may use any of the operators listed in Table 53.

Construct	Description	Examples
-	Minus operator.	x - y
-	Unary minus operator.	-x
!	Not logical operator. Returns 1 if x is 0, else returns 0.	!x
!=	Not equal logical operator. Returns 1 if true, 0 if false.	x != y
%	Modulo; find remainder of the division of x by y.	x % y
&	And logical operator. Returns 1 if true, 0 if false.	x & y
*****	Multiplication operator.	x * y
/	Division operator.	x / y
^	Calculates x raised to the power of y.	x ^ y
|	Or logical operator. Returns 1 if true, 0 if false.	x | y
+	Addition operator.	x + y
<	Lesser than logical operator. Returns 1 if true, 0 if false.	x < y
<=	Lesser than or equal logical operator. Returns 1 if true, 0 if false.	x <= y

Construct	Description	Examples
==	Equal logical operator. Returns 1 if true, 0 if false.	x == y
>	Greater than logical operator. Returns 1 if true, 0 if false.	x > y
>=	Greater than or equal logical operator. Returns 1 if true, 0 if false.	x >= y

Table 53 - Calculation Operators for Alias Scaling

Alias scaling calculations may use any of the functions listed in Table 54.

Construct	Description	Examples
abs	Absolute value.	abs(x)
acos	Arccosine	acos(x)
asin	Arcsine	asin(x)
atan	Arctangent	atan(x)
avg	Returns the average of a set of values.	avg(v1,v2,v3,...)
bin	Converts a binary number to a decimal number.	bin(binNumber)
ceil	Ceiling.	ceil(x)
cos	Cosine.	cos(x)
cosh	Hyperbolic cosine.	cosh(x)
fact	Returns the factorial of x: x*(x-1)*(x-2)*...(x-x+1).	fact(x)
floor	Floor.	floor(x)
hex	Converts an hexadecimal number to a decimal number.	hex(hexNumber)
if	If the condition is true, returns val1 else returns val2.	if(condition, val1, val2)
isNaN	Returns 1 if the number is a NaN value.	isNaN(value)
log	Natural logarithm.	log(x)
log10	Base-10 logarithm	log10(x)
max	Returns larger of a set of values.	max(v1,v2,v3,...)
max	Returns larger of two values.	max(x,y)
max	Returns larger of three values.	max(x,y,z)
min	Returns smaller of a set of values.	min(v1,v2,v3,...)

Construct	Description	Examples
min	Returns smaller of two values.	min(x,y)
min	Returns smaller of three values.	min(x,y,z)
rand	Random value between 0 and 1.	rand()
rand	Random value between min and max.	rand(min, max)
round	Round the fractional part of a number.	round(x)
sin	Sine.	sin(x)
sinh	Hyperbolic sine.	sinh(x)
sqrt	Square root.	sqrt(x)
sum	Returns the sum of a set of values.	sum(v1,v2,v3,...)
tan	Tangent.	tan(x)
tanh	Hyperbolic tangent.	tanh(x)

Table 54 - Calculation Functions for Alias Scaling

CSV File Format

Note: The CSV file format does not support Alias Event configuration settings. If aliases are exported to CSV and then re-imported, all event generation settings will be lost.

The alias import/export file contains lines of comma-separated values. Each line contains the fields listed in Table 55.

- Values containing commas or spaces are enclosed in double-quotation marks (" ").
- Double-quotation marks occurring inside quoted values are escaped with an extra double-quotation mark.
- Lines beginning with a hash mark (#) are ignored (used for comments).

Field	Description
Alias Group	Name of the alias group folder where the alias belongs. Periods in the name indicate subfolders below the root (e.g., ***North.FIC101*** = alias group ***North*** under the Alias Configuration root and alias group ***FIC101*** under that).
Alias Name	Name of the alias. Full alias path name composed of the alias group path, a period, and the alias name (e.g., ***North.FIC101.CV*** for an alias named ***CV***).
Item Path	Fully qualified item ID for the OPC item to which the alias refers.
Data Type	Decimal number indicating the data type for the alias.
Read-only	Specifies whether the alias is read-only or read-write. 0 = read-write 1 = read-only
Poll Always	Specifies whether the alias is polled continuously or not. 0 = only poll as requested by an OPC client (normal) 1 = poll continuously at the specified update rate
Update Rate	Update rate (milliseconds) to use with the **Poll Always** option.

Field	Description
Scaling	Scaling type for the alias. 0 = none 1 = linear 2 = square root 3 = gain/offset 4 = expression 5 = calculation

Table 55 - File Import/Export Field Descriptions

When the scaling field specifies linear (1) or square root (2) scaling, the line contains the additional fields listed in Table 56.

Field	Description
High Raw	Raw value range (double-precision real).
Low Raw	
High Scaled	Scaled value range (double-precision real).
Low Scaled	
Clamp	Clamp options. 0 = none 1 = clamp low value 2 = clamp high value 3 = clamp both high and low values

Table 56 - Additional File Import/Export Field Descriptions - Linear/Square Root Scaling

When the scaling field specifies gain/offset (3) scaling, the line contains the additional fields listed in Table 57.

Field	Description
Gain	Scaling factor (double-precision real).
Offset	Scaling offset (double-precision real).

Table 57 - Additional File Import/Export Field Descriptions - Offset Scaling

When the scaling field specifies expression (4) scaling, the line contains the additional fields listed in Table 58.

Field	Description
Input Expression	Simple formula applied to the value read from the OPC item.
Output Expression	Simple formula applied to values written to the OPC item. The output formula should be the reciprocal of the input formula.

Table 58 - Additional File Import/Export Field Descriptions - Expression Scaling

When the scaling field specifies calculation (5) scaling, the line contains the additional fields listed in Table 59.

Field	Description
Input Calculation	Calculation applied to the value read from the OPC item.

Table 59 - Additional File Import/Export Field Descriptions - Calculation Scaling

Scaling Algorithms

For each type of scaling, a different algorithm (Table 60) is applied to values read from the OPC item. A reverse algorithm is applied to any values written to the OPC item.

Scaling	Algorithm/Example	Reverse Algorithm/Example
Linear	$(x - Raw_{Low})\left(\frac{Scaled_{High} - Scaled_{Low}}{Raw_{High} - Raw_{Low}}\right) + Scaled_{Low}$	$(x - Scaled_{Low})\left(\frac{Raw_{High} - Raw_{Low}}{Scaled_{High} - Scaled_{Low}}\right) + Raw_{Low}$
Square Root	$\sqrt{x - Raw_{Low}}\left(\frac{Scaled_{High} - Scaled_{Low}}{\sqrt{Raw_{High} - Raw_{Low}}}\right) + Scaled_{Low}$	$(x - Scaled_{Low})^2\left(\frac{Raw_{High} - Raw_{Low}}{(Scaled_{High} - Scaled_{Low})^2}\right) + Raw_{Low}$
Gain/ Offset	$Gain \cdot x + Offset$	$\frac{x - Offset}{Gain}$
Expression	`SQRT(INPUT) + 5`	`(OUTPUT - 5) * (OUTPUT - 5)`

Table 60 - Scaling Algorithms

Appendix C Standard Data Types

The Standard data types and their descriptions are listed in Table 61.

Hex	Dec	Data Type	Description
0000	0	VT_EMPTY	Default/Empty (nothing)
0002	2	VT_I2	2-byte signed integer
0003	3	VT_I4	4-byte signed integer
0004	4	VT_R4	4-byte (single-precision) real
0005	5	VT_R8	8-byte (double-precision) real
0006	6	VT_CY	Currency
0007	7	VT_DATE	Date
0008	8	VT_BSTR	Text (UNICODE)
000A	10	VT_ERROR	Error code
000B	11	VT_BOOL	Boolean (TRUE = -1, FALSE = 0)
0011	16	VT_I1	1-byte signed integer
0012	17	VT_UI1	1-byte unsigned integer
0013	18	VT_UI2	2-byte unsigned integer
0014	19	VT_UI4	4-byte unsigned integer
2002	8194	VT_ARRAY | VT_I2	Array of 2-byte signed integers
2003	8195	VT_ARRAY | VT_I4	Array of 4-byte signed integer
2004	8196	VT_ARRAY | VT_R4	Array of 4-byte (single-precision) real
2005	8197	VT_ARRAY | VT_R8	Array of 8-byte (double-precision) real
2006	8198	VT_ARRAY | VT_CY	Array of currency values
2007	8199	VT_ARRAY | VT_DATE	Array of dates
2008	8200	VT_ARRAY | VT_BSTR	Array of text values
200A	8202	VT_ARRAY | VT_ERROR	Array of error codes
200B	8203	VT_ARRAY | VT_BOOL	Array of Boolean values
2011	8208	VT_ARRAY | VT_I1	Array of 1-byte signed integers
2012	8209	VT_ARRAY | VT_UI1	Array of 1-byte unsigned integers
2013	8210	VT_ARRAY | VT_UI2	Array of 2-byte unsigned integers
2014	8211	VT_ARRAY | VT_UI4	Array of 4-byte unsigned integers

Table 61 - Standard Data Types

Appendix D Microsoft Excel as a DDE Client

Microsoft Excel can act as both DDE client and DDE server. When using Excel as the client, DDE items can be bound to specific cells in a spreadsheet, creating a read-only "hot link" to the DDE items. Using Visual Basic, Excel can also be used to transmit (write) the data contained in a specific cell of a spreadsheet to a DDE server.

DDE Read

To perform a DDE Read using Excel, simply enter a DDE Item as a formula in some spreadsheet cell:

=SERVICE|TOPIC! NAME

Excel will initiate a DDE Advise loop, or "hot link", to the topic and item within the service. The DDE Management system (in this case, Windows) will forward any update sent by the DDE service, to the appropriate DDE client (in this case, Microsoft Excel). The DDE service is responsible for update rate/behaviour.

To disable a specific hotlink in Excel, use the **Links** menu option under the **Edit** menu.

Special Characters in Excel

Excel reserves the use of specific characters for use in any cell. If a ***SERVICE***, ***TOPIC***, or ***NAME*** contains any special characters, surround the specific field with single quotation characters (` `):

=SERVICE|TOPIC!'NAME'

=SERVICE|'TOPIC'!'NAME'

=SERVICE|'TOPIC'!NAME

To include single quotation characters in the name, enter two simultaneous single quotation characters:

=matrikon_opc_simulation_1|'Power.Generator"s Value'!'1000\'

DDE Write

To perform any DDE command other than requests (read) or advise (hot link), a Visual Basic (VB) function must be used. Most DDE commands are available by using the application object. For example, the following function will poke (write) the contents of column 1, row 1 to an item in the OPC Simulation server:

```
Sub WriteToDDE()

' open a DDE channel.  Field 1 = SERVICE, Field 2 = TOPIC
'channel = DDEInitiate("matrikon_opc_simulation", "Bucket Brigade.Real4")

' poke to the Item 1000\, the value of cell A1
DDEPoke channel, "1000\", Worksheets("Sheet1").Range("A1")

' close the channel
DDETerminate channel

End Sub
```

Net DDE

Since DDE commands are routed by the operating system, it is not possible to establish a DDE session between two computers unless some transport mechanism is used. This is usually supplied as a plug-in component of the operating system. In the Windows Environment, the NetDDE Windows Service allows DDE communication between two computers.

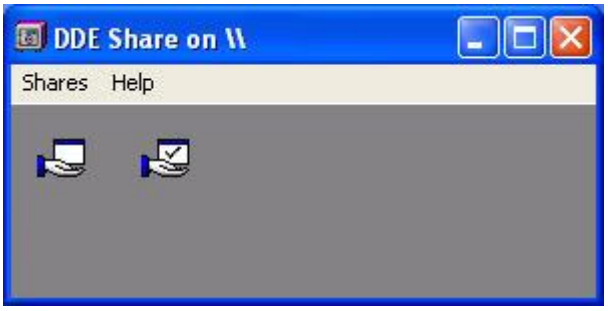


Figure 60 - DDE Share

NetDDE must first be configured using the DDEShare application (*DDEShare.exe*) as shown in Figure 60. Each SERVICE and TOPIC pair, or "NetDDE Share", that is to be available through NetDDE must be assigned a unique name.

Appendix E Configuring Dynamic Data Exchange (DDE)

The server may act as a DDE service. In this way, a DDE client can request data from the server using the following DDE commands:

XTYP_POKE	Write data to an item in the server.
XTYP_REQUEST	Read data from an item in the server.
XTYP_ADVREQ	Read latest data from an active advice loop.
XTYP_ADVSTART	Create and activate an advice loop on a single item in the server.
XTYP_ADVSTOP	Stop and destroy an advice loop on a single item in the server.

DDE service can be enabled or disabled using a checkbox (refer to **Advanced Options** for more information, including the location of, this checkbox). Note that these options will not be available if the server does not support DDE.

Items within the server can be referenced using the following syntax:

SERVICE_NAME |ITEM! SCANDETAILS

SERVICE_NAME represents the name of the DDE service. The first edit box specifies the service name used by this server.

ITEM refers to an Item ID of a point within the server.

SCANDETAILS is a combination of the desired scan rate of the item, and an access path. One or more delimiting characters separate the scan rate and access path. The second edit box can be used to change the delimiters. The delimiting characters must be provided when referencing an item in the server. However, the scan rate and access path are optional. If no scan rate is specified, only data contained in the server cache is available through DDE, and an advise loop is not created.

Changes to the DDE options do not take effect until the server is restarted.

Note: DDE Name Syntax – DDE Names may be described in the following syntax: **SERVICE|TOPIC|NAME**

In this case, consider the **TOPIC** to be the item ID, and the **NAME** to be a combination of the scan rate and access path.

Appendix F Item Quality

Table 62 below shows the scenarios and their corresponding item quality with the tag value.

#	Scenario		Updated Item quality	Hex Value for Item Quality	"LastExceptionCode" Tag Value	"LastExceptionString" Tag Value
1	Valid data received from device.		Good, non-specific	0xC0	0	SUCCESS
2	Connection lost with item having last good sample.		Bad, Last Known Value	0x14	Last received exception code	Last received exception String
3	Connection lost without having last good sample.		Bad, Comm failure	0x18	Last received exception code	Last received exception String
4	Disable Channel /Host /Device		Bad, out of service	0x1C	Last received exception code	Last received exception String
5	The server is connected to the device but it sends an exception as response for requests from server.	01/0x01	Bad, Bad Configuration	0x04	1	ILLEGAL FUNCTION
		02/0x02	Bad, Bad Configuration	0x04	2	ILLEGAL DATA ADDRESS
		03/0x03	Bad, Bad Configuration	0x04	3	ILLEGAL DATA VALUE
		04/0x04	Bad, Device Failure	0x0C	4	SLAVE DEVICE FAILURE
		05/0x05	Uncertain, non specific	0x40	5	ACKNOWLEDGE
		06/0x06	Uncertain, non specific	0x40	6	SLAVE DEVICE BUSY
		07/0x07	Uncertain, non specific	0x40	7	NEGATIVE ACKNOWLEDGE
		08/0x08	Bad, sensor Failure	0x10	8	MEMORY PARITY ERROR
		10/0x0A	Bad, Bad Configuration	0x04	10	GATEWAY PATH UNAVAILABLE
		11/0x0B	Bad, Device Failure	0x0C	11	GATEWAY TARGET DEVICE FAILED TO RESPOND

Table 62 – Item Quality

Appendix G Security

Introduction

The OPC Framework has implemented the OPC Security Customer Interface version 1.0 of the specification. Both the *IOPCSecurityNT* and *IOPCSecurityPrivate* interfaces are fully supported (there are no optional pieces).

IOPCSecurityNT describes how NT authentication is used in OPC Security. *IOPCSecurityPrivate* describes how proprietary user credentials are used in OPC Security. The OPC Security specification defines how user credentials are passed from client to server, but very little else. The type of functionality in an OPC server that is controlled by OPC Security is vendor-determined.

Appendix Structure

This appendix is structured as follows:

- **Introduction** – this introductory section.
- **OPC Security Implementation** – description of user access dependencies.
- **MatrikonOPC Tag Security Configuration Utility** – describes the utility that is used to configure security for MatrikonOPC servers.
- **Security Details** – describes how permissions are applied, how security settings are stored, and other important details.

OPC Security Implementation

Whether a user has access or not will depend on:

1. **Identity** – the user behind the OPC client's actions.
2. **Location** – the OPC item ID or hierarchy path being accessed.
3. **Type** – the type of functionality the user is trying to perform.

Identity

Using the *IOPCSecurityNT* interface for OPC Security, the identity of the OPC client is the domain user that is running the OPC client and connecting to the OPC server. The OPC Framework will attempt to impersonate the client that is connected to the server, and use the name from those credentials for access control. This method does not require any implementation of OPC Security by the OPC client, since no interface calls are necessary.

Using the *IOPCSecurityPrivate* interface for OPC Security, the OPC client must make a call to the OPC server to log on with a user name and password. In this scenario, the OPC client must implement and follow the OPC Security specification in terms of following the calls necessary to provide credentials. The credentials provided through the private interface override those provided from the NT interface. The OPC client may choose to log off, at which point the NT credentials are used until the log on call is made again.

Location

Some of the interface calls made from an OPC client to an OPC server require the client to provide an OPC item ID. The OPC Framework will take the OPC item ID provided, and determine whether access to that location is allowed or not. A particular server installation may want to control access to a particular device or branch in the hierarchy, and is able to do so by setting permissions for a partial item ID or path.

Other interface calls do not provide an ID, but may depend on the state of the server with respect to previous calls. OPC Browsing (DA 2.0) is an example of this. In this case, the full path up to that particular location is used for determining access rights.

Type

There are four types of interface functionality in the OPC Framework (specifically the OPC Outer Layer) that are controlled using OPC Security:

1. Browsing
2. Adding items
3. Reading
4. Writing

Browsing is the least intrusive, followed by adding items, reading values from the items, and finally writing to the items being the most intrusive. There are multiple interfaces, and functions within those interfaces, that can be used to perform the types of functionality listed above. Thus, security control has been added to each possible entry. For example, read control has been added to group subscriptions call-backs as well as synchronous reads and asynchronous reads – denying read control would disable reads for the possible types.

A location that is being accessed by an OPC client may have different permissions for the different types of access. For example, one particular item may be browseable, added, and read, but the writing permissions might be restricted so that the user cannot interfere with the operation of the end device.

Note: Of the four functionality types listed above, three of them make use of the full OPC item ID when an interface call is made from the client to the server: Adding Items, Reading, or Writing. When browsing, only a partial path leading to an OPC item may be known at any location in the browsing hierarchy.

MatrikonOPC Tag Security Configuration Utility

The tag security settings for MatrikonOPC servers are configured and managed via the **MatrikonOPC Tag Security Configuration Utility**, a tool that is installed with the server. This utility allows users to define, per MatrikonOPC server, the following:

- Who has permission
- Which tags/locations they have permissions for
- The type of permissions for those locations

Security settings configured using this utility are read by the OPC Framework and populated in memory upon server start-up, and are monitored for changes.

Note: In order for the **MatrikonOPC Tag Security Configuration Utility** to work properly, the windows user running the utility must have Administrator rights to the computer.

The **MatrikonOPC Tag Security Configuration Utility** is a component of an OPC server installation, and is installed in the following location:

C:\Program Files\Common Files\MatrikonOPC\Common\ACLConfig.exe

The **MatrikonOPC Tag Security Configuration Utility** can be run either directly from its installed location, or from a shortcut found in the **Start Menu**. If the utility is run from the installed location, then the user must select the OPC server for which security will be configured. If the **MatrikonOPC Tag Security Configuration Utility** is run from the shortcut, then it will configure security for the OPC server to which the shortcut belongs.

To run the MatrikonOPC Tag Security Configuration Utility from its shortcut:

1. Click on the Windows **Start** button.
2. Select **Programs -> MatrikonOPC -> <*OPC Server Name*> ->Tag Security**.

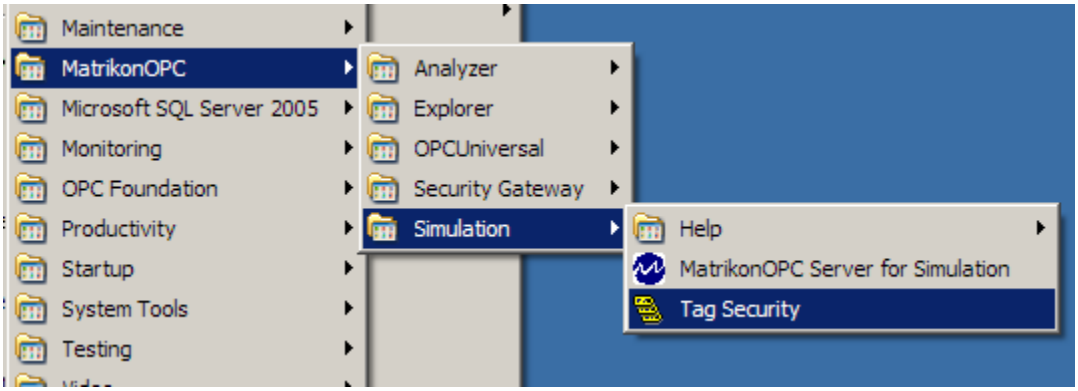


Figure 61 - Start Menu Navigation

After the MatrikonOPC Tag Security Configuration Utility is run:

1. A window appears asking for a password to be entered. This protects the security settings and prevents unauthorized individuals from accessing these settings. For servers that required a security password to be entered during their installation, the **Enter server password** window (Figure 62) appears.

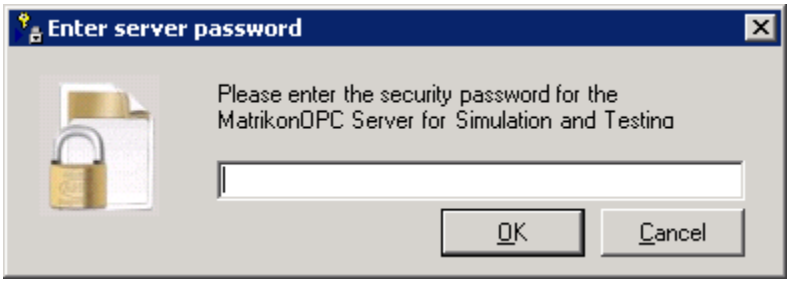


Figure 62 - Enter Server Password Window

2. Enter the password created during the installation of the OPC server.

   Note: The default password provided for you during installation (if you chose to use the default) is ***MatrikonOPC***.

3. Click on the **OK** button.
4. The **MatrikonOPC Tag Security Configuration Utility** screen (Figure 63) appears.

Layout

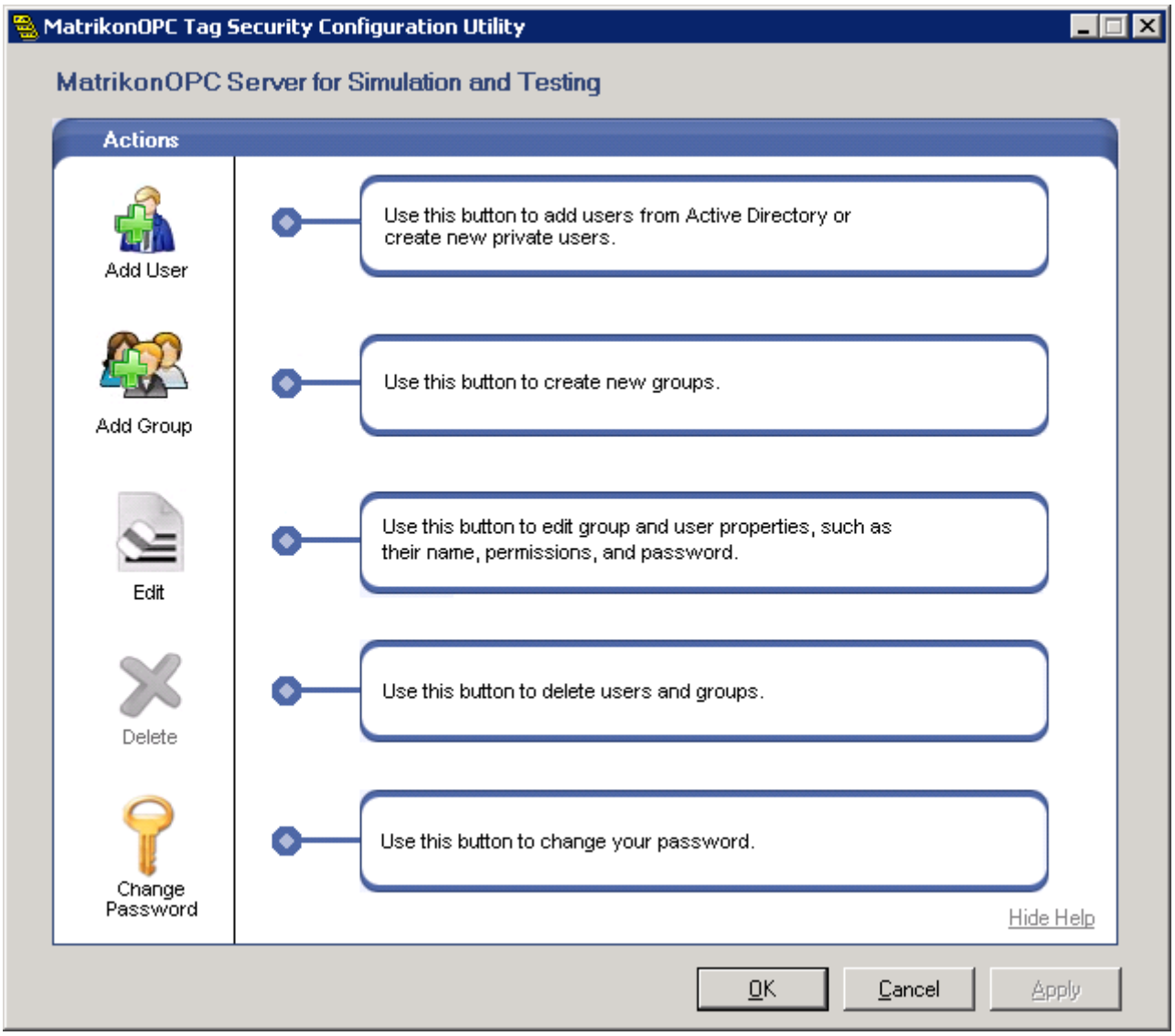


Figure 63 - MatrikonOPC Tag Security Configuration Utility Main Screen

The **MatrikonOPC Tag Security Configuration Utility** screen displays the name of the OPC server that is being configured for tag security. In Figure 63 above, the **Utility** screen shows that the ***MatrikonOPC Server for Simulation and Testing*** tag security is being configured.

The left-hand side of the **Utility** screen shows a column of action icons that can be clicked upon. These actions are further described later in this appendix. Some actions may not be applicable, depending on the context in which they are used, and are identified by greyed-out icons.

In the center of the **Utility** screen there are descriptions for each of the actions that can be performed. The appearance of these descriptions can be toggled at any time by clicking on the **Hide Help/Show Help** link that exists in the bottom-right corner of the **Utility** screen. When the **Help** screen is hidden, a list of users and groups that have been added appears (Figure 64). By default, there is always a user named ***Everyone*** in the **Users & Groups** list. This user defines the permissions used for other NT users that are not configured.

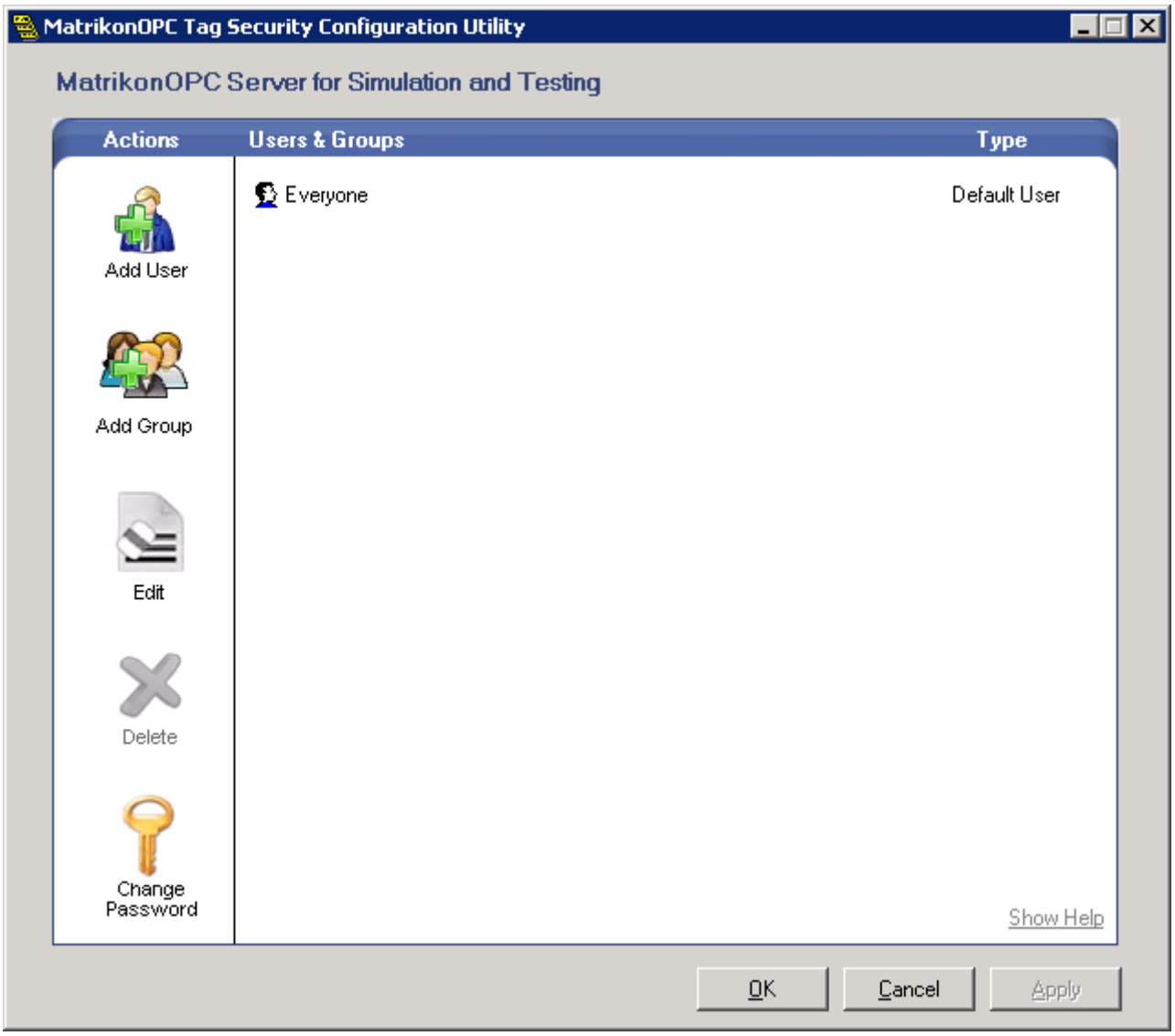


Figure 64 - MatrikonOPC Tag Security Configuration Utility Main Screen: Users & Groups

Note: After any action is performed, the changes made do not take effect until either the **OK** or **Apply** button has been selected. Clicking on the **OK** button saves the changes and closes the **Utility** screen. Clicking on the **Apply** button saves the changes and keeps the **Utility** screen open. Clicking on the **Cancel** button or the **X** button on the top-right of the screen, discards any changes made.

Add User

There are two types of users for which security can be configured: *NT Users* and *Private Users*. Adding a user to the configuration will allow permissions to be defined for that user. For NT users, the *IOPCSecurityNT* interface is used, and for Private users, the *IOPCSecurityPrivate* interface is used.

NT User

To add an NT user:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, click on the **Add User** button.

2. The **Add User** window (Figure 65) appears. The **NT User** option button is selected by default. This will allow the addition of a user from a Windows NT Domain.

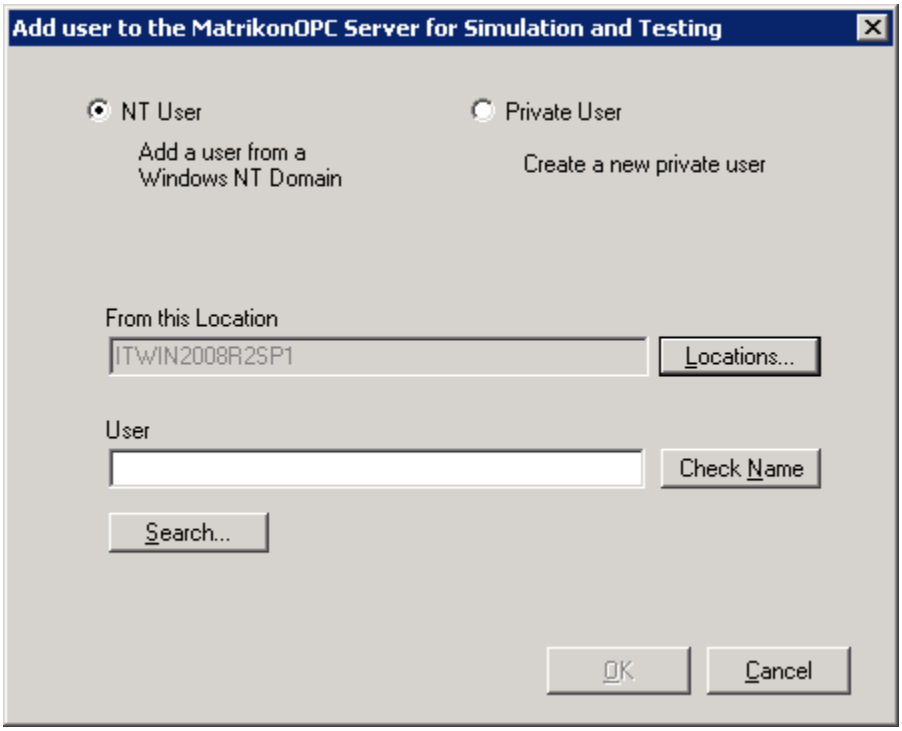


Figure 65 - Add User Window (NT)

3. By default, the local machine name appears in the **From this Location** field. To browse to a different location, click the **Locations** button.
4. The **User** field can be entered manually. Clicking on the **Check Name** button performs validation on the entered user. The user names are the same as the Windows NT domain names used on the network. To avoid manually entering a user, click on the **Search** button, and select a user from the displayed list of users.
5. To add the selected user, click on the **OK** button. The user is then added to the list of users and/or groups currently defined (refer to Figure 66 for a sample list of defined users and groups).
6. The **Edit <Username>** window (Figure 71) appears. Follow the instructions in the **Edit User** section for adding permissions.

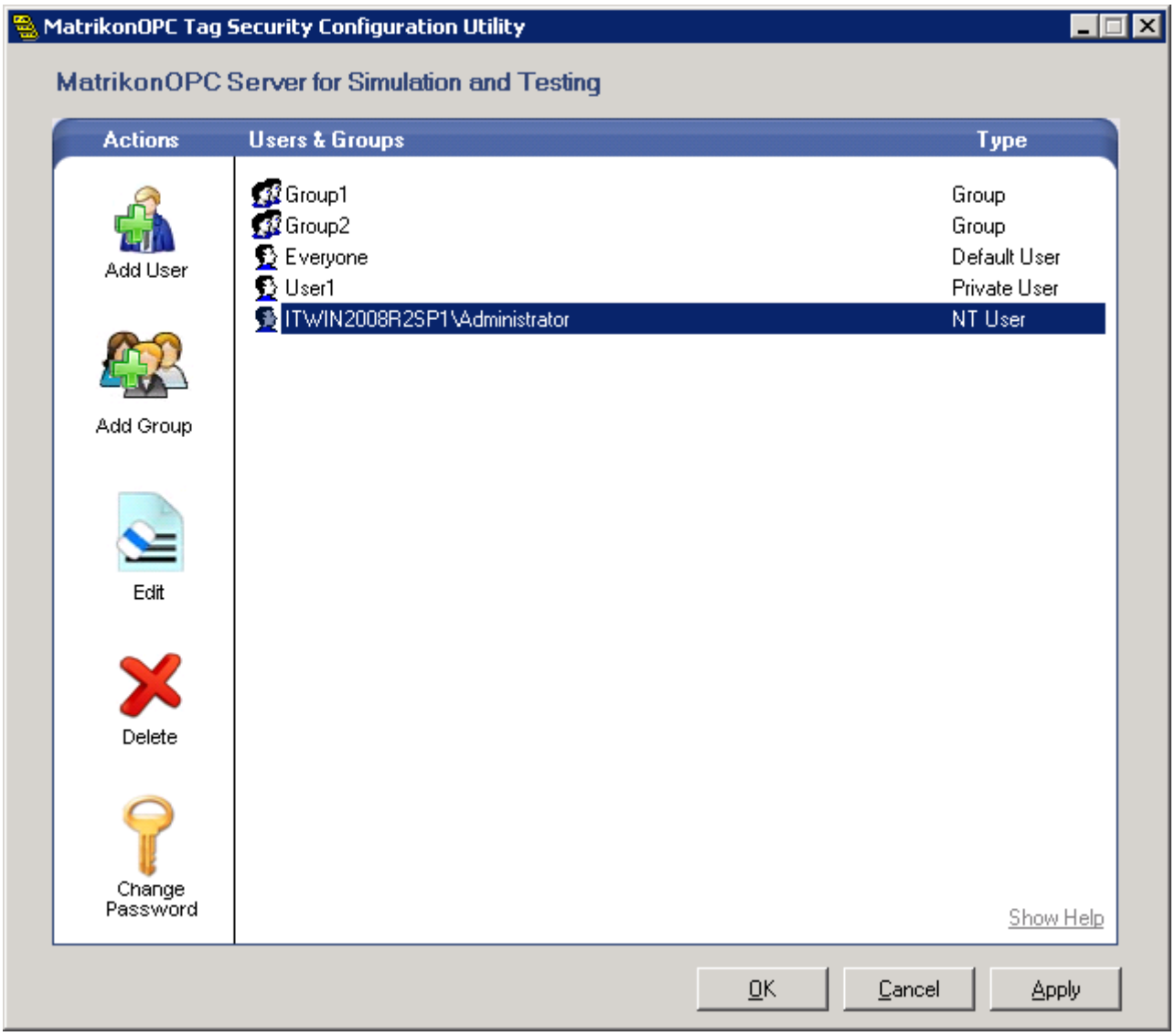


Figure 66 - Defined Users and Groups

Private User

Private users have names and passwords associated with them, custom to the OPC client application, which must implement the OPC Security Private interface for this section to be meaningful. Private users exist only in the OPC server. They are not created in any NT domains.

To add a Private user:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, click on the **Add User** button.
2. The **Add User** window (Figure 65) appears.
3. Select the **Private User** option button (Figure 67).

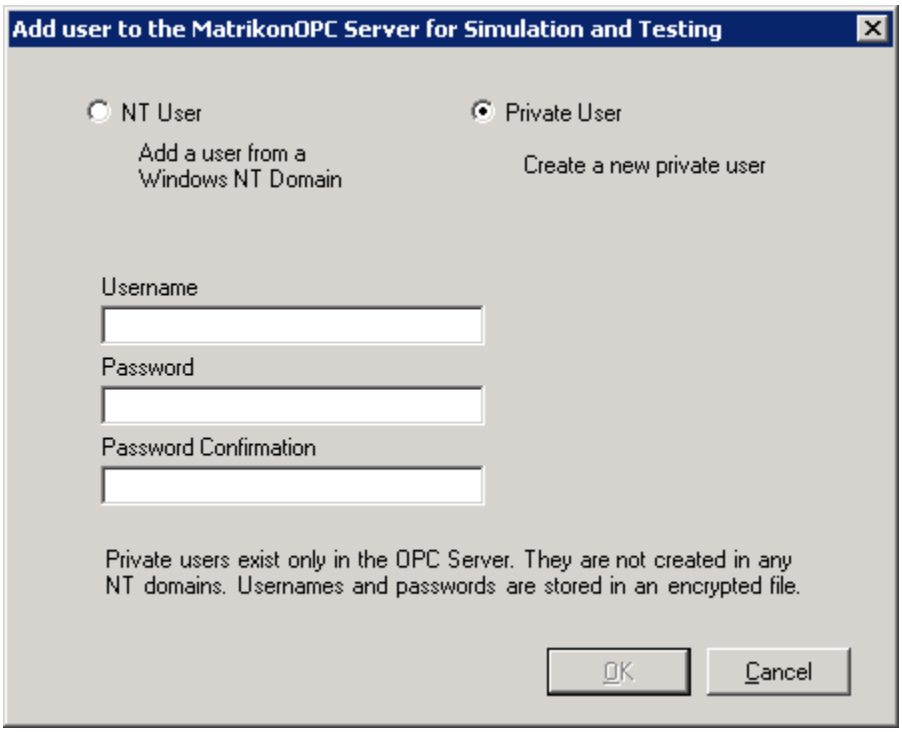


Figure 67 - Add User Window (Private)

4. In the **Username** field, enter a user name.
5. In the **Password** field, enter a password.
6. In the **Password Confirmation** field, re-enter the new password.
7. Click **OK** to create the Private user. It will be added to the list of users and/or groups currently defined (refer to Figure 66 for a sample list of defined users and groups).

   Note: The user name cannot be added if there is already a configured User or Group with that same name.

Add Group

Groups are used to group users together so that they can share common permissions. Groups are not necessary to use OPC Security. However, they are convenient when you need to apply common permissions to multiple users. A group must contain at least one user, and any combination of users can be grouped together.

Note: A user can exist in multiple groups, but if two or more of the groups have the same configured path (refer to the **Edit User** section), and each group defines a permission (Allow/Deny) for the type of access, then the group whose name alphabetically comes first is used.

There are two types of groups for which security can be configured: **NT Groups** and **Private Groups**.

NT Group

To add an NT group:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, click on the **Add Group** button.
2. The **Add Group** window (Figure 68) appears. The **NT Group** option button is selected by default. This will allow the addition of an NT group from a Windows NT Domain.

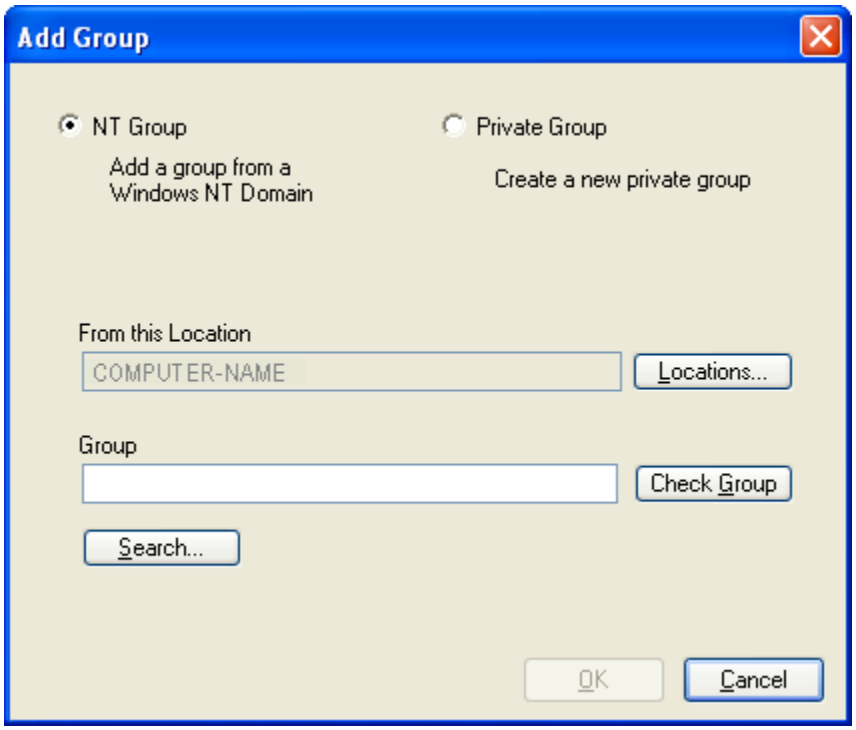


Figure 68 - Add Group Window (NT)

3. By default, the local machine name appears in the **From this Location** field. To browse to a different location, click the **Locations** button.
4. The **Group** field can be entered manually. Clicking on the **Check Group** button performs validation on the entered group. The group names are the same as the Windows NT domain group names used on the network. To avoid manually entering a group, click on the **Search** button, and select a group from the displayed list of groups.
5. To add the selected group, click on the **OK** button. The group is then added to the list of users and/or groups currently defined (refer to Figure 66 for a sample list of defined users and groups).
6. The **Edit <Groupname>** window (Figure 78) appears (for NT Groups, there is only the **Permissions** tab). Follow the instructions in the **Edit Group** section for adding permissions.

Private Group

To add a Private Group:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, click on the **Add Group** button.
2. The **Add Group** window (Figure 68) appears.
3. Select the **Private Group** option button (Figure 69).

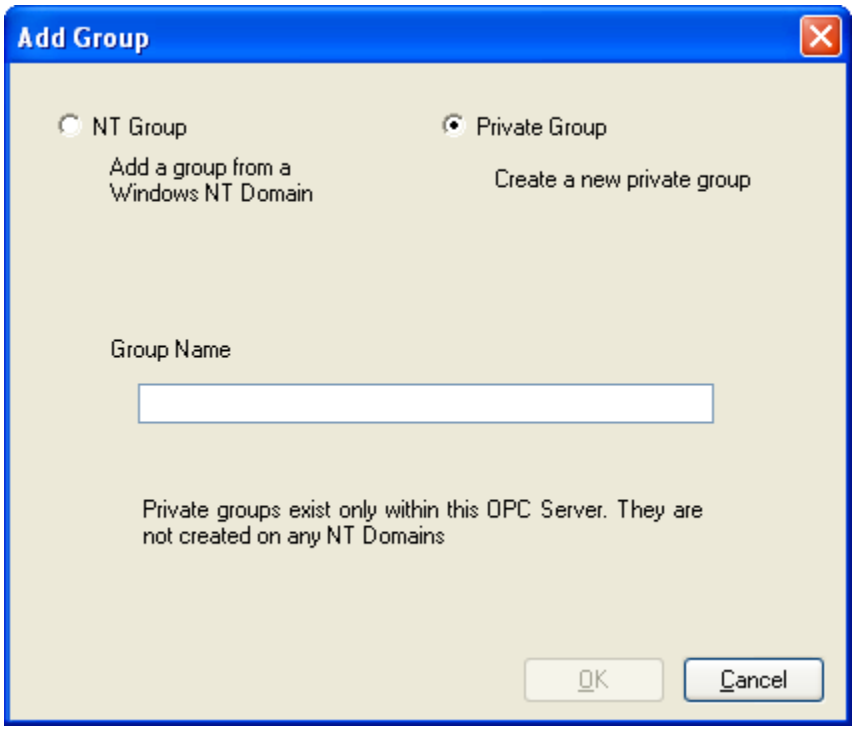


Figure 69 - Add Group Window (Private)

4. Enter a name for the group to be created. The group is private and will only exist in the OPC server. The group cannot be created if there is already a configured user or group with that same name.
5. Click on the **OK** button to create the group. It will be added to the list of users and/or groups currently defined (refer to Figure 66 for a sample list of defined users and groups).
6. The **Edit <Groupname>** window (Figure 78) appears. Follow the instructions in the **Edit Group** section for adding permissions.

Duplicate

Another way to create a new user or group is to duplicate an existing one.

To create a new user or group via duplication:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, right-click your mouse on the user or group you wish to duplicate.
2. From the displayed menu, select ***Duplicate***.
3. The **Enter New Name** window (Figure 70) appears.

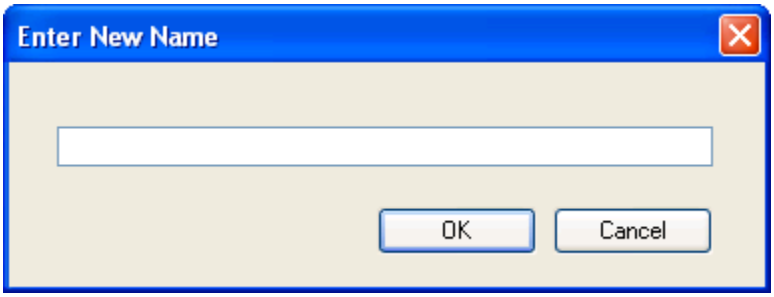


Figure 70 - Enter New Name Window

4. Enter a name for the new user or group to be created.
5. Click on the **OK** button. There must not already be a configured user or group with that same name. The new user or group will then appear in the list, and will contain the same properties as the duplicated parent (i.e., original) user or group.

Edit

Once the desired users and/or groups are defined, they must be edited to configure their respective permissions.

Note: Permissions can also be defined for the user ***Everyone***, which is defined as the default user. For every NT user not configured in the MatrikonOPC Tag Security Configuration Utility, the user ***Everyone*** is used for determining security permissions.

Permissions for allowing/denying browsing, adding items, reading items, and writing items can be configured. Permissions are defined for users and groups for a particular location in the server's address space hierarchy. A location can either be a tag (i.e., OPC Item ID) or a path leading up to an item.

Edit User

To edit a user:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, either:
   - Double-click your mouse on the user you wish to edit, or
   - Right-click your mouse on the required user and select **Edit User** from the displayed menu, or
   - Click your mouse on the required user, and then select the **Edit** button.
2. The **Edit <Username>** window (Figure 71) appears.

   Notes:
   - In this example, the user being edited is ***User1*** so the window's title reads **Edit User1**.
   - There are two tabs on this window: **Permissions** and **General**. The **General** tab only appears for private users.

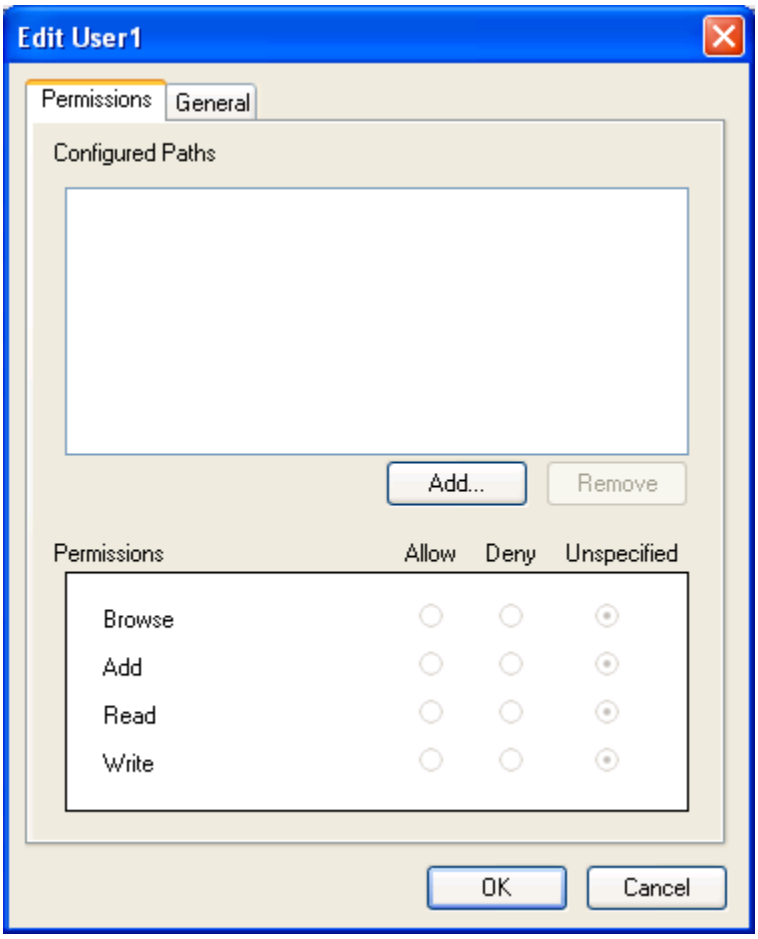


Figure 71 - Edit <Username> Window (Permissions Tab)

Permissions

On the **Permissions** tab, the permissions associated with this user relating to browsing, adding, reading, and writing for any specific paths or OPC items are configured. The **Configured Paths** screen section lists the paths and/or OPC items for which permissions are defined. The **Permissions** screen section indicates those permissions with option buttons, which can be set to ***Allow***, ***Deny***, or ***Unspecified***. Selecting ***Unspecified*** directs the OPC Framework to look for the permission higher up in the hierarchy.

From the **Permissions** tab, when the **Add** button is selected, the **Add Permission** screen is displayed. In the **Server Address Space** screen section, folders and items are shown as nodes are selected and expanded.

Red folders indicate a null branch. This means that the item names below this branch do not contain the branch name, so permissions set on the branch will not reflect on ***Add***, ***Write***, and ***Read*** permissions for the items below the branch. To properly set permissions on the items below the branch, they have to be set on the items themselves.

Figure 72 provides an example of a null branch and items below it, all of which are outlined in black.

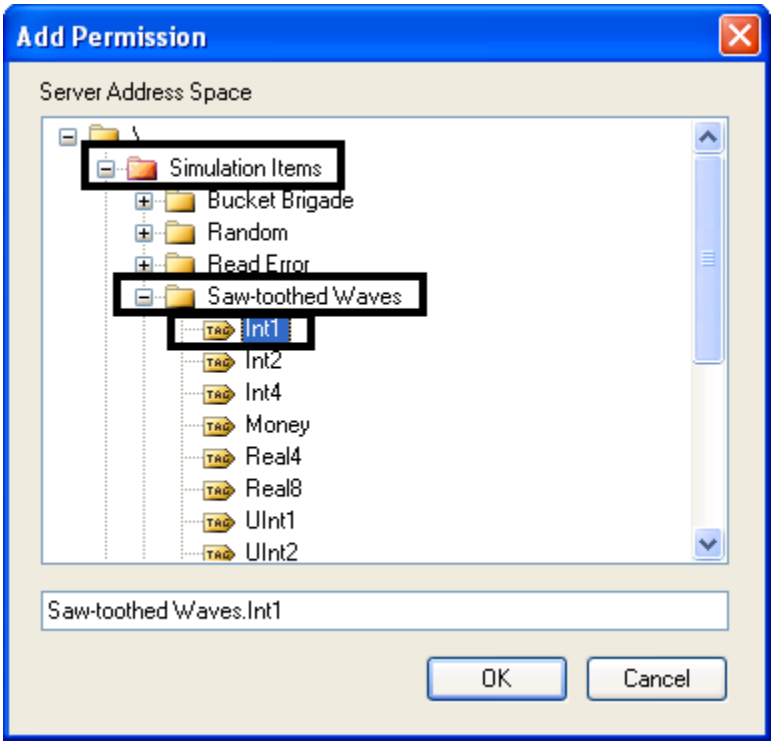


Figure 72 - Add Permission Screen (Server Address Space)

When browsing, using the example in the diagram above, ***Simulation Items*** is a null branch (identified by the red folder), ***Saw-toothed Waves*** is a non-null branch, and ***Int1*** is an item.

Setting permissions on the ***Simulation Items*** branch does not affect ***Read***, ***Write***, and ***Add*** permissions for the ***Saw-toothed Waves*** branch of the ***Int1*** item below that. This is because when adding the ***Int1*** item, the item ID of the ***Int1*** item is actually ***Saw-toothed Waves.Int1*** and not ***Simulation Items\Saw-toothed Waves.Int1***.

Setting ***Read***, ***Write***, and ***Add*** permissions for the ***Saw-toothed Waves*** branch will affect the items below as ***Saw-toothed Waves*** is part of the item ID of the ***Int1*** item (***Saw-toothed Waves.Int1***), and therefore the ***Saw-toothed Waves*** folder does not appear as a red folder like the ***Simulation Items*** folder does.

Note: User permissions override group permissions. For example, if a defined group is denied writing privileges to an OPC item, ***Random.Int4***, while a defined user (a member of that group) is allowed writing privilege to Random items (***Random.<itemID>***), then the user permissions would take precedence for that OPC item (even though the permissions relating to the group are a better match for the OPC item, ***Random.Int4***).

To edit a user's permissions:

1. Click on the **Add** button to begin configuring permissions for the current user.
2. The **Add User Permission** window (Figure 73) appears. This window allows you to browse the OPC server address space to find either a path in the hierarchy, or a particular OPC item.

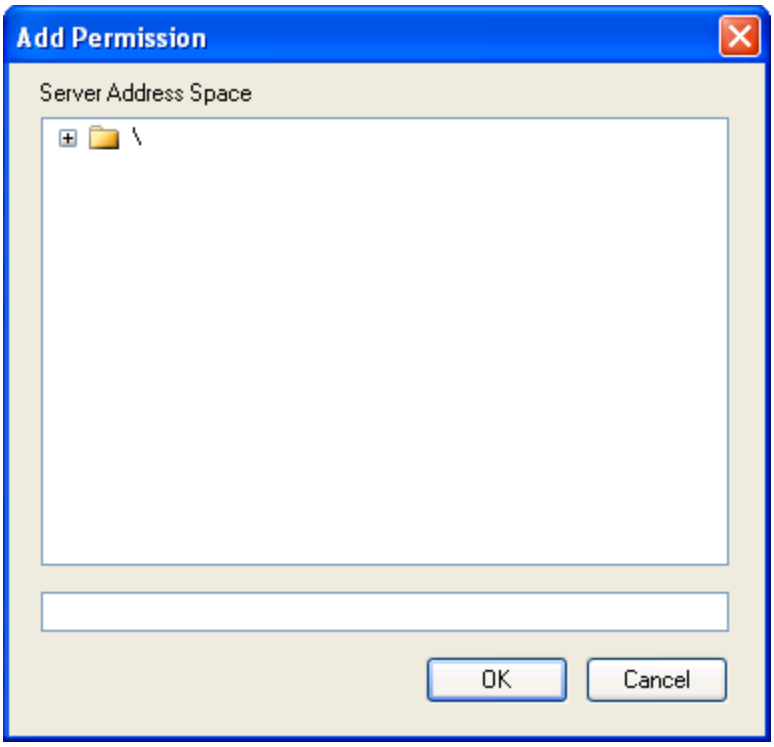


Figure 73 - Add User Permission Window

3. Expand on the server address space's root node and continue to browse to the desired path in the hierarchy, or the desired OPC item.

   Note: Branches (i.e., folders) that are red in colour (see Figure 74) are not part of an item's path, but rather are used for browsing organization. Configuring permissions for adding, reading, and writing will not affect items below it. For more information, refer to **Permissions**.

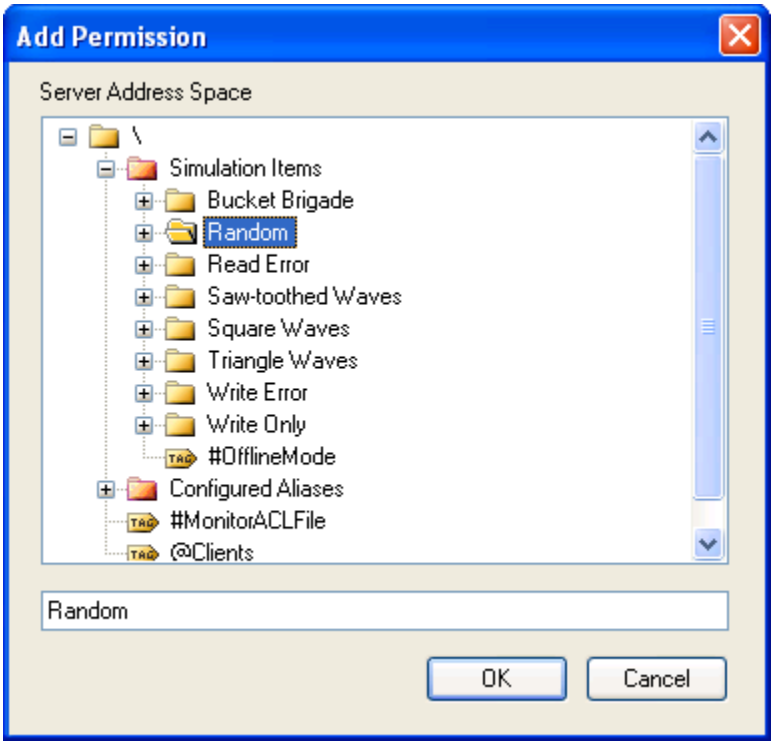


Figure 74 - Add Permission Window: Browsing Server Address Space

4. Click on the **OK** button.
5. The selected path or item will be listed in the **Configured Paths** screen section of the **Edit User** window. Permissions can now be assigned, as in the example depicted in Figure 75.

   Note: In this example, ***User1*** is allowed to browse the ***Random*** path of the OPC server's address space. Adding privileges are denied, and Reading and Writing privileges are unspecified.

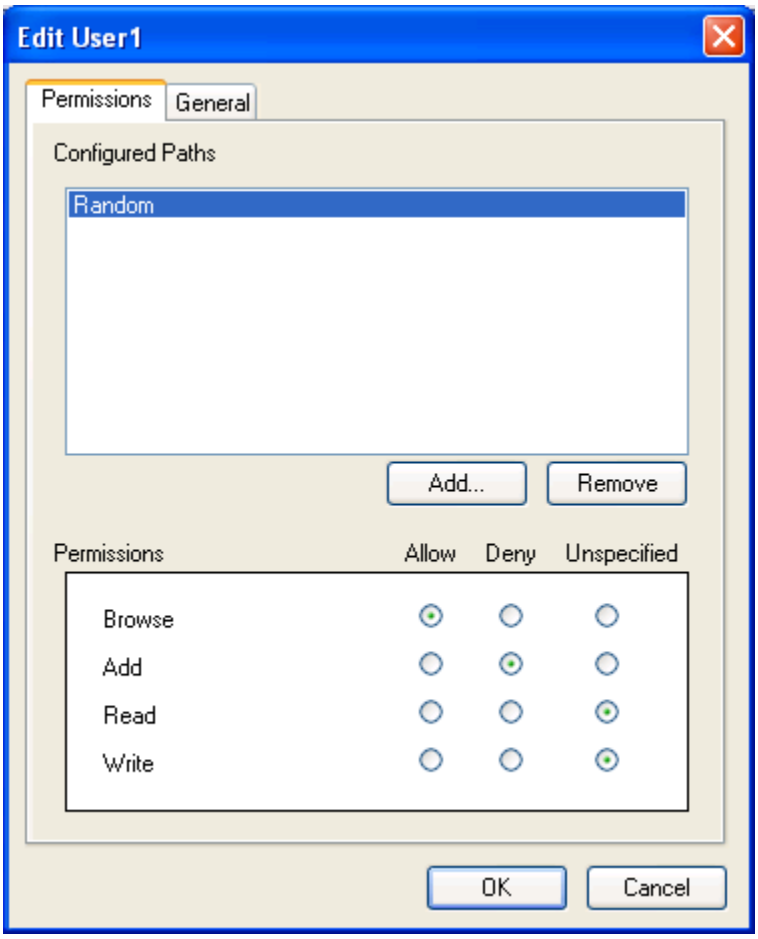


Figure 75 - Edit <Username>: Example

On the **General** tab (Figure 76) of the **Edit <Username>** window, the name and password of the user can be changed.

To edit a user's name:

1. Enter a new name in the **Name** field.
2. Click on the **OK** button.

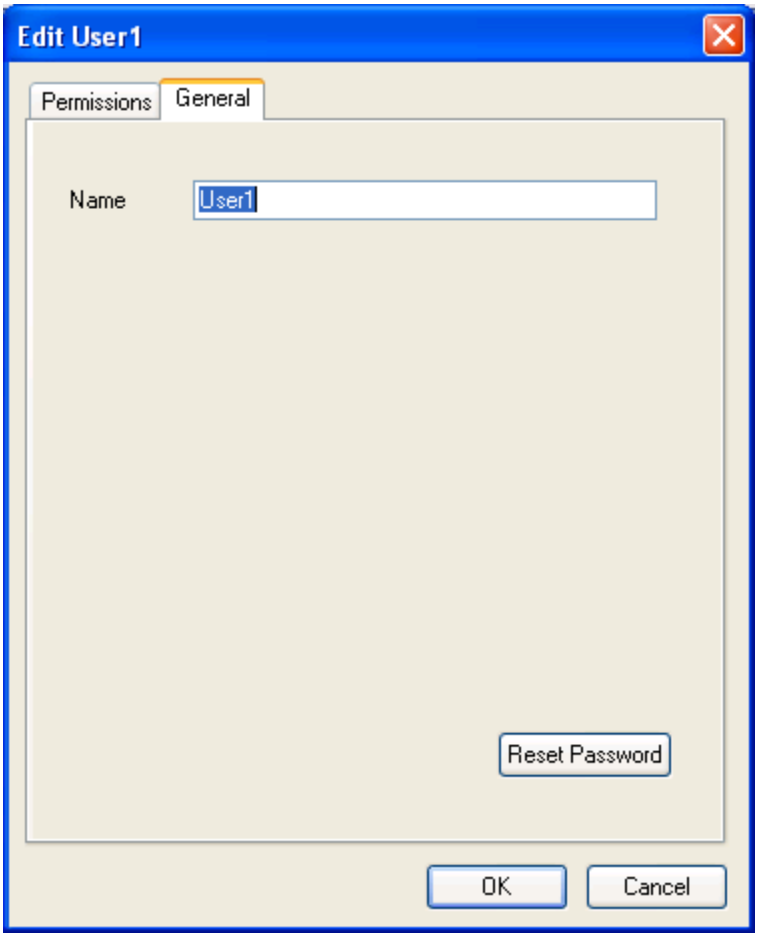


Figure 76 - Edit <Username> Window (General Tab)

To change the user's password:

1. Click on the **Reset Password** button. The **Reset Password** window (Figure 77) appears.

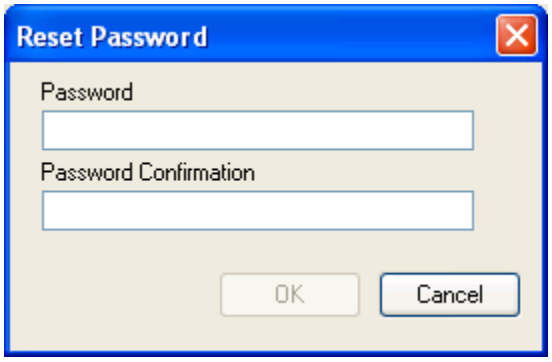


Figure 77 - Reset Password Window

2. Enter the new password in the **Password** field.
3. Re-enter the password in the **Password Confirmation** field.
4. Click on the **OK** button.

Edit Group

To edit a group:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, either:
   - Double-click your mouse on the group you wish to edit, or
   - Right-click your mouse on the required group and select **Edit Group** from the displayed menu, or
   - Click your mouse on the required group and then select the **Edit** button.
2. The **Edit <Groupname>** window (Figure 78) appears.

Notes:

- In this example, the group being edited is ***Group1*** so the window's title reads **Edit Group1**.
- For Private Groups, there are three tabs on this window: **Permissions**, **General**, and **Members**. For NT Groups, there is only the **Permissions** tab.

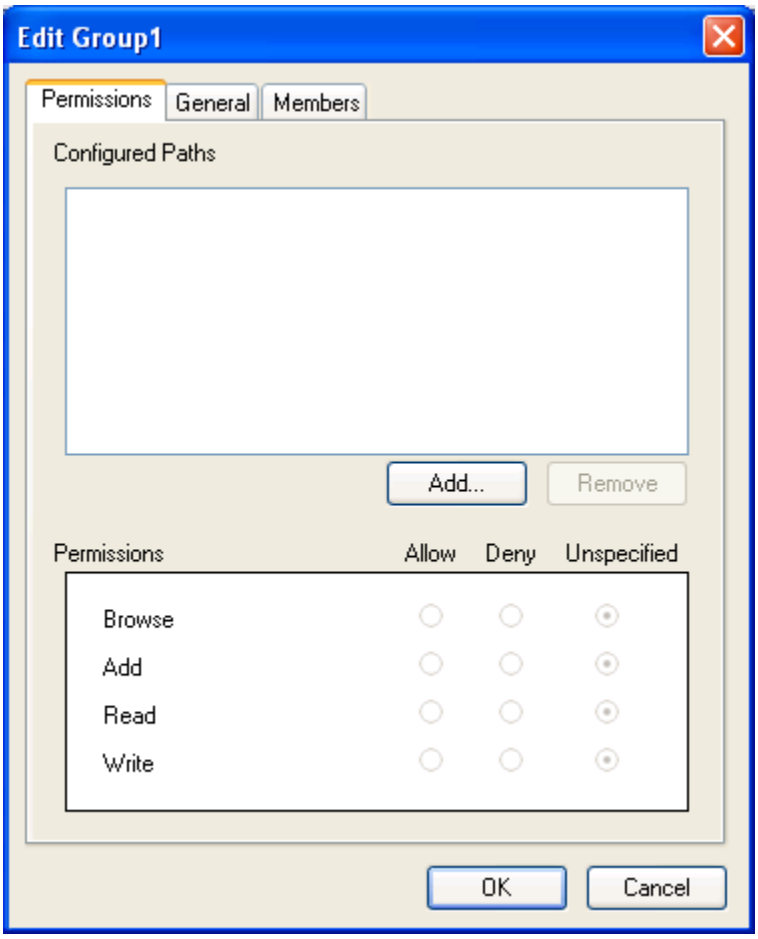


Figure 78 - Edit <Groupname> Window (Permissions Tab)

On the **Permissions** tab, the permissions associated with this group relating to browsing, adding, reading, and writing for any specific paths or OPC items are configured. Refer to the **Edit User** section regarding configuration on the **Permissions** tab. Group and user permissions are configured in the same way.

On the **General** tab (Figure 79) of the **Edit <Groupname>** window, the name of the group can be changed.

To edit a group's name:

1. Enter a new name in the **Name** field.
2. Click on the **OK** button.

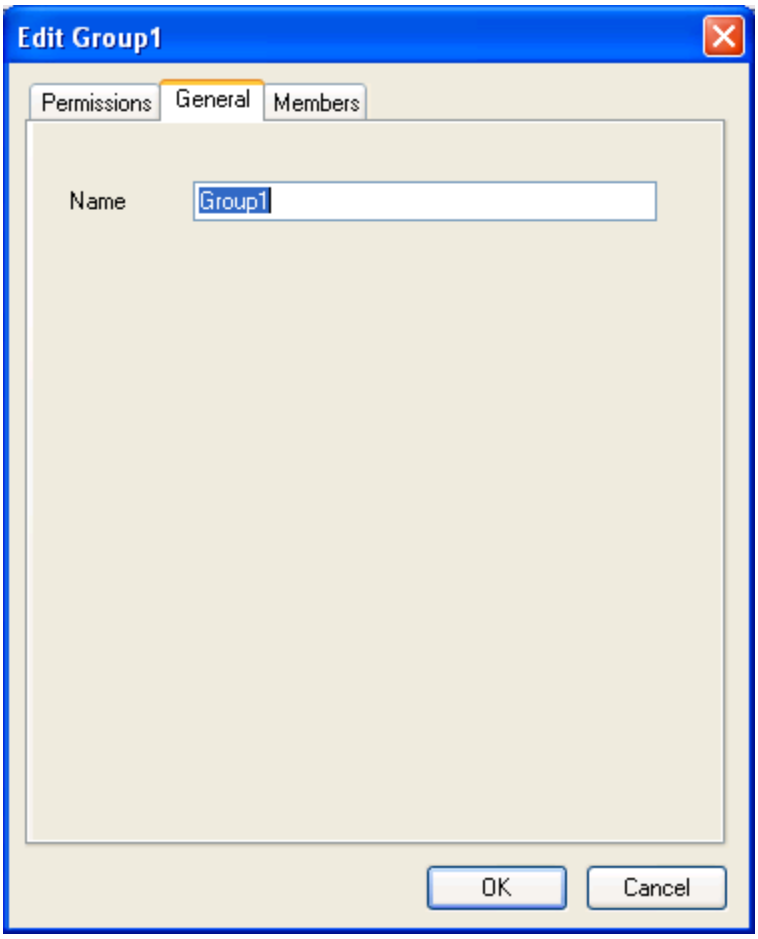


Figure 79 - Edit <Groupname> Window (General Tab)

On the **Members** tab (Figure 80), the users that are part of the current group are defined. By default, the **Group Members** screen section is empty. In Figure 80, ***User1*** has been added to ***Group1***.

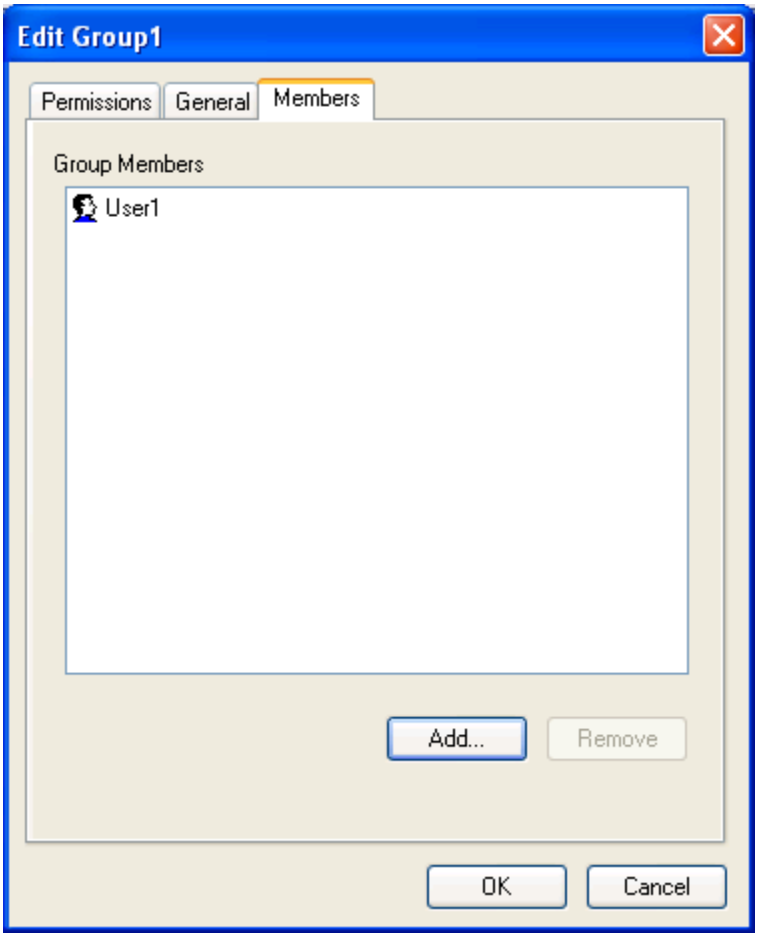


Figure 80 - Edit <Groupname> Window (Members Tab)

To add one or more members to the group:

1. Click on the **Add** button. The **Add Existing User** window (Figure 81) appears.

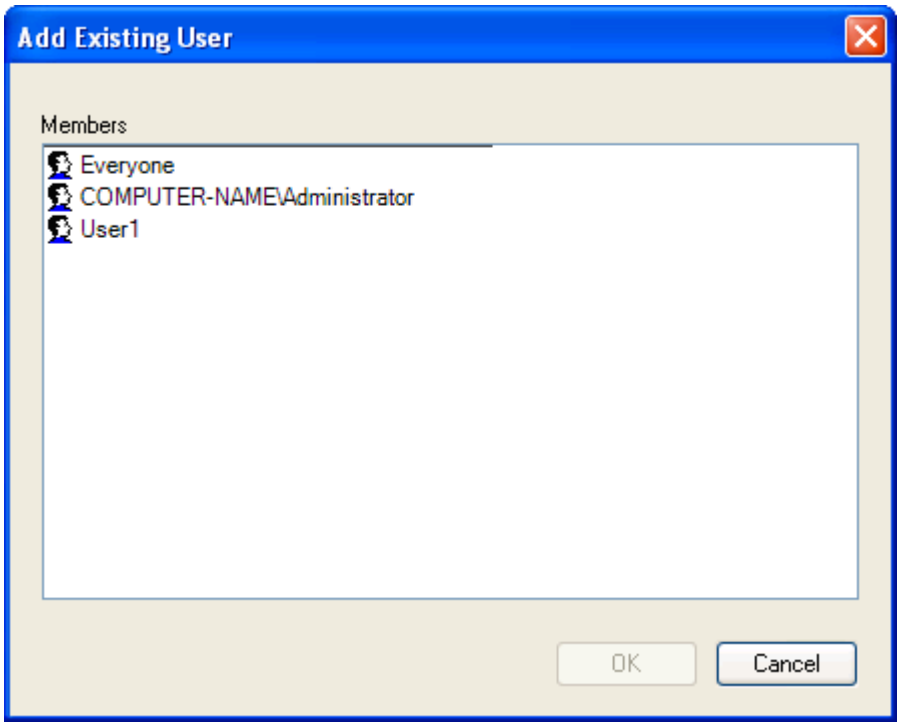


Figure 81 - Add Existing User Window

2. The **Members** screen section lists all the defined users from which you may choose. Select one or more desired users to become members of the group.
3. Click on the **OK** button. The selected user or users appear in the **Group Members** screen section on the **Members** tab (Figure 80).

To remove a user from the group:

1. Select that user in the **Group Members** screen section.
2. Click on the **Remove** button.

   Notes:

   - If a user name and a group name (to which the user belongs) have permissions defined for the same location, the user name will be used. If a user belongs to two groups that have permissions defined for the same location, the group that is alphabetically first will be used.
   - The exact location (full item ID for reading, writing, and adding, or the full path for browsing) is always looked for first when determining the permission for a particular user. If the exact location is not found, or is found but does not contain a defined permission, then the closest match in location is looked at next, and so on. The root location (i.e., empty string) will always match, and is used if no partial location match is made.

Delete

To delete a previously added user or group:

1. From the **MatrikonOPC Tag Security Configuration Utility** main screen, either:
   - Select the required user or group from the list and click on the **Delete** button (Figure 66), or

- For a group, right-click your mouse on the required group and select ***Delete Group*** from the displayed menu, or
- For a user, right-click your mouse on the required user and select ***Delete User*** from the displayed menu

Change Password

To change the security password for the OPC server:

1. From the **MatrikonOPC Security Configuration Utility** main screen, click on the **Change Password** button (Figure 63).
2. The **Change Password** window (Figure 82) appears.

Figure 82 - Change Password Window

3. Enter the old password for the current OPC server.
4. Enter and confirm the desired new password.
5. Click on the **OK** button.

   Note: Changes are applied in the Utility main window by clicking on either the **OK** or **Apply** button.

Examples

This section contains examples of tag security settings made via the MatrikonOPC Tag Security Configuration Utility, and the impact of these settings.

Default

The default tag security configuration provided is quite simplistic as is shown in Table 63:

User/Group	Configured Paths	Browse	Add	Read	Write
Everyone	\	Allow	Allow	Allow	Allow

Table 63 - Default Tag Security

As no users or groups are defined, any NT user connecting to the OPC server will use the ***Everyone*** default user for permissions. Only the root level location is defined under the **Configured Paths** list, which gives full permissions to the entire OPC server. An administrator will want to change these default settings for the OPC server to provide more security on its tags.

Example 1

Table 64 contains a possible tag security configuration enforces complete restricted access except for the NT user ***DOMAIN\Admin***:

User/Group	Configured Paths	Browse	Add	Read	Write
Everyone	\	Deny	Deny	Deny	Deny
DOMAIN\Admin	\	Allow	Allow	Allow	Allow

Table 64 - Example 1

At this point, more users can be added which belong to a more restricted group. This group may be denied write capability or restricted to certain areas in the server hierarchy.

Example 2

Following is an example of a tag security configuration for the MatrikonOPC Server for Simulation and Testing, with sample use-cases. Table 65 lists all defined users and groups and their associated permissions. Table 66 indicates which users are members of the defined groups. ***Unspec.*** permission indicates *Unspecified*. Groups in this example are Private Groups.

User/Group	Configured Paths	Browse	Add	Read	Write
Everyone	\	Allow	Deny	Deny	Deny
Everyone	Configured Aliases	Deny	Unspec.	Unspec.	Unspec.
DOMAIN\john.doe					
DOMAIN\fred.stone					
DOMAIN\sally.simpson					
DOMAIN\the.admin					
Supervisor	\	Deny	Allow	Allow	Deny
Restricted (group)					
Common (group)	\	Allow	Allow	Allow	Deny
Common (group)	Triangle Waves	Deny	Deny	Deny	Deny
Admin (group)	\	Allow	Allow	Allow	Allow
Admin (group)	Bucket Brigade.Int4	Unspec.	Unspec.	Unspec.	Deny

Table 65 - Example 2: Permissions

User/Group	Configured Paths
Restricted	DOMAIN\john.doe
Common	DOMAIN\fred.stone, DOMAIN\sally.simpson
Admin	DOMAIN\the.admin

Table 66 - Example 2: Groups and Members

Tip 1

Adding a user name and not defining any permissions for that user name (or any group it belongs to), will completely restrict their access to information in the OPC server.

Case 1

User ***DOMAIN\john.doe*** connects. The user is unable to perform any function at any level, since no permissions have been defined for this user name or the **Restricted** group to which it belongs.

Tip 2

Use the root location to define permissions for locations that have not been defined. Use the default user to define permissions for users that are not listed.

Case 2

User ***DOMAIN\no.name*** connects. This name is not defined so the default user permissions are used. At the root level location, the default user can browse but no more. The only other location where permissions are defined for the default user is ***Configured Aliases*** – here browsing is denied and adding, reading, and writing permissions are not defined. Thus, ***DOMAIN\no.name*** will be able to browse the entire server except the ***Configured Aliases*** branch and all aliases under it. Also, ***DOMAIN\no.name*** will not be able to do any item addition, reading, or writing. Since item addition, reading, and writing permissions were not defined for the default user for under ***Configured Aliases***, it would use the permissions defined at the root level.

Case 3

User ***DOMAIN\fred.stone*** connects. This user belongs to the defined group ***Common***. At the root level, the group Common can browse, add, and read items, but writing is denied. The only other location which defines permissions for this group is ***Triangle Waves***. This location has denied all permissions for the ***Common*** group.

Case 4

User ***Supervisor*** connects through the private security interface and logs on. This user does not belong to any group, but its name has a root level permission which allows it to add and read items. No other locations have permissions defined. Since the user cannot browse, it will need to know the item syntax for any items it wishes to add and read.

Case 5

User ***DOMAIN\the.admin*** connects. This user belongs to the ***Admin*** group, which at the root level has all permissions allowed. The only other location where the ***Admin*** group has permissions defined, is the full item ID ***Bucket Brigade.Int4***. This user is unable to write to that item. However, the root permissions for browsing, adding, and reading the item apply as they are not defined at this level.

Security Details

ACL File

The permissions that the **MatrikonOPC Tag Security Configuration Utility** creates are stored in a file named **ACL.sec** that is located in the same directory as the MatrikonOPC server executable: **C:\Program Files\Matrikon\OPC\< OPC Server Name >**.

ACL.sec is encrypted with 256-bit AES using the password key that was provided during installation (or changed in the security utility).

While the server is running, if the **ACL.sec** file is moved or deleted from its location above, security functions as it did before the file was moved. If the server is re-started with the **ACL.sec** file missing, no users will have permissions to do anything (i.e., browsing, adding, reading, writing) and must re-configure their security settings.

Note: It is advised that this file is backed up and stored somewhere safe, once tag security configuration has been completed. In the event that this file is lost or deleted by accident, simply

transfer the stored backup (if one was made) to the directory stated above. If **ACL.sec** becomes deleted and no backup was made, tag security will need to be re-configured.

Run-Time Tag Security Configuration

Tag security can be configured and updated while a MatrikonOPC server is running. This run-time security configuration can be controlled using the **#MonitorACLFile** OPC item (found in the root branch when browsing the server).

The **#MonitorACLFile** is a boolean OPC tag that has a default value of ***True*** when the server starts which indicates that the tag security permissions are monitored and applied while the server is running. By adding this item to an OPC group and writing a value of ***False*** to this item, the server no longer applies tag security configuration changes while the server is running. The server would need to be stopped and restarted for the changes to take effect.

By disabling the ability of run-time tag security configuration, a malicious user that has access to the machine cannot delete the ACL file and thus remove all user permissions for OPC security. Run-time tag security configuration can be re-enabled when changes to the tag security need to be made, and then disabled again once they have been applied.

Security should be applied to the **#MonitorACLFile** tag so that unauthorized users are unable to control this ability described above.

Client Connections and Private Log On

When an OPC client connects to the server, the OPC Framework impersonates the client and determines the name of the NT user that is running the client. This client name (consisting of the domain name and the user for that domain) is what the OPC Framework uses when checking permissions set in the **MatrikonOPC Tag Security Configuration Utility**. If this NT user was not added in the **Users & Groups** list (Figure 66), then the user ***Everyone*** is used instead.

If the OPC client chooses to use the *IOPCSecurityPrivate* interface, it must call the *Logon()* function and provide a name and password. The OPC Framework will immediately look for the private user in the **Users & Groups** list (Figure 66). If the user is found and the password provided is correct, the *Logon()* result is successful and the user name provided replaces the NT user that was obtained through impersonation. If not found, the *Logon()* result is failure and the NT user remains as the name used for security permissions. When the OPC client calls *Logoff()*, the NT user becomes the name used for security permissions.

Determining Permission

Whenever a browsing, item addition, read, or write operation is performed by the OPC client, the OPC Framework will call a permission routine to determine if permission is allowed or denied. Three pieces of information are always sent to this routine call:

1. Identity
2. Location
3. Type

Identity is the user name described in the previous section – either the NT domain name, a private name, or the default user ***Everyone***. In addition, identity includes any groups (private or NT) that the user belongs to. Location is either the full item ID or a partial path in the server hierarchy. Type is the access type: browsing, item addition, reading, or writing. Each piece of information is checked in the order listed above.

The following information details the procedure used in determining permission rights:

1. Look for a permission match with the NT domain name or private name. This step occurs only if the NT domain name or private name has been defined in the **Users & Groups** list.
2. If a permission match was not yet found, look for one using each group defined in the **Users & Groups** list that the NT domain name or private name belongs to. The first group that contains a permission match will be the one used. An NT domain name does not need to be defined in the **Users & Groups** list for a group it belongs to, to have a permission match. The NT domain name must be part of **Group Members** for private groups, or in the Active Directory for NT groups.
3. If a permission match was not yet found, look for a permission match using the default user ***Everyone***.

The procedure used in finding a permission match for a particular identity is as follows:

1. Check whether the exact location is found in the **Configured Paths** list for the **identity**. If found:
    a. Check whether the permission type (***browse***, ***add***, ***read***, or ***write***) is specified (Figure 75). If the permission type is specified for the action being performed, a match is found and the permission (either allow or deny) is used.
2. If a permission match was not found in step 1a, check whether a partial match of the location is found in the **Configured Paths** list for the **identity**. A partial match is always a sub-string of the given location that starts with the same characters. This represents starting from the root location and moving down the browsing hierarchy.

    For example, the location **Random** is a partial match of the full item ID **Random.String**. However, ***String*** is not a partial match (even if it is a sub-string) because it does not match the first seven characters. The root location **** is considered a partial match for every full item ID.

    For each partial match from step 2, apply step 1a. The best partial match (i.e., the one that has the most characters) that is successful in step 1a is the one whose permission is used.

Example

The following is an example of a tag security configuration. Table 67 lists all defined users and their associated permissions.

User/Group	Configured Paths	Browse	Add	Read	Write
Everyone					
DOMAIN\john.doe	\	Allow	Allow	Allow	Allow
DOMAIN\john.doe	Random	Allow	Deny	Deny	Deny
DOMAIN\john.doe	Random.Int	Allow	Allow	Allow	Allow

Table 67 - Example 2: Permissions

The user ***DOMAIN\john.doe*** wants to read the item ***Random.Int4***. The permission routine would not be able to find an exact location match under the **Configured Paths** list. However, it would loop through each partial match in the Configured Paths list looking for a defined permission. The location ***Random.Int*** would be the closest match and the permission would be ***ALLOW***.

The user ***DOMAIN\john.doe*** wants to read the item ***Random.Real4***. The permission routine would not be able to find an exact location match under the **Configured Paths** list. However, it would loop through each partial match in the Configured Paths list looking for a defined permission. The location ***Random*** would be the closest match and the permission would be ***DENY***.

Appendix H Offline Mode

Introduction

Offline mode allows the MatrikonOPC Server Framework to be disconnected to the end device, but still return data as if the date was coming from it. This is useful in two situations:

1. A configuration can be set up with to interact with any hardware, allowing any problems in the configuration to be worked out without having them affect the device.
2. Diagnosing problems becomes much simpler when an implementer can remove the device hardware from the architecture.

Offline mode can be applied to the entire configuration of a MatrikonOPC server or to a portion of it.

Offline Mode Toggling

A Boolean control item, named **#OfflineMode**, is provided at each branch in a MatrikonOPC server browsing hierarchy. Adding this item will allow an OPC client to monitor the offline mode for that location using read operations and provide the ability to toggle offline mode for that location using write operations.

When reading the item **#OfflineMode**, the value ***True*** represents enabled and the value ***False*** represents disabled. Likewise, writing a value of ***True*** to **#OfflineMode** enables offline mode and writing a value of ***False*** to **#OfflineMode** disables offline mode. When offline mode is toggled at a particular location in the hierarchy, it also toggles offline mode for all branches underneath it in hierarchy. For example, if offline mode was enabled at the root in the hierarchy, every branch would have its offline mode enabled. Upon server start-up, every branch has the **#OfflineMode** item set to ***False*** (i.e., offline mode is disabled).

Offline Mode Item Values

When offline mode is enabled at a particular location in the browsing hierarchy, all OPC items at that location generate random values according to their data type (the ***Date*** data type is the one exception to this). Table 68 shows the value behaviour for each data type:

Data Type	Behaviour
Boolean (VT_BOOL)	***True*** or ***False***.
Character (VT_I1)	Value between ***-128*** and ***127***.
16-bit Integer (VT_I2)	Value between ***-32768*** and ***32767***.
32-bit Integer (VT_I4 and VT_INT)	Value between -***2147483648*** and ***2147483647***.
Byte (VT_UI1)	Value between ***0*** and ***255***.
Word (VT_UI2)	Value between ***0*** and ***65525***.
Double Word (VT_UI4 and VT_UINT)	Value between *0* and ***4294967295***.
Float (VT_R4)	Value between ***-13388.5962*** and ***13388.5962***.
Double (VT_R8)	Value between ***-8345.512588035*** and ***8345.512588035***.
Date (VT_DATE)	Current system time.

Data Type	Behaviour
String (VT_BSTR)	***Simulated*** or ***Data***.

Table 68 - Offline Mode Item Values

Operational Effects

When offline mode is enabled at a particular location in a server's hierarchy, there is no communication between the server and the end-device/application to which the location refers.

Note: Some servers allow communication to multiple devices/applications. To ensure there is no communication in the entire server, offline mode should be enabled at the root level.

To ensure that there is no communication when offline mode is enabled, the MatrikonOPC Server Framework will disable the object associated with that branch. When an object is disabled due to offline mode, the **Enabled** checkbox will be cleared (i.e., unselected) in its panel within the configuration utility. Enabling the object will have no effect while the object has offline mode enabled. When offline mode is disabled for that particular object, the MatrikonOPC Server Framework will re-enable it and the **Enabled** checkbox will be selected (i.e., checked) once again.

Appendix I Installation

Once the system requirements have been met, you are ready to install the software.

Note: As part of the installation process, the **MatrikonOPC Analyzer** tool is installed and used to detect the system settings that affect the use of this software. No information is communicated back to Matrikon. Information is stored on this system **only** for future use by MatrikonOPC Support to assist with troubleshooting, if required.

To install the software:

1. Insert the MatrikonOPC Server for SCADA Modbus CD into the CD drive.
2. If the MatrikonOPC InstallAware screen does not automatically appear, double-click the installation *.exe* file. The **MatrikonOPC Server – InstallAware Wizard** verifies its contents (Figure 83).

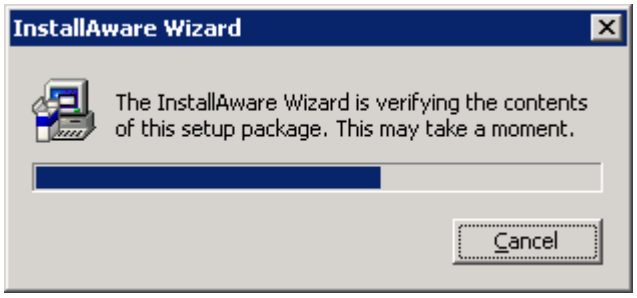


Figure 83 - InstallAware Wizard Verification Window

3. Either a **Pre-Requisites** screen (Figure 84) appears, or you are taken directly to the **Licensing Agreement** (Figure 85) screen. If the **Licensing Agreement** screen is the displayed screen, go to step 5.
4. If the **Pre-Requisites** screen appears, then click on the **Next** button to install the listed pre-requisites.

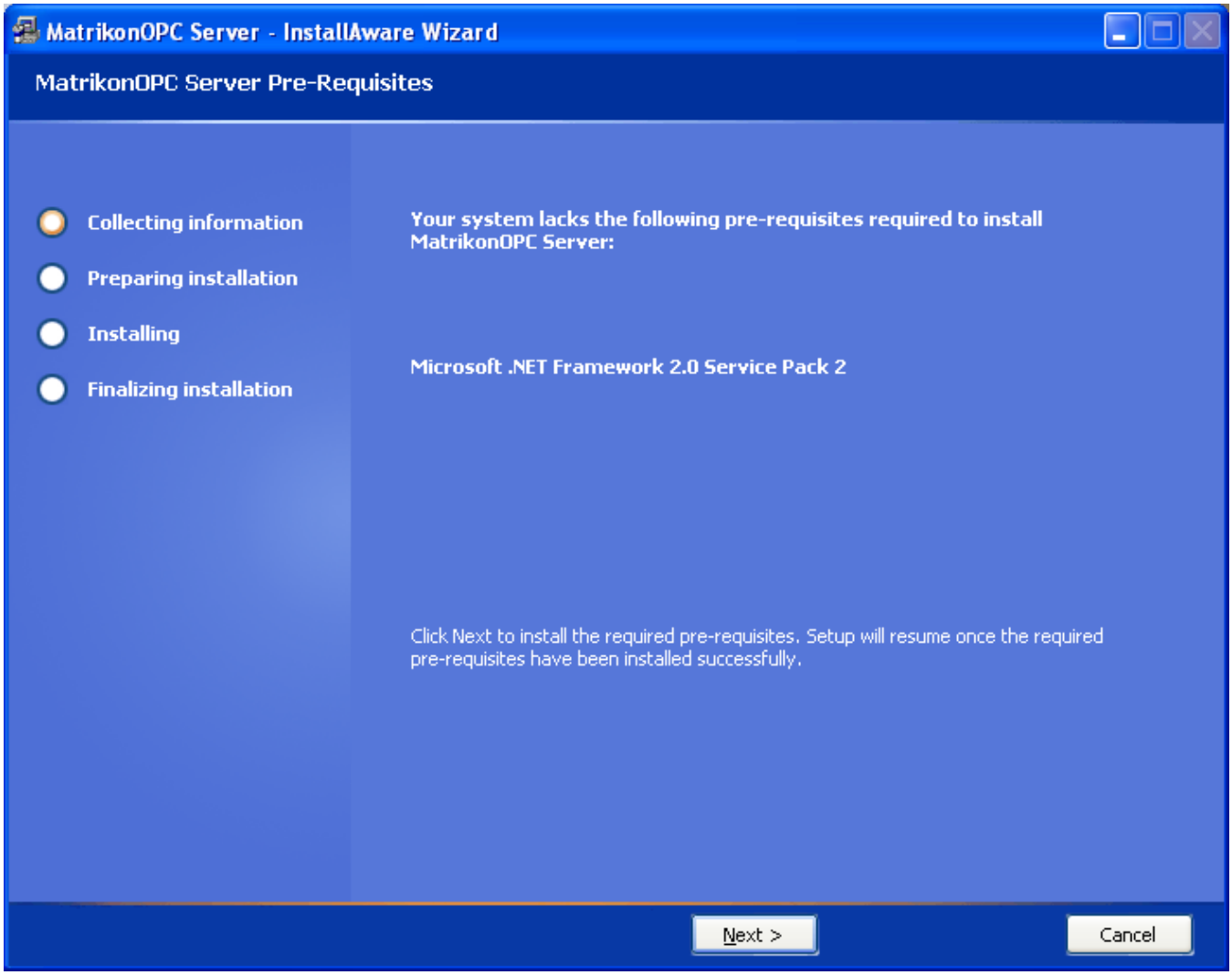


Figure 84 - Pre-Requisites Screen

5. After all pre-requisites have been installed, the **License Agreement** screen (Figure 85) appears.

   Note: From the **License Agreement** screen, you have the option of selecting the **I reject the license agreement** option. Selecting the **I reject the license agreement** option button disables the **Next** button so your options are to return to the previous screen, cancel the install by clicking on the **Cancel** button, or select the **I accept the license agreement** option button enabling you to proceed through the install.

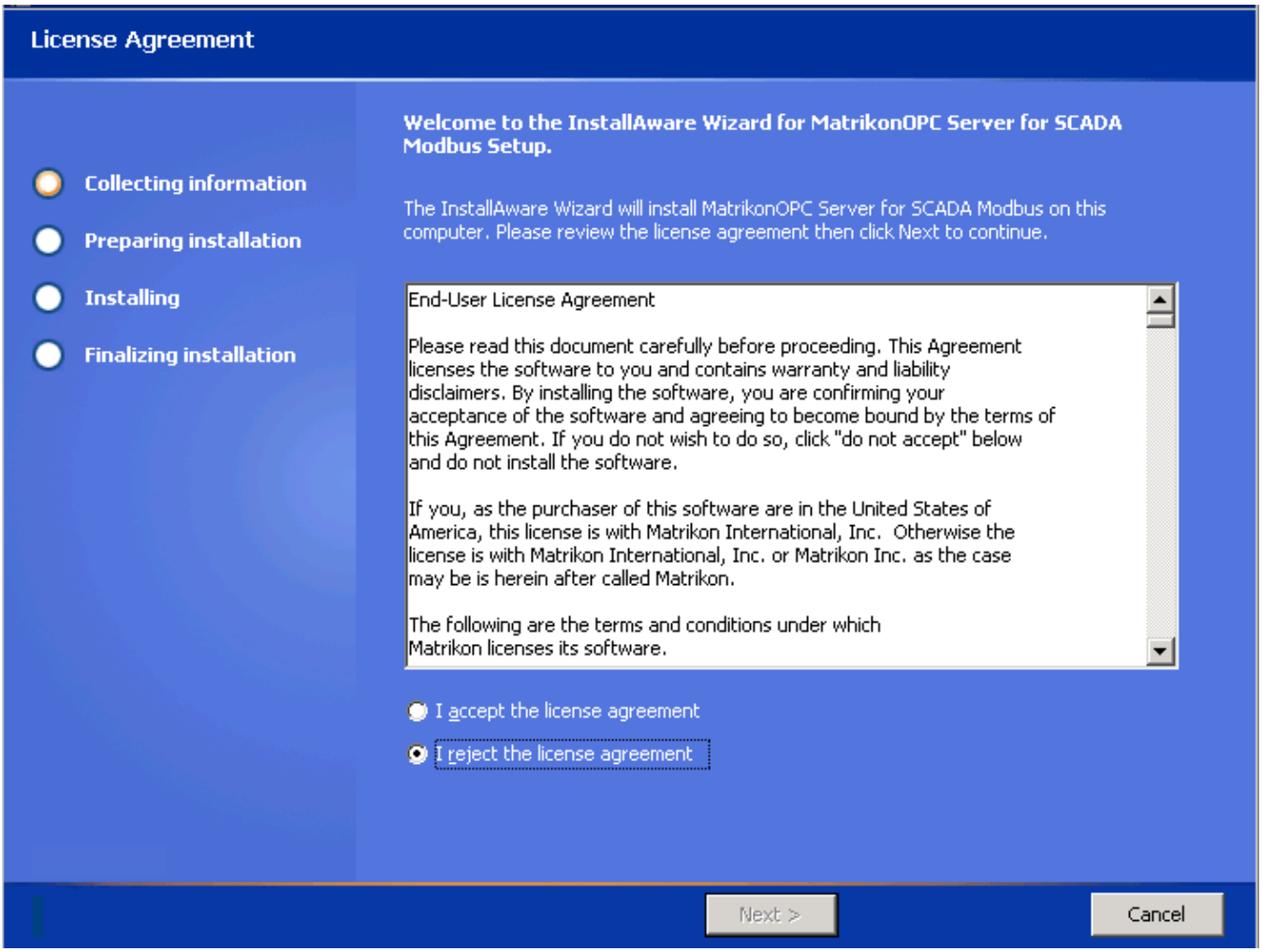


Figure 85 - License Agreement Screen

6. Read the **Software License Agreement**, using the scroll bar to view the entire message.
7. Select the **I accept the license agreement** option button.
8. Click on the **Next** button. The **Setup Type** screen (Figure 86) appears.

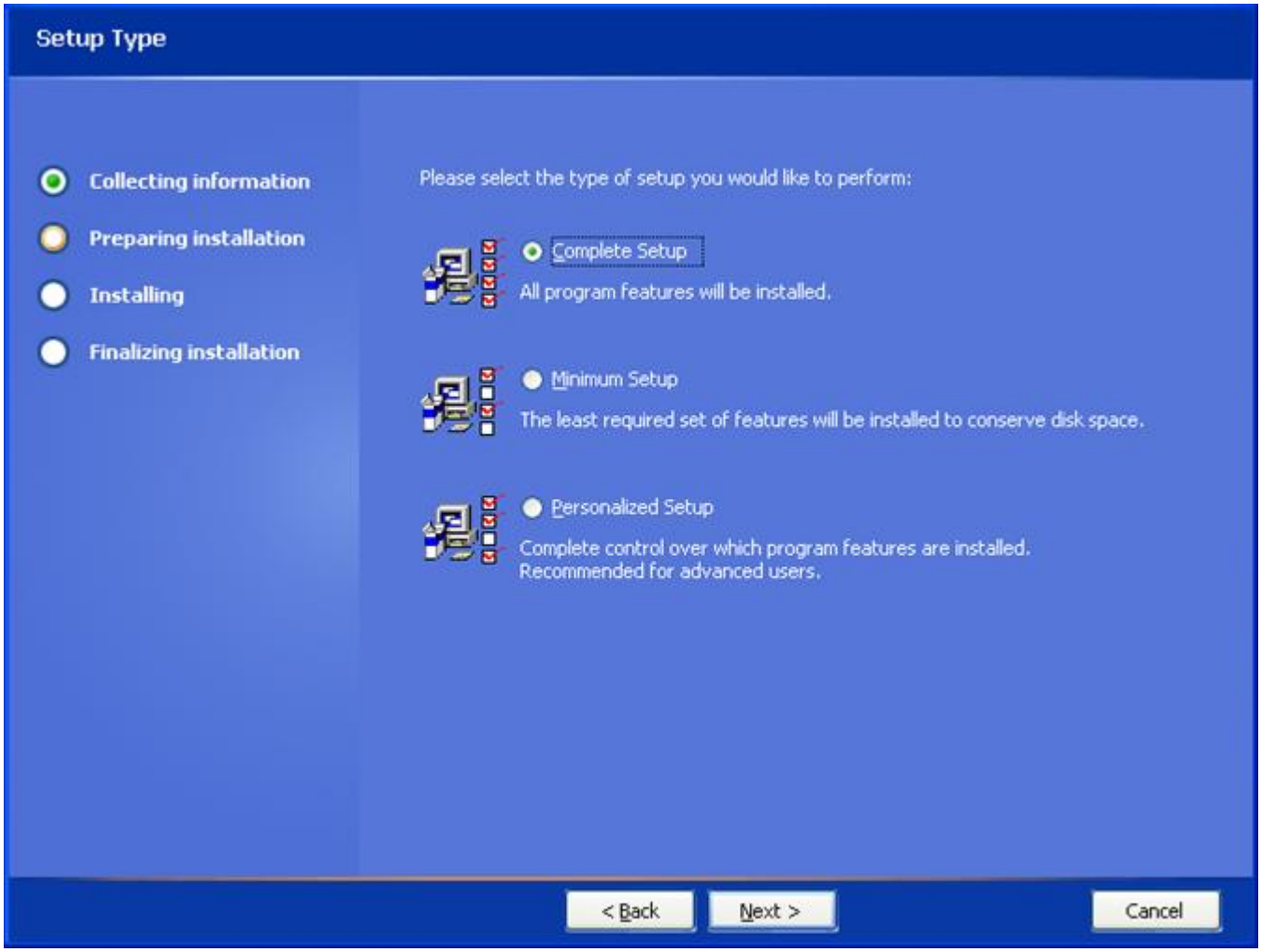


Figure 86 - Setup Type Screen

9. Select the type of setup to be performed.

   Note: MatrikonOPC recommends that you select the **Complete Setup** option.

   If you want to install multiple instances, then you must select the **Personalized Setup** option and click **Next** to proceed. In the **Select Features** page, you can mention the number of instances that you want to install in the **Number of server installs (1 to 5)** box. Only a maximum of 5 instances are supported.

10. Click on the **Next** button. The **Destination Folder** screen (Figure 87) appears.

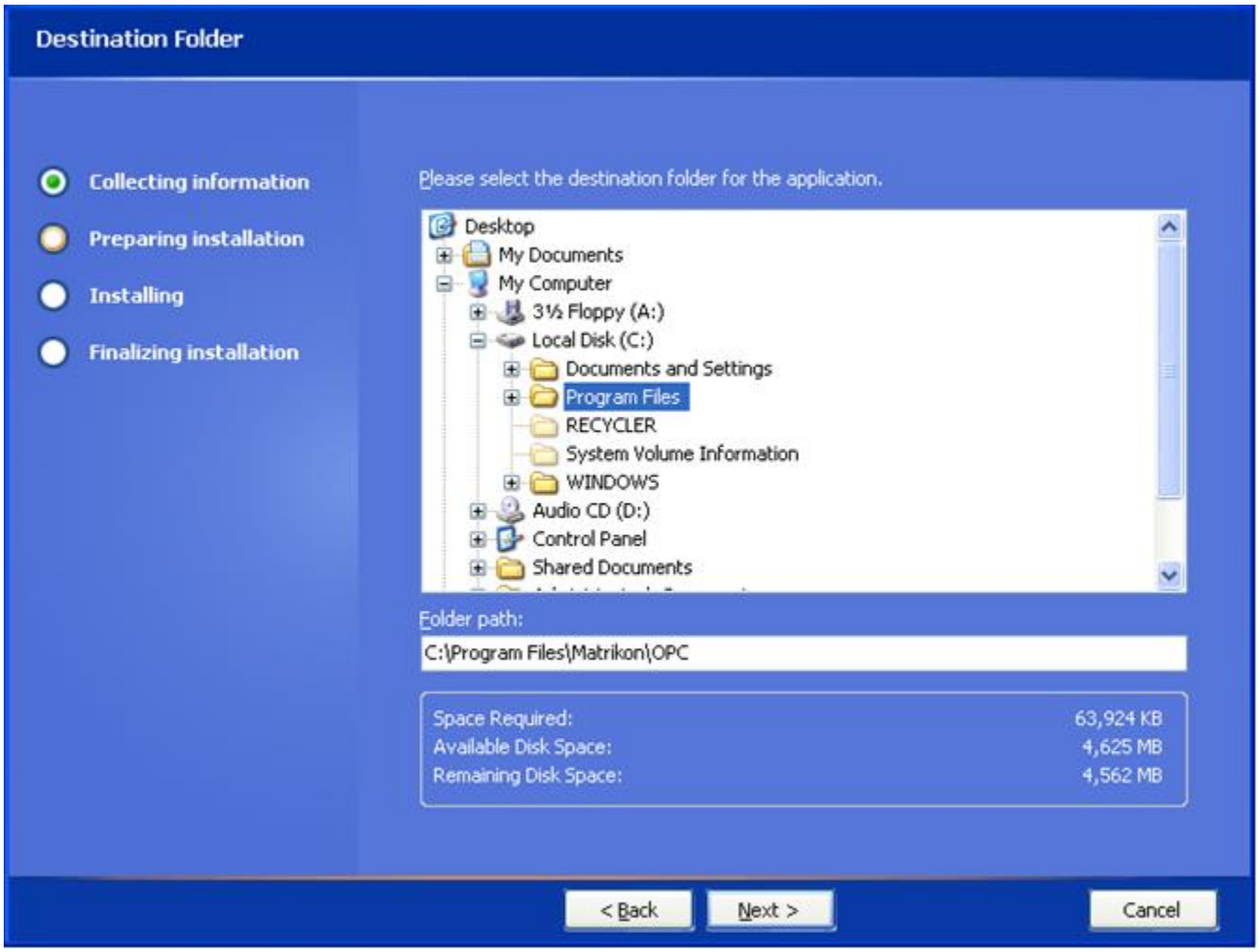


Figure 87 - Destination Folder Screen

11. Select the folder in which to install the MatrikonOPC server, or accept the default location displayed in the **Folder path** field.
12. Click on the **Next** button. The **Start Menu** screen (Figure 88) appears.

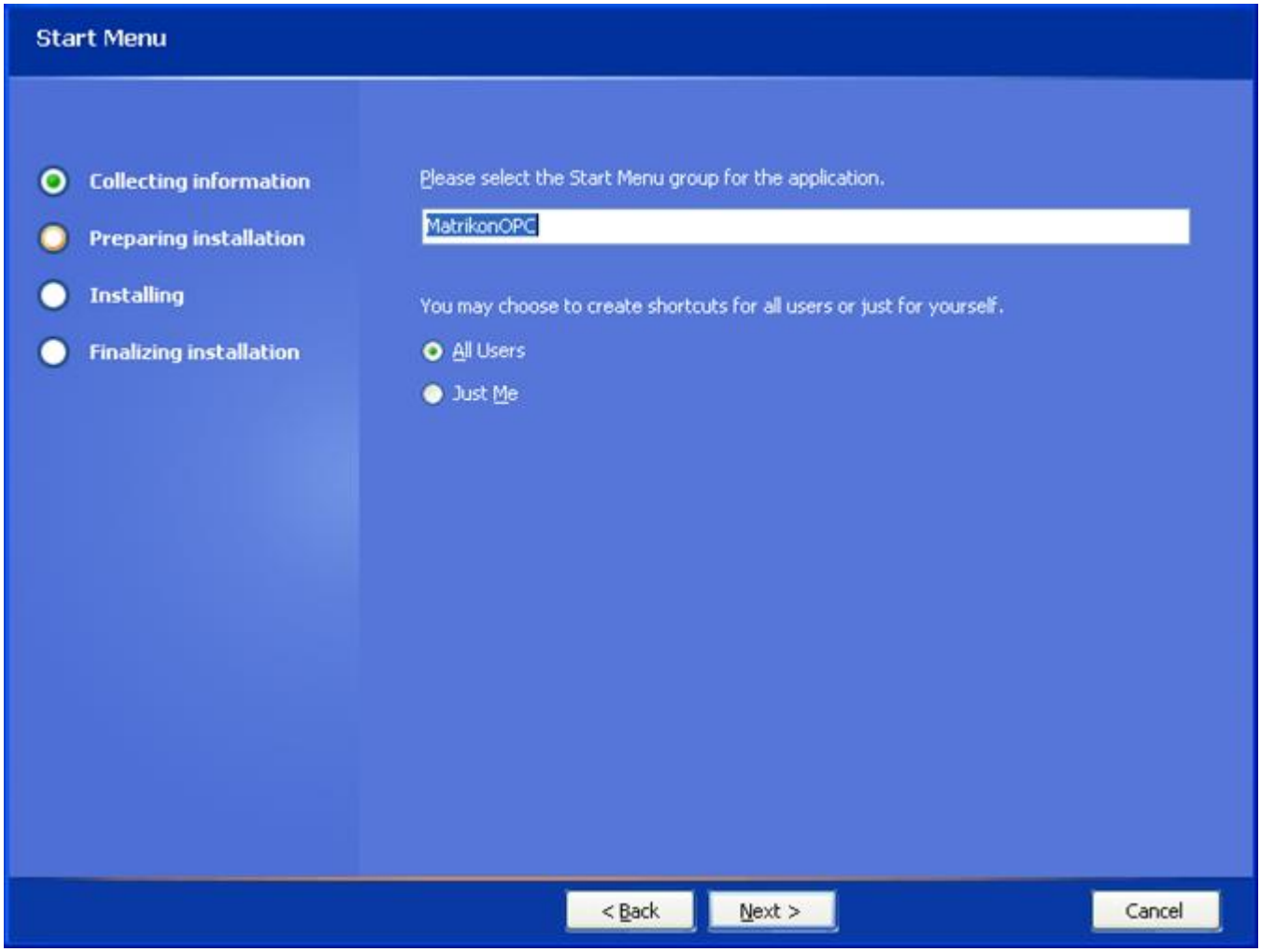


Figure 88 - Start Menu Screen

13. Select the **Start Menu** group and specify whether you want shortcuts created only for yourself, or for all users, by selecting the applicable option button.
14. Click on the **Next** button. The **Enter Password** screen (Figure 89) appears.

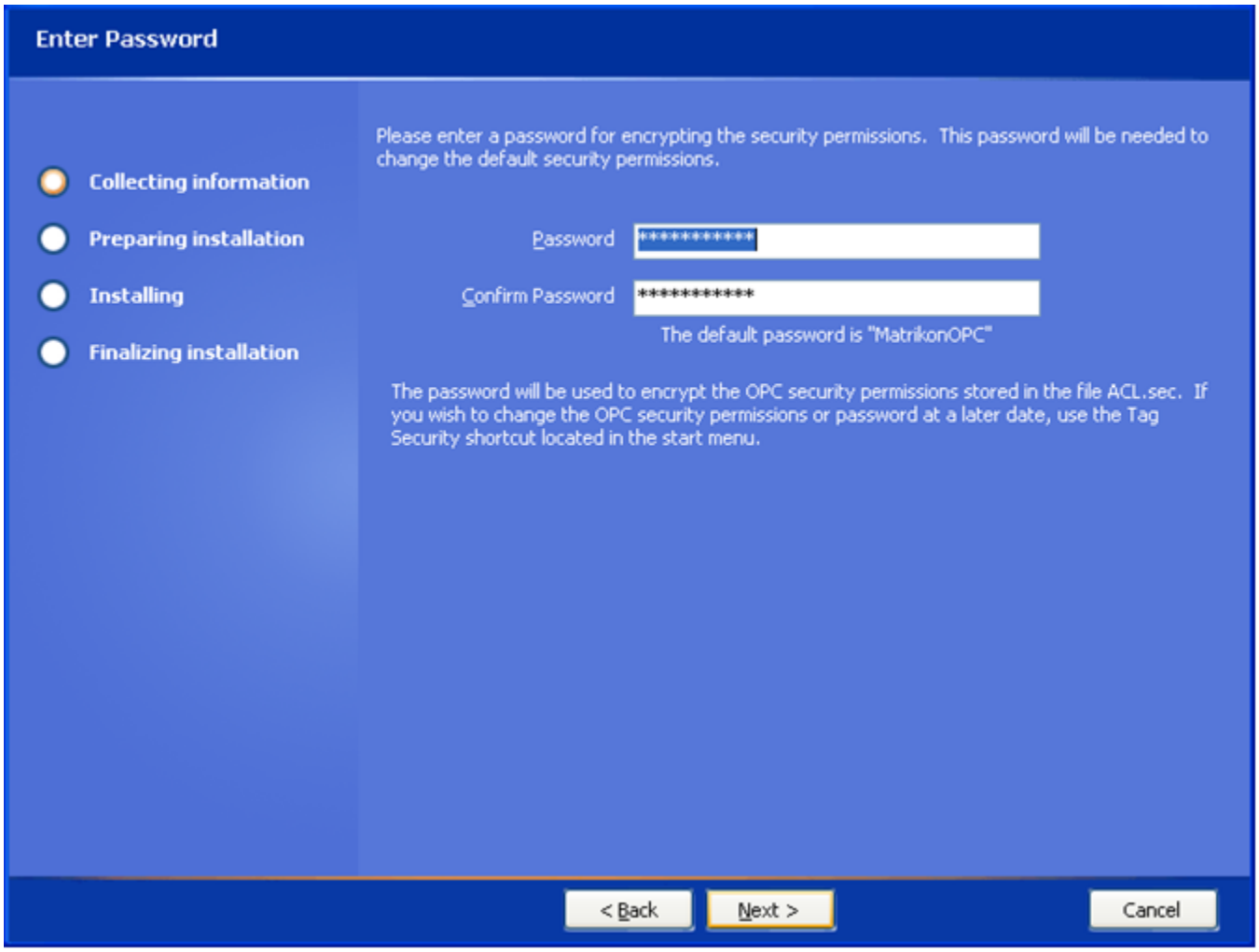


Figure 89 - Enter Password Screen

15. A default password is supplied for you in the **Password** and **Confirm Password** fields.

Notes:

- The default password provided for you is ***MatrikonOPC***. Note this password for future reference.
- If you need to change the password at a later date, access the **Tag Security** shortcut from the **Start** menu. Click on the Windows **Start** button. Select **Programs** -> **MatrikonOPC** -> ***<OPC Server Name>*** -> **Tag Security**. For more information, refer to **Appendix G – Security**.

16. Click on the **Next** button to accept the default password. The **Licensing** screen (Figure 90) appears.

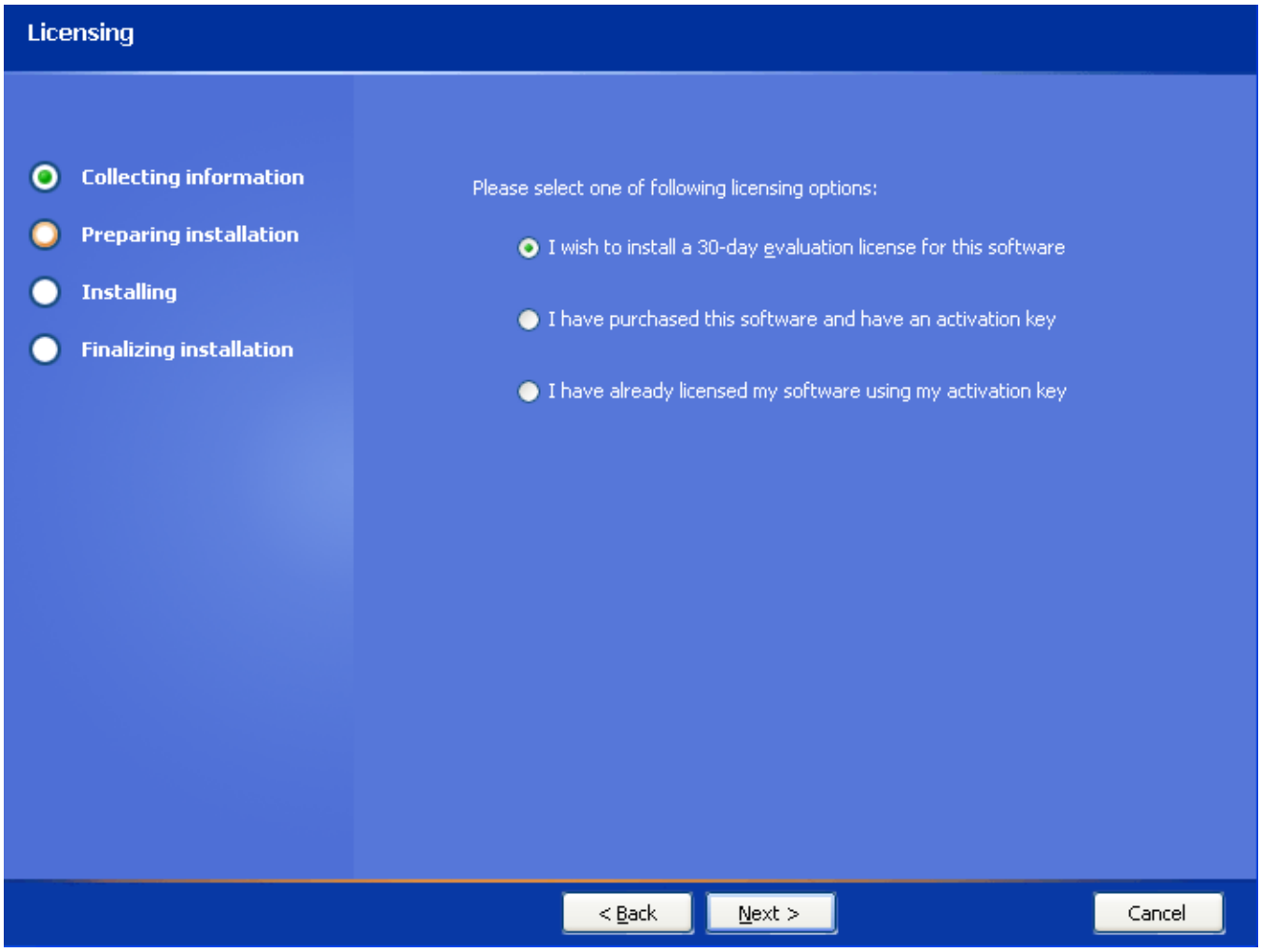


Figure 90 - Licensing Screen

17. Select the applicable licensing option.
18. Click on the **Next** button. The **Ready to Install** screen (Figure 91) appears.

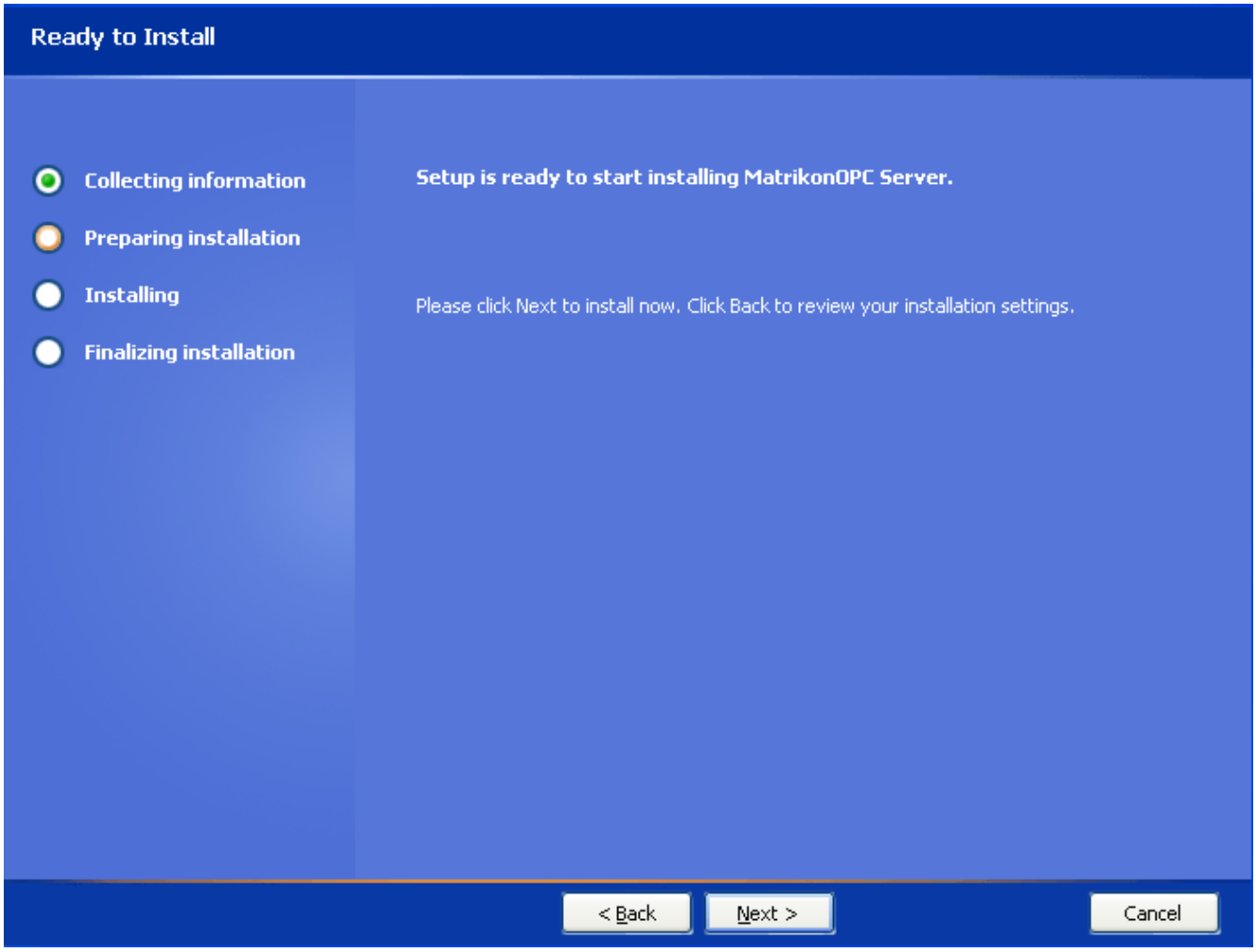


Figure 91 - Ready to Install Screen

19. Click on the **Next** button. The **Installing MatrikonOPC Server for SCADA Modbus** screen (Figure 92) appears, installation begins, and the product files are copied to the computer.

    Note: Prior to starting the installation, you have the option of clicking on the **Back** button to change any of the installation information. Click on the **Cancel** button if you wish to stop or cancel the installation.

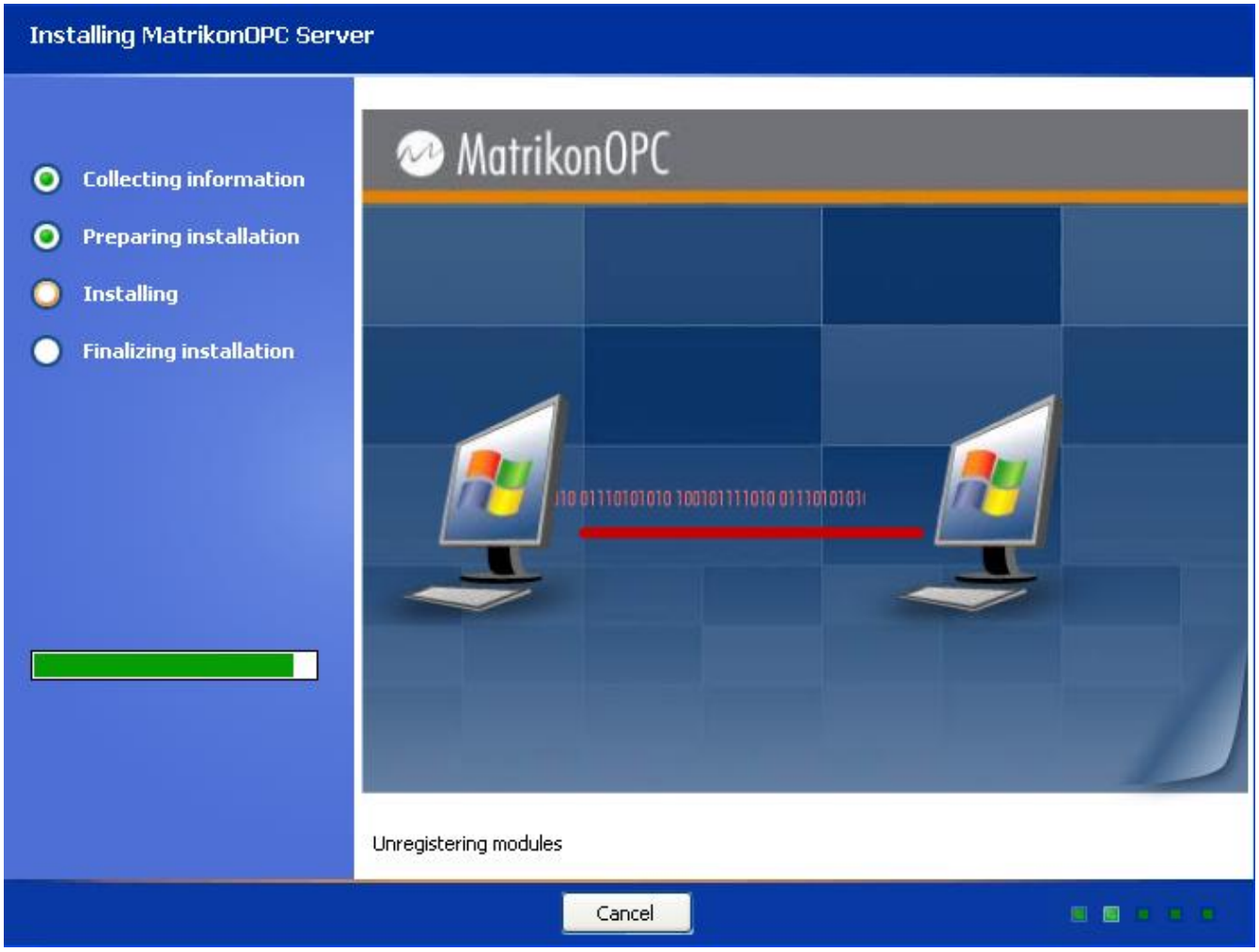


Figure 92 - Installing MatrikonOPC Server Screen

20. When the installation has finished, the **MatrikonOPC Server for SCADA Modbus Setup Complete** screen (Figure 93) appears stating that the MatrikonOPC server has been successfully installed.

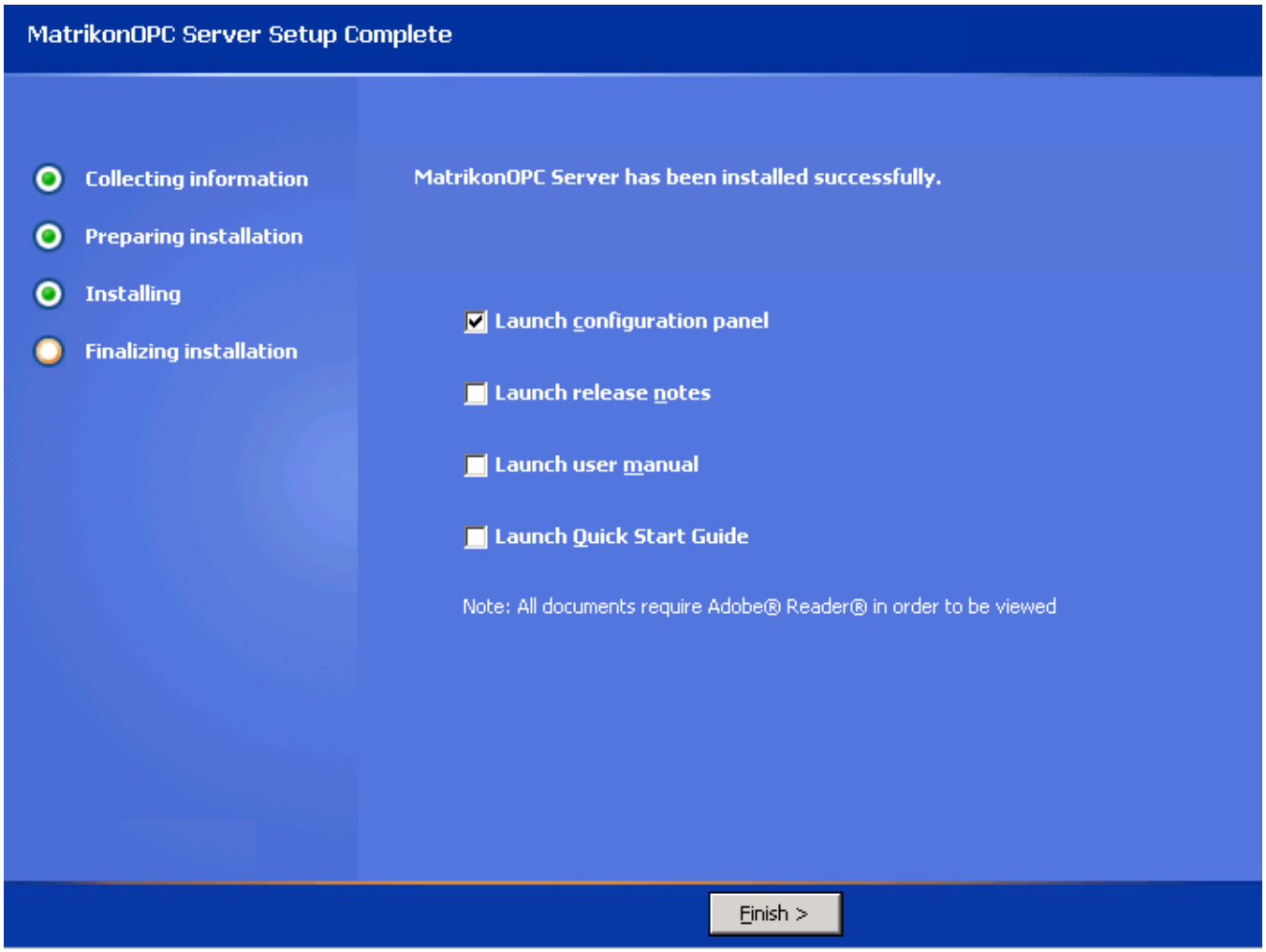


Figure 93 - MatrikonOPC Server Setup Complete Screen

21. At this point, you have the option of launching any or all of the following by selecting the necessary checkbox or checkboxes:
    - Configuration panel
    - Release Notes
    - User's Manual
22. Click on the **Finish** button to complete the installation and exit the Wizard.
23. The necessary files have now been copied to the target computer, the software components are registered, and shortcut icons are created in the **Start** menu.

Note: At this point, it is recommended that you verify the DCOM settings. Reference to the DCOM configuration can be found in the **DCOM Manual**. This configuration varies for different operating systems.

Appendix J Installed Files

The installation program copies all necessary files to the target computer and creates shortcut icons in the **Start** menu.

The files listed in Table 69 are installed by default, in the following location:

C:\Program Files\Matrikon\OPC\SCADA Modbus

File Name	Description
ACL.sec	Access control list containing users and permissions configured for the server.
ACLSecure.exe	Command-line security utility used to encrypt/decrypt the ACL file.
Altova.dll AltovaXML.dll CalcEngine.dll CalcEngineResource.dll CalcEngineServerPS.dll CalcEngineWrapper.dll CalculationGraph.dll clipsdll.dll icudt40.dll icuin40.dll icuuc40.dll log4cxx.dll lsapiw32.dll NetPortalQueryServerPS.dll ProcessNetDataDelivery.dll RmsApiProxy.dll Scheduler.dll ScriptLibrary.dll Xerces-c_2.8.dll	Calculation Engine binary files.
Licensing Procedures.pdf	*Licensing Procedures* document.
MatrikonOPC Server for SCADA Modbus Release Notes.pdf	*Release Notes* for this product.
MatrikonOPC Server for SCADA Modbus User Manual.pdf	This *User's Manual*.
OPCScadaModbus.exe	MatrikonOPC SCADA Modbus executable.
ProgID.txt	Security information file used by the encryption utility.
PSTCFGScadaModbusLib.ocx	ActiveX configuration panels.
Dbghelp.dll	A Microsoft component to assist with debug information output.

Table 69 - Files Installed in "SCADA Modbus" Folder

The files listed in Table 70 are installed by default, in the following location:

C:\Program Files\Common Files\MatrikonOPC\Common

File Name	Description

File Name	Description
OPCAnalyzer Logs\	Logs generated by MatrikonOPC Analyzer.
ACLConfig.exe	MatrikonOPC Tag Security Configuration Utility that configures tag-level security in MatrikonOPC servers.
EULA.pdf	License document.
HASP.exe	Hardware driver executable.
LicenseRemover.exe	Utility for removing software licenses from the system.
LicenseWizard.exe	Utility to add and check existing licenses on the system.
OEM_Matrikon_OPC.dll	MatrikonOPC OEM Badge Library.
OPCAuto.dll	MatrikonOPC Automation Component – enables developers to access OPC data from client applications developed using Automation tools such as *Visual Basic*, *VBA*, and *VB Script*.
opcda20_auto.doc	MatrikonOPC Automation Component Interface Standard
OPCDAAuto.dll	MatrikonOPC Automation Component – enables developers to access OPC data from client applications developed using Automation tools such as *Visual Basic*, *VBA*, and *VB Script*.
opchda10_auto.doc	Developer documentation for the HDA Automation Component.
opchda_ps.dll	The proxy-stub file to allow OPC Clients to make remote connections to an OPC HDA server.
OPCHDAAuto.dll	MatrikonOPC HDA Automation Component – enables developers to access OPC HDA data from client applications developed using Automation tools.
PSTCFG.exe	MatrikonOPC Product Configuration Utility.
PSTCFGBasicInnerLayerLib.ocx	Server ActiveX configuration panel.
PSTCFGInterface.dll	MatrikonOPC Product Configuration Interface Library.
PSTCFGScadaLib.ocx	Server ActiveX configuration panel.
PSTCFGps.dll	MatrikonOPC Product Configuration COM Marshalling Library.
PXPComfigps.dll	MatrikonOPC Product Configuration Library.

Table 70 - Files Installed in "Common" Folder

The files listed in Table 71 are installed in the ***WINDOWS\system32*** folder, or the ***WINDOWS\SysWOW64*** folder for 64-bit systems:

File Name	Description
EXPREVAL.DLL	Expression Evaluation Library for Alias Equations.
OPC_AEPS.DLL	OPC Alarms and Events 1.0 Interfaces Marshalling Library.
OPCBC_PS.DLL	OPC Batch Custom 2.00 Proxy/Stub Library.
OPCCOMN_PS.DLL	OPC Common Interfaces and Marshalling Library.
OPCDXPS.DLL	OPC Data eXchange 1.00 Proxy/Stub Library.

File Name	Description
OPCENUM.EXE	OPC Server List Component.
OPCHDA_PS.dll	OPC Historical Data Access 1.20 Proxy/Stub Library.
OPCPROXY.DLL	OPC Data Access 2.0 and 1.0a Interfaces and Marshalling Library.
OPCSEC_PS.DLL	OPC Security 1.00 Proxy/Stub Library.
ACTXPRXY.DLL	ActiveX Interface Marshalling Library.

Table 71 - Files Installed in "system32" Folder

Appendix K Un-Installation

To successfully un-install MatrikonOPC Server for SCADA Modbus, using the **Add or Remove Programs** from the Microsoft Windows **Control Panel** is recommended.

To un-install MatrikonOPC Server for SCADA Modbus:

1. Click on the **Start** button and highlight the **Control Panel** item.
2. From the displayed menu, select **Add or Remove Programs**.
3. The **Add or Remove Programs** window (Figure 94) is displayed.
4. Scroll through the list of currently installed programs and updates to find and select **MatrikonOPC Server for SCADA Modbus**.

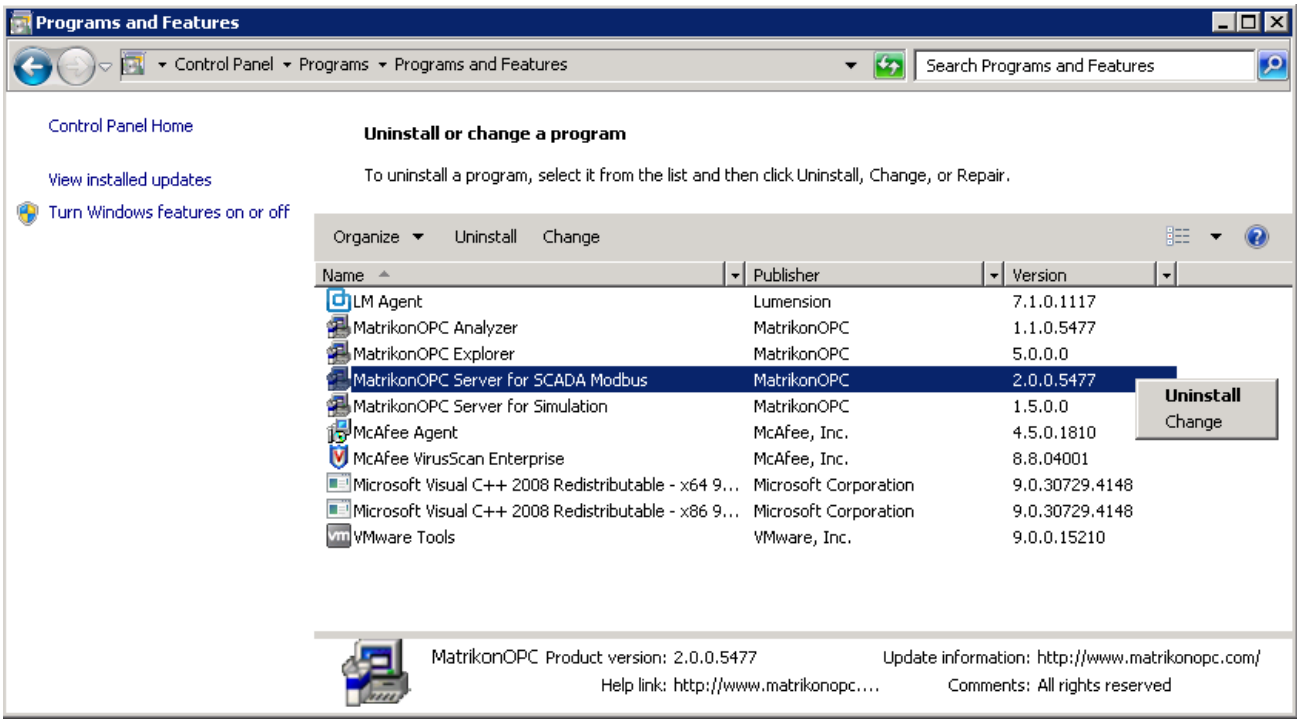


Figure 94 - Add or Remove Programs

5. Click on the **Remove** button associated with the MatrikonOPC Server for SCADA Modbus program to initiate the un-install process.
6. The **MatrikonOPC Server for SCADA Modbus – InstallAware Wizard** appears, and the **Welcome to MatrikonOPC Server for SCADA Modbus Maintenance** screen (Figure 95) is displayed.

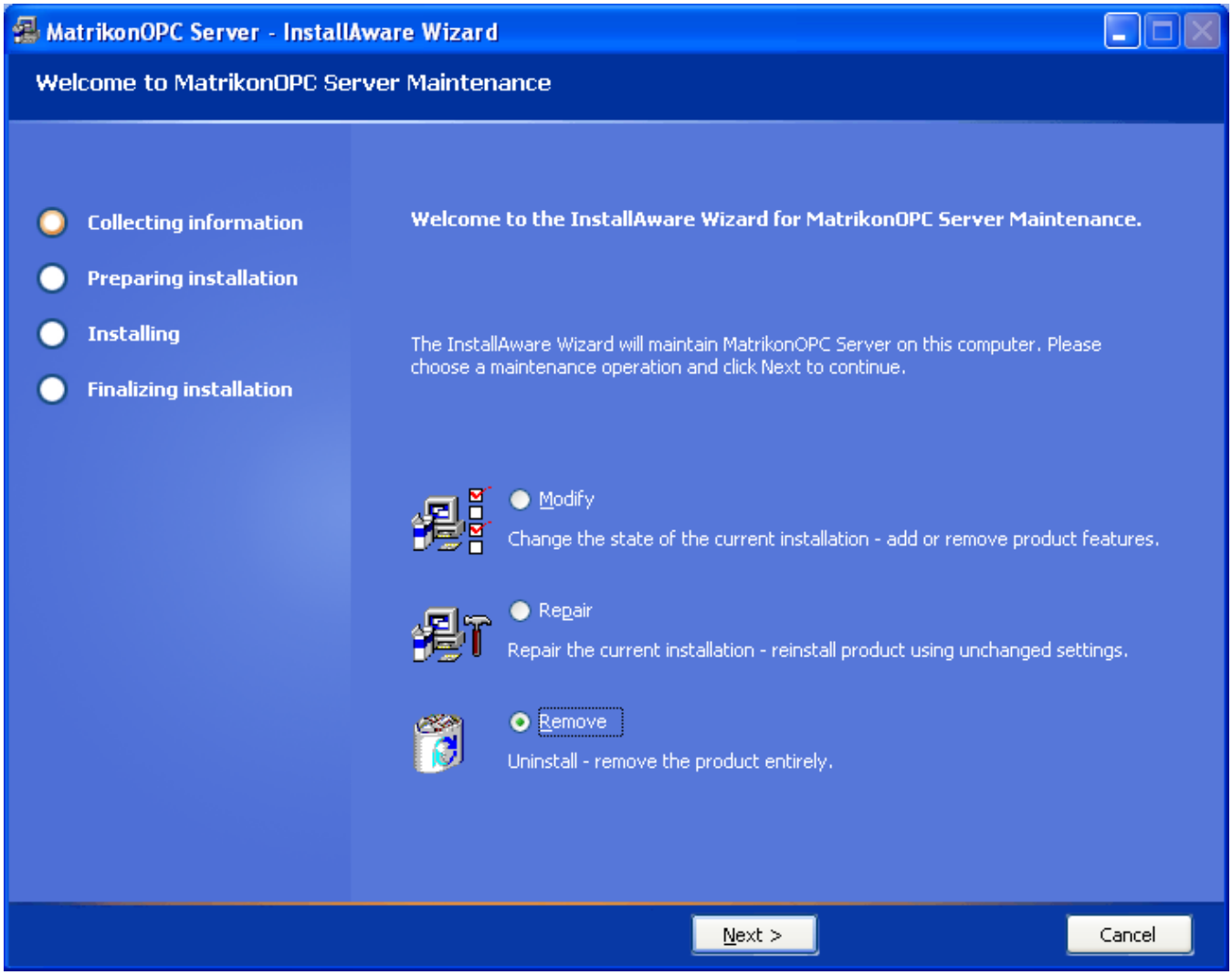


Figure 95 - Welcome to MatrikonOPC Server for SCADA Modbus Maintenance Screen

7. Select the **Remove** option button to un-install MatrikonOPC Server for SCADA Modbus entirely.
8. Click on the **Next** button. The **Ready to Uninstall** screen (Figure 96) is displayed.

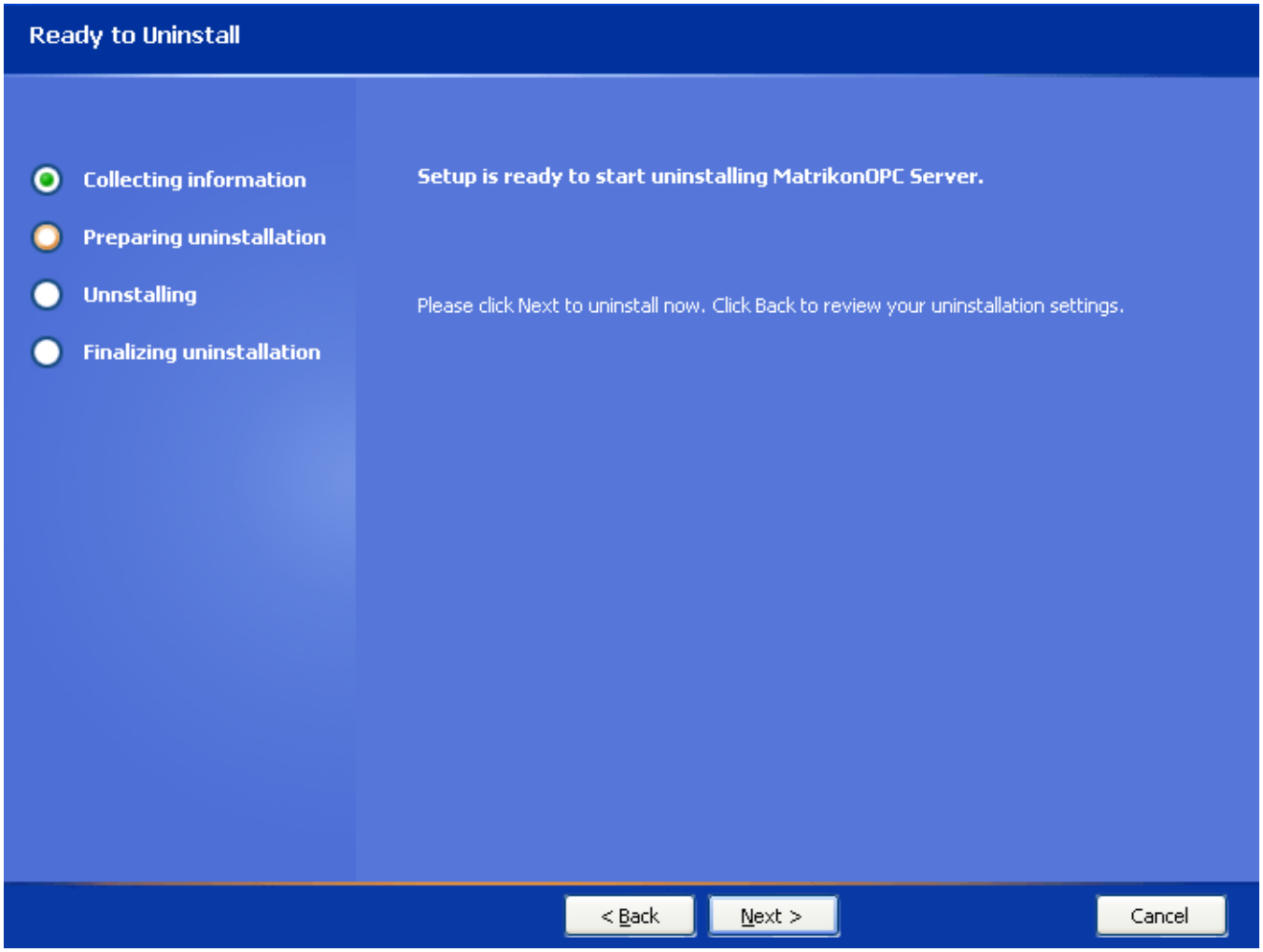


Figure 96 - Ready to Uninstall Screen

9. Click on the **Next** button.
10. The **Uninstalling MatrikonOPC Server for SCADA Modbus** screen (Figure 97) appears and the un-install takes place.

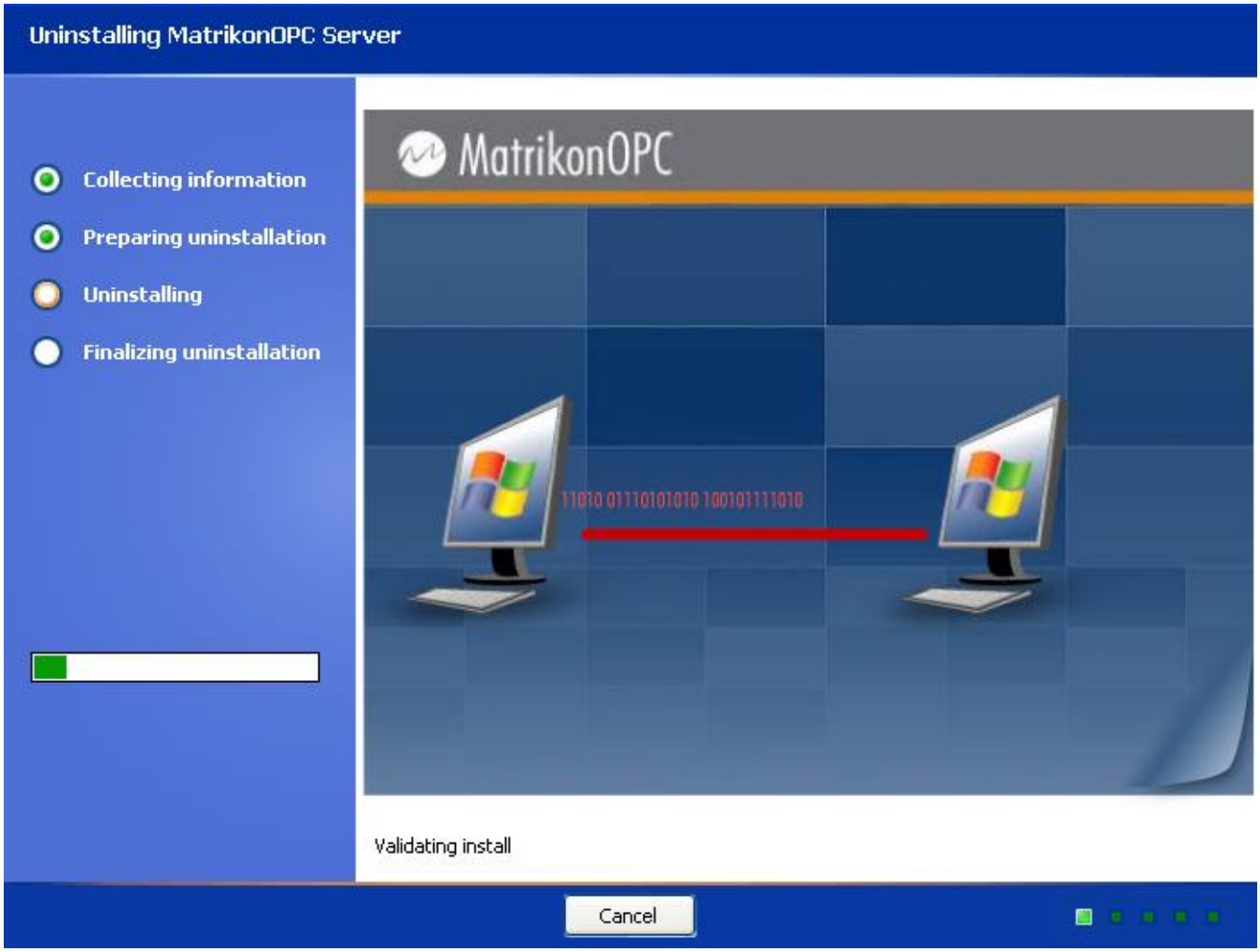


Figure 97 - Uninstalling MatrikonOPC Server for SCADA Modbus Screen

11. When the un-install has finished, the **MatrikonOPC Server for SCADA Modbus Setup Complete** screen (Figure 98) appears stating that MatrikonOPC Server for SCADA Modbus was successfully un-installed.

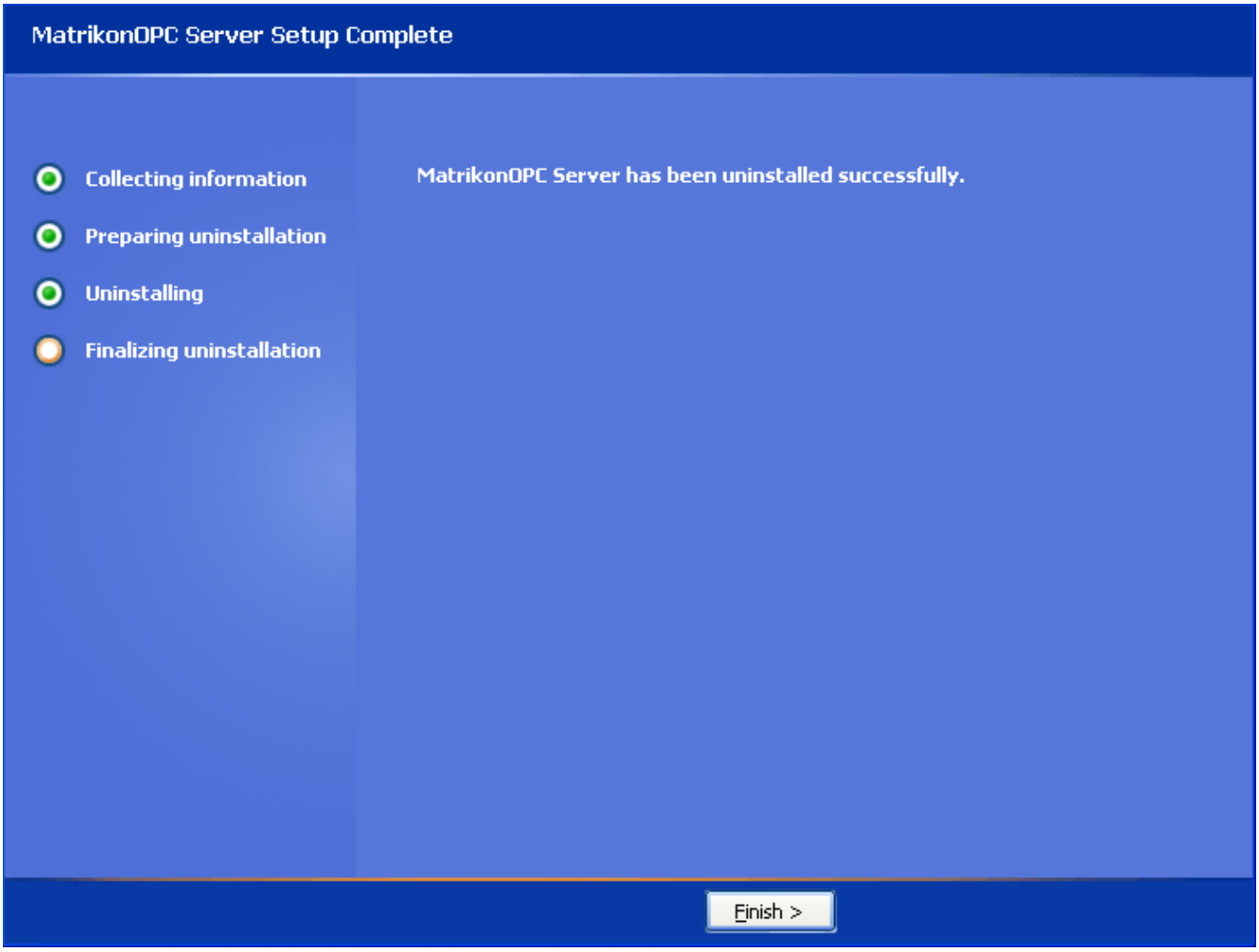


Figure 98 - MatrikonOPC Server for SCADA Modbus Setup Complete Screen

12. Click on the **Finish** button to complete the un-install and exit the Wizard.
13. The program no longer appears listed in the **Add or Remove Programs** window.

Appendix L SCADA Servers Connection Logic

Introduction

In MatrikonOPC SCADA servers, the data is polled in a round-robin mechanism (as long as the items belong to the same Channel Device link). This means that the OPC items will be scanned in sequence depending on the unit device links to which they belong. If one of the device links fail then the server applies the retry sequence on this device link. If after applying the retry sequence, the device link is still failing, the server moves to the next device link in the sequence. The failing device link is treated differently than the rest of the working device links based on the failure and reconnection logic settings that were entered for that particular device link.

Polling

Polling is described as the operation where the server requests and extracts data from the device. As explained in the Introduction section of this appendix, polling is done in a round-robin mechanism depending on which device links are currently being scanned.

Every Channel device link does its own polling for all of its sub-device links. If your server configuration contains multiple Channel device links then they will do the polling independently of each other.

Note: If a unit device link does not have any items added by an OPC client, it will disconnect until a client requests items that belong to that unit device link. Some servers, like the MatrikonOPC Server for SCADA Modbus, may have an option to force a unit device link to stay connected at all times.

Connection Failures and Reconnection Logic

In the case of a connection failure, the reconnection logic (meaning the timeout values, retries counts, and other related options) applied will depend on the type of device link that failed.

Example:

Suppose in our configuration we have a Network Channel device link called ***Channel***. Under the Network Channel device link we have a Network Host device link named ***Host***. Under the Host device link we have multiple unit device links (***U1***, ***U2***).

If failure happens due to an IP Address problem, the reconnection logic applied will depend on the values and options defined in the ***Host*** *d*evice link.

If a failure happens due to a Unit Address problem, the reconnection logic parameters will be those defined in the unit device link that failed.

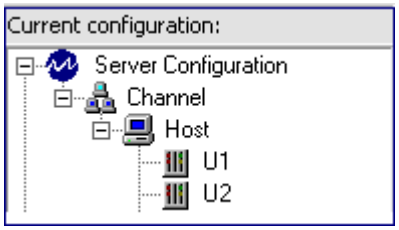


Figure 99 - Example

Appendix M Features and Examples

Introduction

This appendix describes the common features and newest enhancements added for the MatrikonOPC Server for SCADA Modbus, complete with examples.

Round-Robin Example

As explained in **Appendix L – SCADA Servers Connection Logic**, all of the MatrikonOPC SCADA servers poll in a round-robin mechanism. The MatrikonOPC Server for SCADA Modbus is not an exception to that. The best way to explain that is by an example:

Example 1 (Unit Device Link Failure)

Suppose your OPC client is currently scanning the following items (or tags):

- ***A1***, ***A2***, and ***A3*** from device link **Unit A**
- ***B1***, ***B2***, and ***B3*** from device link **Unit B**
- ***C1***, ***C2***, and ***C3*** from device link **Unit C**.

During polling, scanning goes through device link **Unit A** followed by device link **Unit B**, and finally device link **Unit C**. The items are then read in the following order:

A1, A2, A3, B1, B2, B3, C1, C2, C3

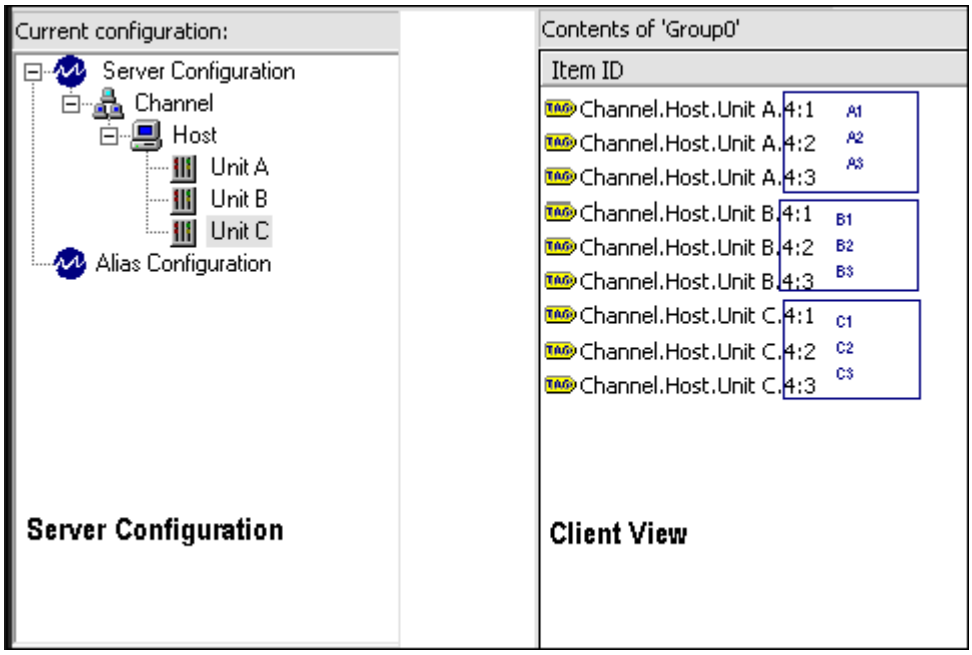


Figure 100 - Example 1

Suppose that the communication on **Unit B** failed for any reason. For this example, we will imagine that the failure came from the unit device link itself and that all of its parent device links are fine. In the next polling cycle, items ***A1***, ***A2***, and ***A3*** are scanned and the server identifies the fact that there is a problem accessing the items that belong to **Unit B**. The server then applies the number of connection retries defined in the **Retries** field on the **Data Acquisition** tab.

After the specified number of retries is consumed, scanning will continue on to the next device link, and then the next, in order. In the next polling cycle, if **Unit B** is still failing the server checks the

Failed Polls Allowed option on the **Data Acquisition** tab. This option is used to specify how many more cycles the server tries the number of retries defined in the **Retries** field before determining the connection has failed and setting the **Quality** for all to ***Bad, Last Known Value***.

After the number of poll cycles defined in the **Failed Polls Allowed** field has elapsed, the device link will no longer be in the polling sequence. This means that **Unit C** will be scanned directly after **Unit A**, and **Unit B** will be skipped.

At this point the server checks the **Reconnect delay** option defined in **Unit B** and it will only try to connect to **Unit B** at that delay interval. That is, if **Reconnect delay** is set to ***X***, the server tries to connect to **Unit B** every ***X*** seconds. Then, every time the server tries to connect to **Unit B** it applies the number of retries defined in the **Reconnect retries** field.

The previous scenario can be summed up in the following steps:

1. **Unit B** disconnects due to a communication failure.

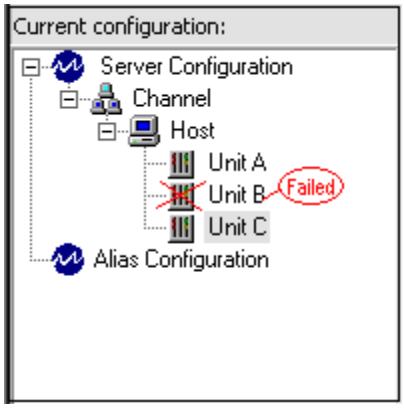


Figure 101 - Unit B Fails

2. The retry sequence is applied to **Unit B** and the server continues scanning the remaining device links.

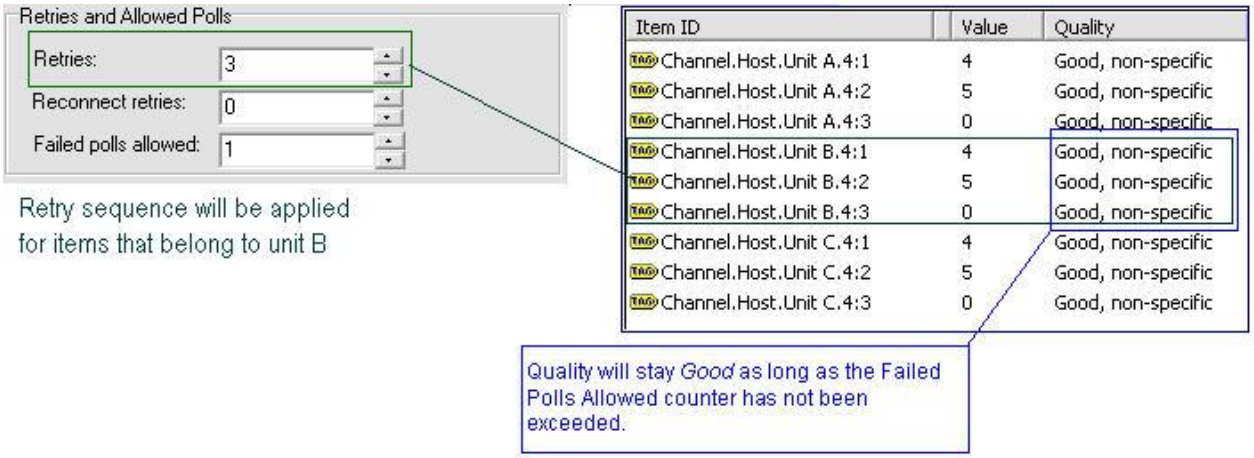


Figure 102 - Retry Applied on Unit B Items

3. The server checks the **Failed Polls Allowed** counter. If the counter has not yet been exceeded, the same retry sequence is applied to **Unit B** in the next poll, and the **Quality** for the items stay ***Good***.

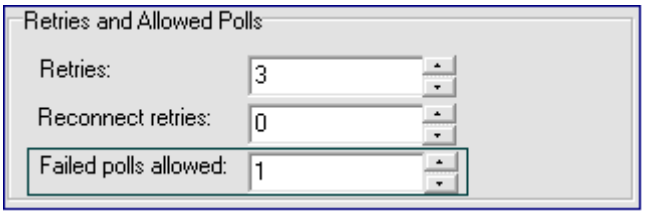


Figure 103 - Failed Polls Allowed Counter

4. After the **Failed Polls Allowed** counter has been exceeded, the **Quality** of the those items that belong to **Unit B** will change to ***Bad*** and it is removed from the polling sequence.

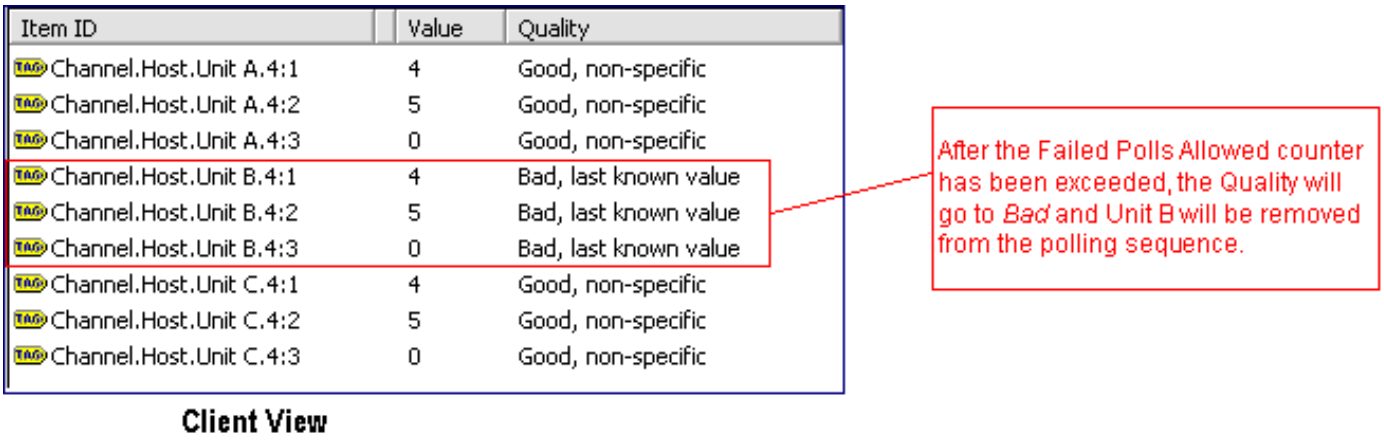


Figure 104 - Unit B Confirmed Failure

5. The server checks the **Reconnect delay** interval and if it passes, the server tries to reconnect to **Unit B** again.

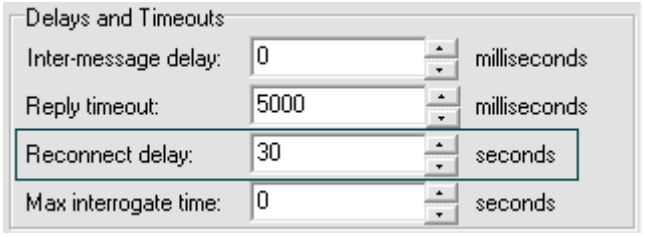


Figure 105 - Reconnect Delay

6. Thereafter, the number of retries applied is that which has been defined in the **Reconnect retries** field of **Unit B**.

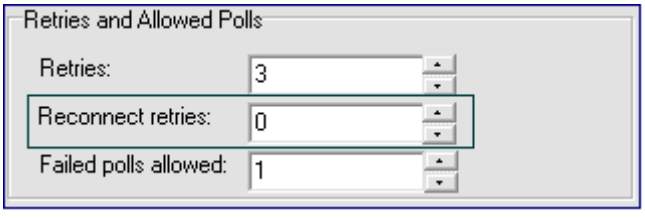


Figure 106 - Reconnect Retries

Note: The reconnection logic applied depends on which type of device link failed to connect. For example, if what fails is the Modbus Unit address (defined in the unit device link) then the reconnection logic applied will depend on the options defined in the unit device link. On the other hand, if what fails is the Socket then the reconnection logic applied will depend on the options defined in the Network Host device link.

Scanning and *Stay Connected* Option

In the normal operation any unit device link that is currently not being scanned from an OPC client, will disconnect until a client requests items that belong to that unit device link. To force the unit device link to be connected at all times even if none of its items are being currently scanned, the **Stay Connected** option must be selected.

Note: This option needs to be used when the unit device link is used as a redundant link in a Redundancy Group.

writeLimit OPC Item

Under any Channel device link, there is an OPC item called **writeLimit**. This OPC item was designed to prevent the writes from taking up the whole channel bandwidth. This item can be accessed and edited through OPC. This item indicates the number of successive writes allowed before forcing the next poll in the queue for that specific Network Channel. If the value of that item is set to ***0*** then the feature will be turned off. After the forced poll is completed, the server reschedules the polls for that specific network channel starting now.

Example:

Suppose the OPC client is currently scanning items ***A***, ***B***, and ***C*** from the **Network Channel**, as well as the **writeLimit** OPC item. Suppose the OPC client's update rate is set to ***23*** seconds.

Assume we performed the following actions:

- Wrote **10** in the **writeLimit** OPC item.
- Wrote **4** successive values in **Item A** in a time span of 4 seconds.
- Wrote **3** successive values in **Item B** in a time span of 3 seconds.
- Wrote **5** successive values in **Item C** in a time span of 5 seconds.

Since the client's update rate is 23 seconds, the data is requested from the client every 23 seconds, which means polling happens every 23 seconds.

The server performs the **4** successive writes to **Item A**, then **3** successive writes to **Item B**, and then it will do only **3** of the **5** writes to **Item C**. The server then forces the next poll to occur immediately instead of waiting for it until 10 writes have passed. The polls are then rescheduled for this specific Network Channel starting now, meaning the next poll happens exactly 23 seconds from now, and so on.

PacketSent and *PacketReceived* OPC Items

PacketSent and ***PacketReceived*** OPC items are found only under the unit device links and they can only be accessed through OPC. The ***PacketSent*** item shows the last packet that was sent from the server in a hexadecimal format. The ***PacketReceived*** OPC item shows the last packet that was received from the device in a hexadecimal format.

Nc.H2.U.PacketReceived	02 03 0A 00 00 00 00 00 00 00 00 04	Good, non...	01/06/200...	Active
Nc.H2.U.PacketSent	02 36 00 00 00 06 02 03 00 00 00 05	Good, non...	01/06/200...	Active

Figure 107 - Packets Sent and Received at a Certain Point

Note: These OPC items update at the rate the client requests. So, if the packets are sent out or received faster than the update rate, the OPC client will miss some of the hexadecimal data.

Total Reads and Writes Indicators

For any type of device link in the MatrikonOPC Server for SCADA Modbus, there are four OPC items (or counters) that can be used to monitor the total number of write or read request packets that are being sent by the server. This is also the case for the total number of reply packets received from the device. Those items are accessible through OPC and can be found under any type of a device link. The four items are defined as follows:

- **TotalReadsSent** - the number of read request packets sent by the server through this specific device link until now.
- **TotalReadsReceived** - the number of reply packets received from the device until now, in response to the read requests.
- **TotalWritesSent** - the number of write request packets sent by the server through this specific device link until now.
- **TotalWritesReceived** - the number of reply packets received from the device until now, in response to the write requests.

Figure 108 shows an example where the number of packets is being monitored for every type of device link. The server configuration being used contains the Network Channel device link ***Nc***, the Network Host device link ***H2*** and the Modbus Unit device link ***U***.

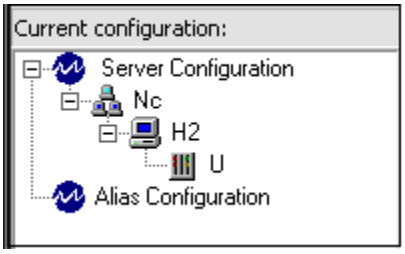


Figure 108 - Server Configuration

Item ID	Access Path	Value	Quality	Timestamp	Status
Nc.H2.TotalReadsReceived		5620	Good, non-specific	01/06/2009 12:23:21.020 PM	Active
Nc.H2.TotalReadsSent		2811	Good, non-specific	01/06/2009 12:23:21.005 PM	Active
Nc.H2.TotalWritesReceived		2	Good, non-specific	01/06/2009 11:37:00.170 AM	Active
Nc.H2.TotalWritesSent		1	Good, non-specific	01/06/2009 11:37:00.154 AM	Active
Nc.H2.U.TotalReadsReceived		5620	Good, non-specific	01/06/2009 12:23:21.020 PM	Active
Nc.H2.U.TotalReadsSent		2811	Good, non-specific	01/06/2009 12:23:21.005 PM	Active
Nc.H2.U.TotalWritesReceived		2	Good, non-specific	01/06/2009 11:37:00.170 AM	Active
Nc.H2.U.TotalWritesSent		1	Good, non-specific	01/06/2009 11:37:00.154 AM	Active
Nc.TotalReadsSent		2811	Good, non-specific	01/06/2009 12:23:21.005 PM	Active
Nc.TotalWritesReceived		2	Good, non-specific	01/06/2009 11:37:00.170 AM	Active
Nc.TotalWritesSent		1	Good, non-specific	01/06/2009 11:37:00.154 AM	Active
Nc.Totalreceived		2811	Good, non-specific	01/06/2009 12:23:21.020 PM	Active

Figure 109 - Monitoring Request and Reply Counters from the Client

In Figure 109 we can see that the number of read packets sent by **Nc.H2.U** *(*which is the full path of the unit device link **U***)* was ***2811***. The same unit device link received ***5620*** packets meaning that for every request packet being sent by the server, there is an average of approximately ***2*** reply packets received from the device.

We can also see that the total number of write requests that was sent by the server (i.e., ***TotalWritesSent***) is ***1*** which means that only one write operation was done up until this point. This write operation had ***2*** reply packets received from the device to indicate success or failure.

In this example, the values of those counters were the same for the different type of device links. It should be noted that is not always the case. If the Network Host device link had two unit device links being scanned instead of one, the values of the counters on the Network Host device link will be more than those of its individual unit device links. This is because it will be the summation of the number of packets of all of its sub-device links that are currently being scanned by the OPC client.

Note: The qualities of any of the four OPC items will be ***Bad*** until a write or a read operation is completed for the device link to which they belong.

Appendix N OPC Items

Introduction

This appendix explains the OPC items that can be found in SCADA servers and what they are used for.

Base OPC Items

These OPC items are found in the Root Device Link of the server. This link is the root for all of the configured device links.

OPC Item	Description
#MonitorACLFile	A Boolean that indicates if the server is currently monitoring the Access List file or not (i.e., if the security is enabled or not).
@Clients	A string the shows the clients the server is currently connected to.
@Purge	If ***1*** is written to that item, all of the device links that are marked for deletion will be removed as long as they are neither enabled nor connected to a client.
@ClientCount	This is a read-only unsigned integer item which provides the count of OPC clients connected to the server. **Note:** This count will include opened PSTCFG GUI.

Table 72 - Root Device Link OPC Items

Generic OPC Items

OPC Item	Description
Bpsin	Number of bytes received by the server per second.
Bpsout	A counter indicating the number of bytes sent from the server per second.
Broadcast	Number of broadcast packets sent until now.
Bytesin	Number of bytes received by the server until now.
Bytesout	Number of bytes sent from the server until now.
Checksumerror	A counter indicating the number of times a check sum error or invalid response occurs.
Commfail	A Boolean that indicates whether communication failed or not on that specific device link. This option exists in all types of device links
Configuration	Shows the configuration for that type of device link in a string format. This option exists for all types of device links
Connected	A Boolean that indicates whether that specific device link is connected or not.
Currentunit	Indicates the full path for the unit device link in a blank string that is currently polling. It should be noted that if the polling operation is very fast, you might not have a chance to see the value because when the polling is done the value returns to a blank string.
Deleted	Disables the device link and mark it for deletion.
Description	Description defined for the current device link.

Enabled	Indicates whether the device link is enabled or not.
Errorresponse	The number of errors received from the device until now for the current object.
Failure	Number of failures that happened up until now, for the current object. The same value can be seen in the server's GUI in the lower left-hand side under the name **Failed**.
Incomplete	A counter indicating the number of incomplete responses from the device.
Invalid	A counter that indicates the number of invalid or incomplete responses from the server.
Lastresponse	A timestamp that shows the time of the last response received from the device.
Lastsent	A timestamp that shows the time the last packet was sent to the device.
Name	The name of the current object.
Noresponse	A counter that increments when there is no response from the device.
Overrun	A counter that indicates the number of overruns that happened up until now. This counter can also be seen in the server's GUI the lower left-hand side.
Precentreturn	The percentage of number of packets received compared to the number of packets sent.
Percentvalid	The percentage between (the total packets initially sent + the retries) and (the total packets received + the error responses).
Pollcomplete	The number of polls completed up until now.
Received	The number of packets received up until now.
reconnectDelay	The value set for the reconnect delay option in the server's GUI.
Resetstats	Resets the statistics of the server.
Retried	A counter for the number of retries accomplished up until now.
Scan	A Boolean that is ***True*** if the object is currently scanning.
Sent	A counter that indicates the number of packets sent from this object up until now. This option can only be seen in the lower left-hand side of the server's GUI.
TimeOut	A counter indicating the number of timeouts.
TotalReadsReceived	The total number of read packets received to the current object up until now.
TotalReadsSent	The total number of read packets sent from the current object up until now.
TotalWritesReceived	The total number of write packets received to the current object up until now.
TotalWritesSent	The total number of write packets sent from the current object up until now.
Totalsent	Total number of packets sent from the current object and any sub-objects.
Totalreceived	Total number of packets received to the current object and any sub-objects.
Unsolicited	A counter indicating the number of unsolicited messages.

Table 73 - Generic OPC Items

Network Channel Object OPC Items

OPC Item	Description
InterDeviceDelay	An OPC item that gives access to the **Inter-Device Delay** option which allows you to enter a value specifying the minimum amount of time (in milliseconds) between the last response from one unit, and the next request to another unit on this channel.
LogRx	An OPC item that gives access to the **Log bytes received** option which shows the hexadecimal format of the packets received in the log file (***-1*** means the option is enabled and ***0*** means the option is disabled).
LogTx	An OPC item that gives access to the **Log bytes transmitted** option which shows the hexadecimal format of the packets sent in the log file (***-1*** means the option is enabled and ***0*** means the option is disabled).
MaxUnsolicited	Maximum unsolicited messages allowed. Default is ***2048***.
MinUnsolicited	Minimum unsolicited messages allowed. Default is ***4***.
writeLimit	This OPC item is fully explained in **Appendix M – Features and Examples**.
Scheduleallpolls	This OPC item allows access to the **Schedule all polls** checkbox in the server's GUI.
Threshold	This OPC item allows access to the **Threshold rate** field in the server's GUI. The OPC item is measured in milliseconds, whereas the field itself in the server's interface is in seconds.

Table 74 - Network Channel OPC Items

Network Host Object OPC Items

OPC Item	Description
SocketTO	The value of the socket timeout. This is the same as the **Socket timeout** field in the server's GUI for the network host object.
Host	The IP address of the remote device. This is the same as the **Host name/IP** field in the server's GUI for the network host object.
Service	The port number of the remote device. This is the same as the **Service name/port** field in the server's GUI for the network host object.
Reconnectdelay	The reconnect delay configured for the connection. This is the same as the **Reconnect delay** field in the server's GUI for the network host object.

Table 75 - Network Host OPC Items

Modbus Unit Object OPC Items

OPC Item	Description
PacketSent	The hexadecimal format of the last packet sent from the server.
PacketReceived	The hexadecimal format of the last packet received to the server.
Name	User name assigned to the object, which becomes an element of the item ID for data items. Automatically converted to title case for display purposes.

OPC Item	Description
Enabled	Enables or disables communication for the object.
Description	User-defined text (64-character maximum) for documentation purposes.
Address	This OPC Item is the value defining a Modbus slave address (***1-255***). A value of ***0*** indicates a broadcast unit (write-only). Changing this option requires the server to reset communication with this unit.
Mode	This OPC item is for entering a mode indicating whether the server communicates with this unit as a master or by emulating a slave. Changing this Item requires the server to reset communication with this unit. It accepts two values: ***0-Master*** and ***1-Slave***.
Format	Indicates the message format to use for communication. Changing this option requires the server to reset communication with this unit. It accepts three values: ***0-RTU(CRC16)***, ***1-ASCII(LRC)***, and ***2-RTU TCP***.
Basepolltime	This OPC item allows you to enter a normalization time for scheduling polls, stored as UTC. Changing this item requires the server to re-optimize communication with this unit.
Writeoptimism	This OPC item displays the level of optimism for writing control values to the unit: ***0 pessimistic***, ***1 semi-optimistic***, and ***2 fully optimistic***. If the unit is configured for ***fully optimistic*** writes, the server immediately copies the written value to the item's cache with an ***Uncertain*** quality before sending the write request. If the unit is configured for either ***fully optimistic*** or ***semi-optimistic*** writes, the server copies the written value to the item's cache with either a ***Good*** or a ***Bad*** quality (depending on result of write operation) after receiving the response from the device. If the unit is configured for ***pessimistic*** writes, the server does not alter the item's cache during the write operation. The server updates that value for the item on the next read.
Allowwrites	This OPC item allows writes to the unit. Writes are not allowed when the value is ***0***. Writes are allowed when the value is ***-1***.
AllowWriteInterleave	With this option enabled (value = **1**), server gives better response on write requests. Can be used in frequent and heavy write request scenario. **Note:** *It should be used with due consideration as this may degrade response on read requests. After this configuration value is changed, restart the server for the change to take effect.*
Latchdata	This OPC item enables the option to have items "freeze" at their last Value/Quality/Timestamp when the unit is set off-scan. A value of ***-1*** indicates that the option is enabled, and a value of ***0*** indicates the option is disabled.
VerifyMessageSize	This OPC item is the same as the **Verify Message Size** checkbox in the server's GUI. When this option is enabled the OPC server verifies that the response message has the correct length before accepting it. A value of -***1*** indicates that the option is enabled and a value of ***0*** indicates the option is disabled.
StayConnected	This OPC item is the same as the **Stay Connected** checkbox in the server's GUI. If that option is enabled, the device link always stays connected even if items from that device link are not currently being scanned by an OPC client.

OPC Item	Description
	A value of ***-1*** indicates that the option is enabled and a value of ***0*** indicates the option is disabled. **Note:** This option is required to be on if this unit is being used in a Redundancy Group configuration.
Intermessagedelay	This OPC item allows you to enter a value defining the minimum amount of time (in milliseconds) between the last request/response to/from this unit and the next request to this same unit. **Note:** If this is set to ***0*** then the setting will be disabled. Default = ***0***.
Replytimeout	This OPC item allows you to enter a value defining the maximum amount of time (in milliseconds) to wait for the response to a request. **Note:** If this setting is too short, communication can become confused. Default = ***5000***.
Retries	This OPC item allows you to enter a value defining the maximum number of times to resend a request and wait for a response before concluding that communication has failed with the unit.
Reconnectdelay	This OPC item allows you to enter a value defining the minimum amount of time (in seconds) to wait before attempting to reconnect after communication has failed with a unit. Default = ***30***.
Reconnectretries	This OPC item allows you to enter a value defining the maximum number of times to resend a request and wait for a response when trying to reconnect after communication has failed with a unit.
FailedPollsAllowed	This OPC item is the maximum number of polls allowed to elapse after the first communication failure before the **Quality** goes to *Bad* and considers the device link as having an ongoing communication failure. Refer to **Appendix L – SCADA Servers Connection Logic** for more details Default = ***1***.
Interrogate	This OPC item allows you to command the server to "interrogate" the Modbus unit. Interrogation of a unit means to place a particular unit on a communication channel into interrogate mode. This has the effect of interleaving high-priority polls for that unit into the round-robin sequence. The interleaved polls only request items with scan rates less than the Threshold Rate; scheduled reads continue in the normal fashion (unaffected). To trigger an interrogation, a non-zero value should be written to this item. The unit remains in interrogation mode until 0 is written or until the amount of time configured in the MaxInterrogate item has elapsed. Default = ***0***.
Maxinterrogate	This OPC item allows you to enter a value defining the maximum amount of time (in seconds) to allow a unit to be interrogated. A value of ***0*** indicates that interrogation must be stopped manually. Default = ***0***.
Maxreadlength	This OPC item allows you to enter a value defining the maximum number of

OPC Item	Description
	registers (or blocks of 16 coils) per read request. **Note:** Changing this option requires the server to re-optimize communication with this unit.
Maxreadspaces	This OPC item allows you to enter a value defining the maximum allowable number of consecutive inactive registers (or blocks of 16 coils) to include in a block read for optimization. **Note:** Changing this option requires the server to re-optimize communication with this unit.
Maxwritelength	This OPC item allows you to enter a value defining the maximum number of registers (or blocks of 16 coils) per write request. Set to ***1*** to force single writes. **Note:** Changing this option requires the server to re-optimize communication with this unit.
LastExceptionCode	This OPC item displays the last exception code received from the device. Default = ***0***(success).
LastExceptionString	This OPC item displays the details about the last exception received from the device.
cpuThrottle	This is same as the CPU Throttle field in server's GUI for the Modbus unit object under Optimization tab.
cpuSleep	This is same as the CPU Sleep field in server's GUI for the Modbus unit object under Optimization tab.

Table 76 - Modbus Unit OPC Items

REGISTER Addressing

The type and address of a Register is defined in the Modbus Unit Object level. Refer to the **OPC Data Items** section of this manual for more information.

Fail-Over Channel Object OPC Items

OPC Item	Description
Name	User name assigned to the object, which becomes an element of the item ID for data items. Automatically converted to title case for display purposes.
Enabled	Enables or disables communication for the object.
Description	User-defined text (64-character maximum) for documentation purposes.
Configuration	Shows the configuration for the Fail-Over Channel object.

Table 77 - Fail-Over Channel OPC Items

Serial Channel Object OPC Items

OPC Item	Description
Name	User name assigned to the object, which becomes an element of the item ID for data items. Automatically converted to title case for display purposes.
Enabled	Enables or disables communication for the object.

OPC Item	Description
Description	User-defined text (64-character maximum) for documentation purposes.
Device	This OPC item allows you to enter the COM port number for the connected device.
Baud	This OPC item is for entering the Baud rate of the communication through OPC. Values range between ***110*** and ***256000***. Default = ***9600***.
Parity	This OPC item displays and allows selecting the type of parity to be used. Options available are ***0-none***, **1-*odd***, **2-*even***, ***3-mark***, and ***4-space***. Default = ***none***.
Databits	This OPC item is for entering the number of data bits through OPC. Values range between ***4*** and ***8***. Default = ***8***.
Stopbits	This OPC item is for entering the number of stop bits through OPC. Values available are ***1***, ***1.5***, and ***2***. Default = ***1***.
RTScontrol	Same as the RTS Line option in the GUI. Select a Request To Send (RTS) preset. Available options are **0-*disable***, **1-*enable***, **2-*handshake***, and **3-*toggle***. Default = ***disable***.
DTRcontrol	Same as the DTS Line option in the GUI. Allows you to select a Data Terminal Ready (DTR) Available options are **0-*disable***, **1-*enable***, and **2-*handshake***. Default = ***disable***.
OutxDsrFlow	This OPC item allows you to enable or disable Data Set Ready (DSR) flow control. Default = ***0***. A value of ***-1*** indicates that the option is enabled and a value of ***0*** indicates the option is disabled.
DSRsensitivity	This OPC item allows you to enable or disable Data Set Ready (DSR). Default = ***0***. A value of ***-1*** indicates that the option is enabled and a value of ***0*** indicates the option is disabled.
OutxCtsFlow	This OPC item allows you to enable or disable Clear To Send (CTS) flow control. Default = ***0***. A value of ***-1*** indicates that the option is enabled and a value of ***0*** indicates the option is disabled.
OutxOnOffFlow	This OPC item allows you to enable or disable XON/XOFF software flow control. Default = ***0***. A value of ***-1*** indicates that the option is enabled and a value of ***0*** indicates the option is disabled.
Interdevicedelay	This OPC item allows you to enter a value specifying the minimum amount of time (in milliseconds) between the last request/response to/from one unit and the next request to another unit on this channel. If this OPC Item is set to 0 then the option will be disabled. Default = ***0***.

OPC Item	Description
Reconnectdelay	This OPC item allows you to enter a value specifying the minimum amount of time (in seconds) to wait before attempting to recover communications after the communication channel itself has failed. Default = ***1***.
Threshold	This OPC item allows you to enter a value specifying the maximum achievable scan interval (in seconds) for the circuit. The server polls items with update rates faster than this on a continuous basis. Changing this option requires the server re-optimize communications with all units on this channel. If it is set to 0 then the option is turned off. Default = ***0***. **Note:** All units on the channel must re-optimize if this parameter changes.
ScheduleAllPolls	This OPC item is the same as the **Schedule all polls** option in the server's GUI. When this option is enabled, rather than polling them continuously, the server schedules items with update rates faster than the threshold rate to poll at the threshold rate. Default = ***1***.
ReadIntervalTimeout	This OPC item allows you to enter a value specifying the maximum amount of time (in milliseconds) between the reception of one character and the arrival of the next. A value of ***0*** indicates no limit. Default = ***0***.
backoffDelay	This OPC item allows you to enter a value specifying the minimum amount of time (in milliseconds) for DCD to be low before transmitting to avoid collisions. Set both **backoff delay** and **backoff random** items to ***0*** to ignore DCD. Default = ***0***.
backoffRandom	This OPC item allows you to enter a value specifying the maximum amount of extra random time (in milliseconds) for DCD to be low before transmitting. Set both **backoff delay** and **backoff random** items to ***0*** to ignore DCD. Default = ***0***.
writeLimit	This OPC item is fully explained in **Appendix M – Features and Examples**.

Table 78 - Serial Channel Object OPC Items

Radio Object OPC Items

OPC Item	Description
Name	User name assigned to the object, which becomes an element of the item ID for data items. Automatically converted to title case for display purposes.
Enabled	Enables or disables communication for the object.
Description	User-defined text (64-character maximum) for documentation purposes.
Keyup	This OPC item allows you to enter a value defining the minimum amount of time (in milliseconds) to wait after setting the RTS signal to allow the transceiver to "warm up".
keydown	This OPC item allows you to enter a value defining the minimum amount of

OPC Item	Description
	time (in milliseconds) to wait before clearing the RTS signal to allow the transceiver to "cool down".

Table 79 - Radio Object OPC Items

Dial-Up Object OPC Items

OPC Item	Description
Answered	Indicates that a call has been answered.
Busy	Indicates that the line was busy.
Carrier	A Boolean that indicates if there is a carrier for the Dial-Up object or not.
Dialout	Indicates that a dial out occurred.
Lastanswered	The date of the last time a call was answered.
Lastdialout	The date the last time a dial out occurred.
Lastresponse	The date of the last response.
Lastsent	Last sent time.
Nocarrier	Indicates that no carrier was found in the response.
Nodialtone	Indicates that there is no dial tone.
Noresponse	Indicates that there is no response.
Percentanswered	The percentage of calls answered based on the number of dial outs.
Name	User name assigned to the object, which becomes an element of the item ID for data items. Automatically converted to title case for display purposes.
Enabled	Enables or disables communication for the object
Description	User-defined text (64-character maximum) for documentation purposes.
Dial	This OPC item allows you to enter any valid sequence of characters that can follow an ADT command.
PostConnectDelay	This OPC item allows you to enter a value specifying the minimum amount of time (in seconds) to wait after establishing a dial-out connection before transmitting.
Redials	This OPC item allows you to enter a value defining the maximum number of times to redial a connection before concluding that communication has failed with the units on that connection.
ReconnectDelay	This OPC item allows you to enter a value specifying the minimum amount of time (in seconds) to wait before attempting to recover communications after communication has failed with the connection.
ReconnectRedials	This OPC item allows you to enter a value defining the maximum number of times to redial when trying to reconnect after communication has failed with a dial-up connection.
Reset	This OPC item allows you to select a modem reset command. This command is sent before initialization commands.

ResetDelay	This OPC item allows you to enter a value specifying the amount of time (in milliseconds) to wait after the reset command completes.
Initialize	This OPC item allows you to enter modem initialization command or commands, issued before dialling out and after hanging up. The command should always end with *E0Q0V1*.
CommandRetries	This OPC item allows you enter a value to define the maximum number of times to resend a command wait for a response before concluding that communication has failed with the modem.
CommandTimeout	This OPC item allows you to enter a value defining the maximum amount of time (in seconds) to wait for the modem to respond to AT commands with a valid result code.
DialTimeout	This OPC item allows you to enter a value defining the maximum amount of time (in seconds) to wait for the modem to receive a carrier from the dialled connection before aborting the call.
HangupTimeout	This OPC item allows you to enter a value defining the maximum amount of time (in seconds) to wait for the modem to hang up after issuing the ATH0 command.
Idle timeout	This OPC item allows you to enter a value defining the maximum amount of time (in seconds) to allow an idle connection to continue (i.e., no communication) before hanging up.
CarriageReturn	This item is the same as the **Command terminator** option in the GUI. Allows you to enter or select a value defining an ASCII character to be used to terminate commands and responses (usually a carriage return).
EscapeCharacter	This OPC item allows you to enter a value defining an ASCII character that is sent three times to escape into command mode (usually a plus sign).
EscapeGuardTime	This OPC item allows you to enter or select a value defining the amount of time (in milliseconds) to wait before and after sending the escape sequence.

Table 80 - Dial-Up Object OPC Items